applied sciences

Special Issue Reprint

Artificial Intelligence in Fault Diagnosis and Signal Processing

Edited by
Roque Alfredo Osornio-Rios, Athanasios Karlis and Andres Bustillo Iglesias

mdpi.com/journal/applsci

MDPI

Artificial Intelligence in Fault Diagnosis and Signal Processing

Artificial Intelligence in Fault Diagnosis and Signal Processing

Guest Editors

Roque Alfredo Osornio-Rios
Athanasios Karlis
Andres Bustillo Iglesias

MDPI

Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Guest Editors
Roque Alfredo Osornio-Rios
Engineering Faculty
Autonomous University of Queretaro
San Juan del Rio
Mexico

Athanasios Karlis
Department of Electrical and Computer Engineering
Democritus University of Thrace
Xanthi
Greece

Andres Bustillo Iglesias
Department of Computer Engineering
University of Burgos
Burgos
Spain

Editorial Office
MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Special Issue, published open access by the journal *Applied Sciences* (ISSN 2076-3417), freely accessible at: https://www.mdpi.com/journal/applsci/special_issues/X0Q6C00HSX.

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

ISBN 978-3-7258-4397-8 (Hbk)
ISBN 978-3-7258-4398-5 (PDF)
https://doi.org/10.3390/books978-3-7258-4398-5

Cover image courtesy of Andres Bustillo

Contents

About the Editors

Roque Alfredo Osornio-Rios

Professor Roque Alfredo Osornio-Rios worked at the Faculty of Engineering of the Autonomous University of Queretaro (Mexico). His research interests were focused on signal processing in hardware and control. His passion for research resulted in an extensive scientific output, including over 80 articles in highly prestigious international journals, and recognition through various awards, such as the National Award for Technological Innovation in 2005 and the National Research Award in 2016. He passed away during this Special Issue period.

Athanasios Karlis

Athanasios Karlis received Dipl.-Eng. and Ph.D. degrees from the Electrical and Computer Engineering Department, Aristotle University of Thessaloniki, Greece, in 1991 and 1996, respectively. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece. He is also the Founder and an Advisor of the Democritus University of Thrace IEEE IAS SBC. His research interests include electrical machines and drives, renewable energy sources, and electrical power systems. He received the 2015 IEEE Outstanding Branch Chapter Advisor Award and the 2016 Outstanding IEEE IAS Student Branch Chapter Advisor Award.

Andres Bustillo

Professor Andres Bustillo works at the Computer Engineering Department of the University of Burgos (Spain). His research interests are focused on the applications of artificial intelligence and extended reality techniques to different fields, especially industry. He is Head of the XRAI-Lab at the University of Burgos, a laboratory focused on developing and validating Extended Reality applications to fields like psychology, industrial training and cultural heritage. His scientific work has resulted in authorship and co-authorship of over 80 publications in the last 15 years.

applied sciences

MDPI

Editorial

Artificial Intelligence in Fault Diagnosis and Signal Processing

Andres Bustillo [1,*] and Athanasios Karlis [2]

1 Department of Computer Engineering, Universidad de Burgos, 09006 Burgos, Spain
2 Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece; akarlis@ee.duth.gr
* Correspondence: abustillo@ubu.es

1. Introduction

Industry 4.0 has become a driving force in both the research and improvement of manufacturing processes within industrial environments. However, this term serves as an umbrella under which different technologies converge with a common goal: enhancing industrial conditions in terms of productivity, reliability, ecological impact, and worker's safety [1]. Within this umbrella is the integration of various advanced technologies, such as robotics, digitalization, the Internet of Things (IoT), extended reality (XR), 3D printing, artificial intelligence (AI), and digital twin (DT) technology [2]. Each of these advancements plays a distinct yet complementary role in the evolution of industrial automation. For example, robotics and 3D printing increase the automation of manufacturing processes, while others, such as IoT and AI, provide tools to manage and extract valuable insights from large volumes of data [3]. Additionally, some technologies focus directly on workers' safety and training, such as extended reality [4]. Although these technologies may seem disconnected at first glance, all of them play a major and complementary role in the common goal of industry automatization and advancement. For instance, the inclusion of biosensors in XR simulators to improve workers skills and their productivity will require the use of IoT to extract real data from their performance and AI tools to adapt the XR simulators in real time to the worker's emotional state and cognitive workload [5]. Digital twins (DTs) replicate physical assets in near-real-time, integrating real-world sensor data with advanced modeling for enhanced analysis. In fault prognostics, they aid by simulating expected operational behavior or potentially destructive cases, enabling early diagnostics and risk assessment. AI further benefits from the resulting data augmentation, improving predictive accuracy and decision-making [6].

To maximize industrial process reliability, the detection and diagnosis of faults is essential. Early fault detection in machinery will avoid damage in both critical machine components and workpieces. Additionally, in terms of industrial safety, this would facilitate safer operations, reducing the risk to plant workers. Therefore, the early detection and correct diagnosis of faults will facilitate decision-making that allows corrective actions to be taken to repair damaged components. In recent years, fault detection techniques, combining artificial intelligence and signal processing, have emerged to achieve this goal for different industrial processes, such as energy generation [7], machining [8], or chemical processes [9]. However, the topic continues to generate new trends [10] in methodologies related to multiple fault detection, novelty detection, data mining, hardware development, etc. Continuous advancements in this domain have led to the integration of more sophisticated methodologies that leverage domain-specific knowledge to enhance multi-fault classification and predictive maintenance capabilities. These developments highlight the increasing role of feature engineering and specialized knowledge in refining AI-driven

 https://doi.org/10.3390/app15073922

fault detection strategies, contributing to more accurate and computationally efficient solutions [11].

This Special Issue aims to compile novel research findings merging artificial intelligence and signal processing techniques to boost fault diagnosis. The selected works consider not only the development of new techniques but also their industrial implementation, with real-world industrial requirements in mind.

2. An Overview of Published Articles

This Special Issue compiles 15 research works and one review focusing on artificial intelligence and signal processing techniques applied to fault diagnosis in different industrial problems and environments.

In these works, different signals are acquired and processed. Signals extracted from most of the most widely used sensors are considered: linear strain gauges, temperature sensors, vibration and acoustic sensors, among others. These signals address a variety of real-world problems, such as data imbalance, management of missing data and outliers, data augmentation, pseudo-anomaly sample generation, and the creation of benchmark datasets.

In addition, different signal processing techniques and artificial intelligence techniques have been proposed as the most suitable solutions for modeling and extracting information from signals of different natures. Most systems combine time-series data analysis methods for preprocessing with supervised or unsupervised learning methods for system prediction. Some examples of the time-series data analysis methods for preprocessing proposed in these works are cluster-based segmentation or Mel-frequency cepstral coefficients, discrete wavelet transform, and hybridization of principal component analysis and kernel regression methods based on mutual information. The learning methods for system prediction proposed in these works are cutting-edge solutions, such as convolutional neural networks, extreme learning machine classifiers, the YOLOv8 deep neural network, deep support vector data description models, and one-dimensional convolutional neural networks.

Finally, this combination of artificial intelligence and signal processing techniques has been applied to different fault diagnosis cases. Some of these cases are related to high-end industries, such as the fault detection of next-generation chemical reaction devices, temperature-based warnings in high-speed trains, or defect detection in the manufacturing of laminated composites or printed circuit boards. Some others are focused on well-spread moving devices with a high impact on our society, such as real-time monitoring of conveyor belt damage, engine fault detection of traditional vehicles, fault detection of gears in machinery systems, or positional accuracy degradation in industrial robots. Lastly, others are focused on mechanically fixed structures, such as superficial defect detection for concrete bridges or submerged structure–foundation systems. In this way, this Special Issue covers a wide range of highly demanding applications.

To support this descriptive analysis, a bibliometric analysis was conducted on the 16 manuscripts included in this Special Issue. Figure 1 shows a cloud graph of the words included in the keywords of these works. Firstly, this figure outlines the orientation of all works with the Special Issue topic, being the words machine learning and those words related to fault diagnosis (predictive maintenance, faults diagnosis, and detection) the most relevant. Secondly, it includes words related to the most promising machine learning techniques for these purposes, such as deep learning, convolutional neural networks, or adversarial learning. Thirdly, it includes words about the main signals to be taken into consideration in this field, such as temperature, vibration, or acoustic. Fourthly, it includes words that highlight the main challenges of machine learning in this field of application, such as data imbalance, benchmark datasets, or cluster detection. Finally, it includes words related to the final application of these techniques to this research field, such as belt conveyors, circuit boards, or failure analysis. In summary, Figure 1 shows the convergence of published keywords in the works included in the Special Issue, by means of using different machine learning techniques and sensors' signals in different industrial problems but facing the same challenges and finding generalizable solutions.

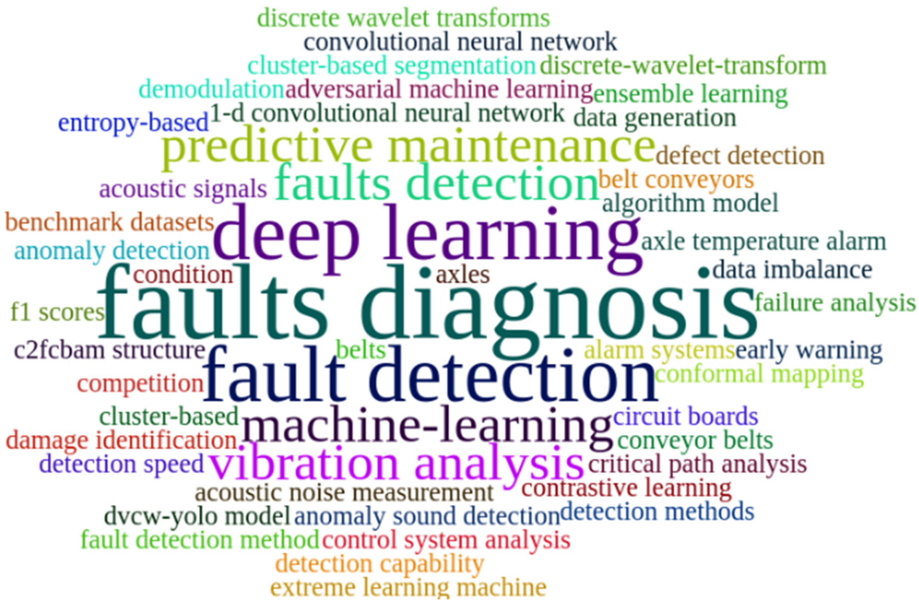


Figure 1. Cloud graph of most repeated keywords in the works included in this Special Issue.

3. Conclusions

Artificial intelligence techniques play a major role in fault diagnosis and signal processing of complex systems. These techniques help overcome intrinsic limitations in fault diagnosis, such as dataset imbalance, the scarcity of rare fault instances, signal deviations caused by component wear over time, and the lack of labeling processes due to time and cost constraints [12]. Addressing these challenges effectively and implementing these techniques in industrial settings will aid in the broader adoption of Industry 4.0 paradigms, ensuring a more sustainable future of manufacturing, energy generation, and goods and people transport [13]. This Special Issue includes 15 works that try to give some light to this exciting future.

Author Contributions: Conceptualization, A.B.; methodology, A.B.; formal analysis, A.B.; writing—original draft preparation, A.B.; writing—review and editing, A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Spanish Centro para el Desarrollo Tecnológico y la Innovación (CDTI), grant number MIG-20221059 (Project MHAYA).

Acknowledgments: This Special Section was firstly defined and promoted by Roque A. Osornio-Ríos, who passed away during this Special Issue period. We dedicate the memory of this Special Issue to him. Those of us who had the pleasure of knowing him saw a person who was approachable, optimistic, and full of energy—always finding common ground between cultures, ideas, and interests. He was warm and personable, with a ready smile and a sharp sense of humor that could captivate you and inspire you to embark on new scientific and personal journeys.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhong, R.Y.; Xu, X.; Klotz, E.; Newman, S.T. Intelligent Manufacturing in the Context of Industry 4.0: A Review. *Engineering* **2017**, *3*, 616–630. [CrossRef]
2. Ghobakhloo, M.; Mahdiraji, H.A.; Iranmanesh, M.; Jafari-Sadeghi, V. From Industry 4.0 Digital Manufacturing to Industry 5.0 Digital Society: A Roadmap Toward Human-Centric, Sustainable, and Resilient Production. *Inf. Syst. Front.* **2024**, 1–33. [CrossRef]
3. Ukwaththa, J.; Herath, S.; Meddage, D.P.P. A Review of Machine Learning (ML) and Explainable Artificial Intelligence (XAI) Methods in Additive Manufacturing (3D Printing). *Mater. Today Commun.* **2024**, *41*, 110294. [CrossRef]
4. Adriana Cárdenas-Robledo, L.; Hernández-Uribe, Ó.; Reta, C.; Antonio Cantoral-Ceballos, J. Extended Reality Applications in Industry 4.0.—A Systematic Literature Review. *Telemat. Inform.* **2022**, *73*, 101863. [CrossRef]
5. Guillen-Sanz, H.; Checa, D.; Miguel-Alonso, I.; Bustillo, A. A Systematic Review of Wearable Biosensor Usage in Immersive Virtual Reality Experiences. *Virtual Real.* **2024**, *28*, 74. [CrossRef]
6. Jeremiah, S.R.; El Azzaoui, A.; Xiong, N.N.; Park, J.H. A Comprehensive Survey of Digital Twins: Applications, Technologies and Security Challenges. *J. Syst. Archit.* **2024**, *151*, 103120. [CrossRef]
7. Balachandran, G.B.; Devisridhivyadharshini, M.; Ramachandran, M.E.; Santhiya, R. Comparative Investigation of Imaging Techniques, Pre-Processing and Visual Fault Diagnosis Using Artificial Intelligence Models for Solar Photovoltaic System—A Comprehensive Review. *Measurement* **2024**, *232*, 114683. [CrossRef]
8. Kounta, C.A.K.A.; Arnaud, L.; Kamsu-Foguem, B.; Tangara, F. Review of AI-Based Methods for Chatter Detection in Machining Based on Bibliometric Analysis. *Int. J. Adv. Manuf. Technol.* **2022**, *122*, 2161–2186. [CrossRef]
9. Mowbray, M.; Vallerio, M.; Perez-Galvan, C.; Zhang, D.; Del Rio Chanona, A.; Navarro-Brull, F.J. Industrial Data Science—A Review of Machine Learning Applications for Chemical and Process Industries. *React. Chem. Eng.* **2022**, *7*, 1471–1509. [CrossRef]
10. Abid, A.; Khan, M.T.; Iqbal, J. A Review on Fault Detection and Diagnosis Techniques: Basics and Beyond. *Artif. Intell. Rev.* **2021**, *54*, 3639–3664. [CrossRef]
11. Soori, M.; Jough, F.K.G.; Dastres, R.; Arezoo, B. AI-Based Decision Support Systems in Industry 4.0, A Review. *J. Econ. Technol.* **2024**. [CrossRef]
12. Ramírez-Sanz, J.M.; Maestro-Prieto, J.-A.; Arnaiz-González, Á.; Bustillo, A. Semi-Supervised Learning for Industrial Fault Detection and Diagnosis: A Systemic Review. *ISA Trans.* **2023**, *143*, 255–270. [CrossRef]
13. Jayashree, S.; Reza, M.N.H.; Mohiuddin, M. Impact of Cleaner Production and Environmental Management Systems on Sustainability: The Moderating Role of Industry 4.0. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *795*, 012013. [CrossRef]

applied sciences

MDPI

Article

An Enhanced Contrastive Ensemble Learning Method for Anomaly Sound Detection

Jingneng Liao [1,2], Fei Yang [1,2,*] and Xiaoqing Lu [1,2]

[1] Hubei Key Laboratory of Power Equipment & System Security for Intergrated Energy, Wuhan University, Wuhan 430072, China; chingneng.liao@whu.edu.cn (J.L.); luxiaoqing2012@hotmail.com (X.L.)
[2] School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China
* Correspondence: f.yang@whu.edu.cn

Abstract: This paper proposes an enhanced contrastive ensemble learning method for anomaly sound detection. The proposed method achieves approximately 6% in the AUC metric in some categories and achieves state-of-the-art performance among self-supervised models on multiple benchmark datasets. The proposed method is effective in automatically monitoring the operating conditions of the production equipment by detecting the sounds emitted by the machine, to provide an early warning of potential production accidents. This method can significantly reduce industrial monitoring costs and increase monitoring efficiency to improve manufacturing facility productivity effectively. Existing detection methods face challenges with data imbalance caused by the scarcity of anomalous samples, leading to performance degradation. This paper proposes an enhanced data augmentation method that improves model robustness by allowing the data to retain the original features while adding noise close to the real environment through a simple operation. Secondly, model feature extraction is enhanced by using channel attention to fuse time-frequency features. Thirdly, this paper proposes a simple anomaly sample generation method, which can automatically generate real pseudo anomaly samples to help the model gain anomaly detection capability and reduce the impact of data imbalance. Finally, this paper proposes a statistical-based bias compensation that further mitigates the impact of data imbalance by distributing samples through statistical induction. Experimental verification confirms that these changes enhance anomalous sound detection capability.

Keywords: anomaly sound detection; deep learning; self-supervised learning; fault diagnosis

1. Introduction

Anomaly sound detection (ASD) is an algorithm used to ascertain whether the sound emitted by an object under test is in an abnormal state. This algorithm is widely applied in various fields, particularly in the industrial manufacturing systems [1,2]. Machine equipment generates sounds during operation, which reflect its current working status. Utilizing ASD algorithms can effectively monitor the operational conditions of the machines being tested and issue early warnings for potential anomalies, thereby preemptively averting potential production accidents and machine failures. However, in reality, it is often challenging to obtain a large number of valid anomaly sound samples. Consequently, anomaly sound detection tasks frequently encounter the issue of extreme data imbalance. This imbalance also leads to a blurred definition boundary between normal and anomaly samples, which will affect the accuracy of anomaly sound detection methods.

ASD generally requires sufficient positive and negative samples. However, in the real-world scenarios, collecting comprehensive and representative negative samples can be challenging due to their inherent diversity and scarcity. Consequently, many existing studies exploit only positive samples instead [3,4]. One of the most common methods is based on statistics, such as the outlier detection [5], K-means [6] and density-based detection method [7]. This method assesses whether the input sample is anomalous by measuring the distance from the observation to the centroid of each sample cluster. In recent decades, classifier-based methods have become a trend. For example, deep learning-based methods [8,9] train a neural network to find the relationships between the sound representations and the equipment status. A classifier is then used to detect anomalies. What is more, reconstruction-based methods can also be applied to anomaly detection, such as autoencoder [10] and Generative Adversarial Networks (GANs) [11]. This method performs anomaly detection by calculating the reconstruction error between the reconstructed sample and the input sample. After training, the model can only accurately reconstruct and map normal samples. Therefore, abnormal samples often cannot be reconstructed properly, resulting in a larger reconstruction error.

Although these methods are effective, they still have many drawbacks. For example, statistical-based methods directly adopt feature reduction techniques, which may fail to accurately capture the distribution of complex real-life data. This can lead to excessively large clusters, thereby compromising the anomaly detection capability. Similarly, reconstruction-based methods are also struggle with complex and high-dimensional data, making it difficult to efficiently distinguish between normal and abnormal samples when determining the decision boundary, which affects the model's judgment results. While GANs can mitigate this shortcoming, most GAN models have strict requirements for training samples, and the performance of the generator and discriminator affect each other. These issues also contribute to poor model performance. With the development of deep learning networks, using complex neural network models can effectively fit high-dimensional data to obtain a more accurate feature distribution. However, in reality, it is often difficult to collect a large and diverse set of abnormal samples. This results in complex neural network models performing anomaly detection in the absence of abnormal samples, leading to a significant reduction in model performance as the model is unable to understand the distribution of abnormal samples and potentially causing overfitting. Additionally, due to data imbalance, the model may develop bias, leading to degraded detection performance or even model collapse, resulting in trivial solutions. Although numerous methods [12,13] exist to address the bias problem in neural networks caused by data imbalance, most of these methods are based on federated learning. Therefore, when applied in real manufacturing environments, they are constantly faced with communication and device heterogeneity issues, while these methods may be difficult to apply in large-scale industrial deployment environments.

Therefore, this paper proposes a novel self-supervised learning approach. The proposed network is not only able to learn the feature representations of normal samples while mitigating the bias arising from the imbalanced dataset, but it is also able to swiftly generate effective abnormal samples and attend to learn the feature distribution of the potentially real-life abnormal samples. The detailed contributions are as follows. Firstly, this paper employs a novel improved spectrogram augmentation method and proposes a cross-domain data representation approach specifically for original audio format data, encompassing features in the time domain, frequency domain and statistical domain. Secondly, this paper proposes an improved novel self-supervised learning network for anomaly sound detection tasks. After being tested on multiple datasets, the model demonstrates excellent performance. Lastly, this paper proposes a simple method for generating anomaly samples, which can effectively assist the model in learning the features of potential real-life anomaly samples.

2. Materials and Methods

2.1. Motivation

As neural networks continue to evolve, neural network-based deep learning methods have increasingly become the primary approach for ASD. However, in real-life industrial manufacturing environments, it is difficult to obtain accurate data labels, making it difficult for common deep learning approaches to perform. This results in the formulation of a solution based on a self-supervised approach. The self-supervised learning process enables the model to effectively capture the global feature distribution. Self-supervised learning methods spontaneously mine feature information based on auxiliary tasks through the construction of these tasks, and generalize this to general features, thereby accomplishing the modeling of the model's global feature distribution. In the task of ASD, common self-supervised learning approaches employ auxiliary classification methods or contrastive learning techniques. For example, reference [14] collects phase and amplitude features from the data, utilizing a self-supervised classification network and attention method for auxiliary classification, thereby enabling the model to acquire feature modeling of normal sample data and realize the capability of abnormal sound detection; reference [15] employs a flow-based density estimation model, integrating self-supervised classification methods with unsupervised density likelihood estimation approaches to enhance the model's capability for ASD; reference [16] employs contrastive learning method by calculating the cosine angle between positive and negative samples, enabling the model to simultaneously learn the distributions of normal and abnormal features.

Although these techniques may prove effective in the detection of anomalies, the resulting precision may be compromised. This is because the issue of neural network bias, which is caused by the imbalance in the amount of data, is not taken into account in the feature representation. Furthermore, the problem of model collapse, which occurs in the absence of valid anomaly samples, is also overlooked. Consequently, this paper presents a novel model that addresses these issues. It is capable of gathering more comprehensive feature data, rectifying the bias issue inherent to neural networks, and performing multi-domain fusion and adaptive weighting of multi-domain information. Furthermore, it can quickly and effectively generate high-quality anomaly samples during the training process, thereby enhancing the model's overall anomaly detection capabilities. The model has been subjected to extensive experimentation with multiple datasets, and a comparative analysis of the experimental results has demonstrated its efficacy in anomaly detection and generalization.

2.2. Data Processing

To enhance the performance of the model in capturing the features of the original data, this paper implements multi-domain data processing in the source audio data, encompassing not only the time domain but also the frequency domain and statistical domain. This enables the model to obtain a more comprehensive representation of the data in low dimensions, facilitating a more nuanced and comprehensive understanding of the data. In this paper, the log-mel transform [17] is employed for frequency domain feature analysis and extracts the log-mel power spectrogram and the log-mel magnitude spectrogram from the input audio signals. In contrast, the Mel spectrogram is used to represent the features of the statistical domain.

2.3. Data Augmentation

In order to improve the model's generalization ability, this paper used a mixup [18] based on the audio clipping augmentation approach. For the spectral feature maps, the target spectrograms were randomly cropped at the top left and bottom right, and the clipped portion was spliced with other spectrograms, while mixup enhancement was used to increase the diversity of the data and reduce model overfitting. The procedure is shown in Figure 1. Compared to other cropping augmentation approaches such as cutout [19], the method proposed in this paper ensures the possession of both original and interference features at every time pin and every frequency value. In order to maximize the introduction of interference factors without significantly disturbing the original feature information, the cutting points on the frequency domain axis and the time domain axis are designed to be complementary during the shearing process. The methodology is as follows:

$$
\begin{aligned}
x &\in (1, \frac{3}{4}F), \\
y &\in (1, \frac{3}{4}T),
\end{aligned} \tag{1}
$$

where F represents the frequency domain axis length, T represents the time domain axis length, x and y represent the random cropping points on their respective axes, respectively.

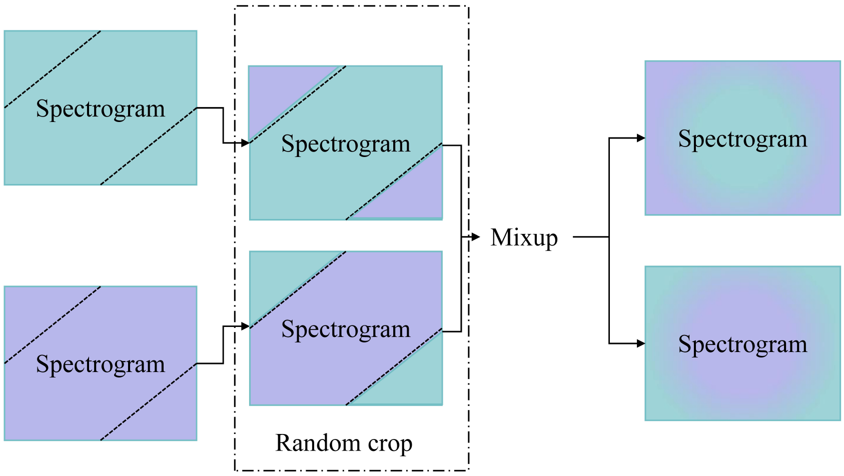


Figure 1. Spec1 and Spec2 represent two different spectrograms, and the cropping points for each axis in each spectrogram are complementary.

Subsequently, a mixup-based augmentation was conducted on the two cropped spectrograms, utilizing a mixing ratio factor to combine the two inputs and the corresponding labels in a specified ratio. The mixup-based augmentation is as follows:

$$
\begin{aligned}
x &= (1-\lambda)x_1 + \lambda x_2, \\
y &= (1-\lambda)y_1 + \lambda y_2,
\end{aligned} \tag{2}
$$

where x_1, x_2 refers to input samples and y_1, y_2 refers to their corresponding labels. x, y refer to the mixed sample and its label.

2.4. Contrastive-Based Spectrogram Domain Model

For the raw data, this paper transforms the signal data into log-mel power spectrogram and log-mel energy spectrogram via log-mel transformation [17], and then concatenates these two spectrograms via channel dimension, which enhances information features of frequency domain at minimal cost.

After that, this paper designs an improved contrastive learning-based model. The structure is depicted in Figure 2. This model is divided into a pre-training feature contrastive training and anomalous samples fine-tuning model. The aim of the pre-training process is to allow the model to gain feature extraction capability for each category of sound frequency domain data by increasing the similarity of two samples of the same category. The purpose of the fine-tuning process is to generate anomaly samples through the anomaly sample generator to give the model anomaly detection capability and to improve the fit of the anomaly decision boundary.

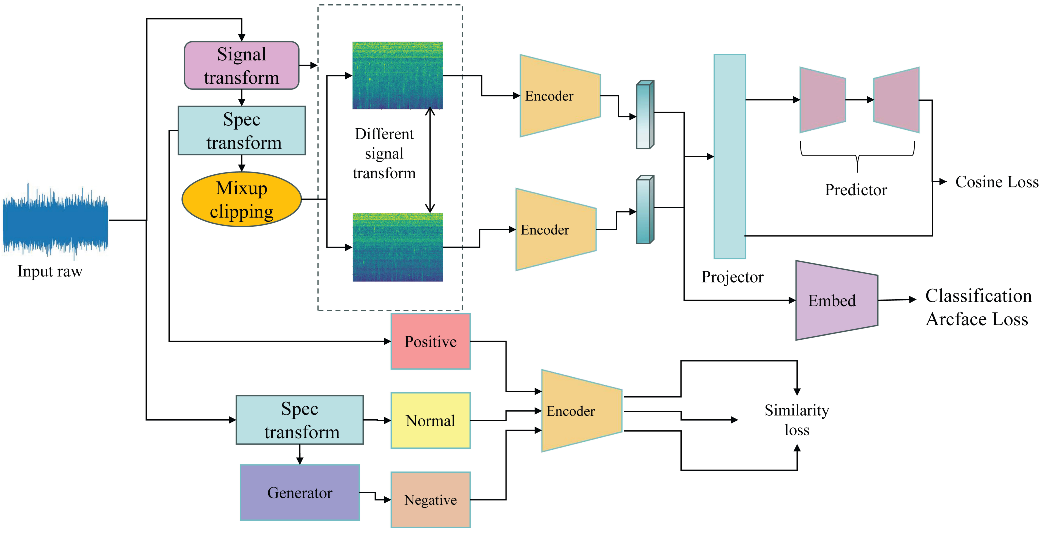


Figure 2. Model structure: The model is divided in two sections, the upper section is the pre-trained model and the lower is the fine-tuned model.

2.4.1. Pre-Train Progress

During the pre-training progress, a multi-task learning approach is used to improve model performance and help the model to converge. The pre-training structure is improved from the Simple Siamese network (Simsiam) [20]. This paper not only calculates the cosine similarity, but also employs a self-supervised classification task to further enhance the model feature representation capabilities. This paper employs GhostnetV2 [21] model as the feature embedding network which can not only accurately capture features but also reduce the number of model parameters, thereby enhancing computational speed. For this backbone network, this paper only uses the front feature extraction convolutional layer without using the last fully connected classification layer. Moreover, to further improve the model generalization ability and data diversity, this paper randomly employs signal transformations such as time shift, pitch shift, fade in/out and add noise. The time shift transformation shifts the audio signal either forward or backward. In this paper, the time shift rate is chosen from the range of 0–56. The pitch shift transformation shifts the pitch of the waveform. In this paper, the step to shift the waveform is set to -20. The fade in/out transformation adds a fade in or fade out to a waveform at its beginning or end. This paper uses two types of fading: exponential and half sinusoidal. And adding noise transformation inserts a white Gaussian noise in the waveform. In this paper, the signal-to-noise ratio (SNR) is chosen in the range of $[-10, 10]$. After applying these transformations, the input spectrogram maps are then fed into the Simsiam structure. By maximizing the similarity of the same class samples, the backbone model can efficiently model sample features. Moreover, in order to improve the model performance better, this paper designs a self-supervised classification task for this backbone model.

For the contrastive learning pre-trained model, the total cosine similarity can be calculated as follows:

$$L = \frac{1}{2}D(p_1, z_2) + \frac{1}{2}D(p_2, z_1),$$
$$D(d_1, d_2) = -\frac{d_1}{\|d_1\|_2} \cdot \frac{d_2}{\|d_2\|_2}, \tag{3}$$

where p and z represent two feature vectors, and D represents the cosine similarity function. However, due to the lack of different class samples, the model will collapse easily during the training progress. Therefore, to avoid this problem, a stop-gradient operation [20] is imposed on one of the feature vectors. So the final cosine similarity loss functions is reformulated as follows:

$$L = \frac{1}{2}D(p_1, stopgrad(z_2)) + \frac{1}{2}D(p_2, stopgrad(z_1)), \tag{4}$$

After the contrastive learning progress, the backbone is trained with an auxiliary classification task. In the auxiliary classification task, the feature vectors of the input spectrograms from the Siamese network are fed into the embedding head, which outputs the predicted label values. Subsequently, the ArcFace [22] loss function is utilized to calculate the loss. The arcrface loss functions is as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s(cos(\theta_{y_i}+m))}}{e^{s(cos(\theta_{y_i}+m))} + \sum_{k=1,k\neq y_i}^{K} e^{scos\theta_k}}, \tag{5}$$

where N represents the batch size number, s represents the scale parameter, m represents the margin parameter, θ_{y_i} represents the angle value of the target category corresponding to the sample y_i, θ_k represents the angle value of the non-target category corresponding to sample y_i and K represents the number of non-target categories.

2.4.2. Fine-Tune Progress

After pre-training, the backbone model is immediately fine-tuned. This paper employs a simple novel negative generator which can generate pseudo-negative samples from normal samples. This paper uses two simple methods to generate fake samples. The first is Spectrogram Flip (SF) and the second is Spectrogram Plus (SP). The pseudo-anomaly sample example is shown in Figure 3.

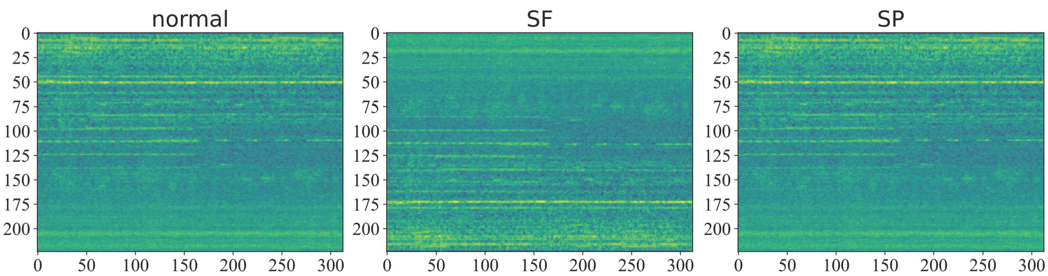


Figure 3. Negative examples: SF denotes spectrogram flip, SP denotes spectrogram plus. The color represents the energy at each sampling point, with brighter colors indicating higher energy levels.

SF: This paper inverted the top and bottom for the input spectrogram. This operation allows the normal information highlighted in the normal sample to be overwritten. For example, in such cases wherein the regular data are a high-frequency sound, the corresponding anomalous data may be a low-frequency sound. Flipping can then be applied to generate negative samples that are similar to the real negative samples. This is achieved by shifting the energy from the high-frequency portion of the sample to the low-frequency portion. As a result, the low-frequency portion becomes more prominent than the high-frequency portion, and the energy of the high-frequency portion is displaced by the energy of the low-frequency portion. As a result, the flipped sample can be considered as a real negative sample.

SP: This paper performed a stacking operation for the input spectrogram. Within the spectrograms, the normal features of the sound is often prominent, while the environmental noise features are relatively inconspicuous. During the stacking process, the original sound features may exceed their original normal range due to multiple stacking, leading to over-noise anomalies and the formation of noise. Furthermore, after stacking, some of the originally weak noise features may replace the original sound features, causing the model's focus to shift from normal features to environmental noise features.

After generating pseudo samples, in order to help the model achieve precise anomaly detection boundaries while improving its generalization capability, this paper establishes three sample clusters. Among them, the original samples are designated as the main anchor cluster, the samples generated by the generator form the anomaly cluster and the samples enhanced through signal augmentation constitute the generalization cluster. After generating three different categories of clusters, this paper then employs an improved similarity loss function to train the encoder backbone. The improved loss function takes into account both cosine angle and Euclidean distance, and this loss function can adequately generalize the separation of vector angles and spatial distances between clusters with different attributes which can improve the model performance. The improved loss function is as follows:

$$
\begin{aligned}
L_{cos} &= -D(x_{anchor}, x_{pos}) + D(x_{anchor}, x_{neg}), \\
L_{dis} &= \|x_{anchor} - x_{pos}\|_2{}^2 - \|x_{anchor} - x_{neg}\|_2{}^2 + m, \\
L_{loss} &= \frac{1}{N}\sum_{1}^{N}(L_{dis} + \alpha)\exp^{(sL_{cos}+\beta)},
\end{aligned}
\tag{6}
$$

where α, β and s are three scale parameters, m represents a distance margin, x_{anchor} represents the original anchor sample, x_{pos} represents the augmented positive sample against the anchor sample and x_{neg} represents the negative anomaly sample against the anchor sample. In this paper, m is set to 0.05, s is set to 1, α is set to 0.01 and β is set to 0.005.

2.5. Time Domain Model

Although spectrogram-based networks are capable of effectively extracting features, due to the nature of convolution calculations, spectral-based network models tend to capture only frequency domain features while neglecting the causality and correlation of time domain features. In order to fully represent the features of the input data, this paper used not only the frequency features, but also the time domain features. This paper designs a time domain CNN model, the model framework of which is summarized in Table 1. In this model, the first convolutional layer takes a convolutional kernel of size 256. The reason for using a large convolutional kernel is to take into account the fact that the sampling frequency of the input samples is 16KHz, so a small kernel size will not only affect the response speed of the model, but also reduce the global feature extraction ability of the shallow network. The model then utilizes multiple blocks for subsequent feature extraction. The structure of the block is shown in Figure 4. In each block, all the layers employ depthwise separable convolutions, which can ensure feature extraction capabilities while simultaneously reducing computational load and enhancing computational speed.

Table 1. Temporal model structure: The block contains three convolutional blocks which form an inverted residue module. In each block, *t* represents intermediate dimension scale factor of the inverted residual module of the block, *c* represents output dimension of each layer and *s* stands for the stride size of each layer

Input	Operator	Parameters		
		t	c	s
1 × 160,000	Conv1d (k = 256)	-	16	64
1 × 1250	block	3	32	1
1 × 1250	block	3	32	2
1 × 625	block	3	32	2
1 × 313	block	2	64	3
1 × 105	block	2	128	2
1 × 53	block	2	256	1
1 × 53	block	2	256	2
1 × 27	block	2	256	2
1 × 14	block	2	256	2
1 × 7	block	2	256	1
1 × 7	Avgpool (k = 7)	-	512	1

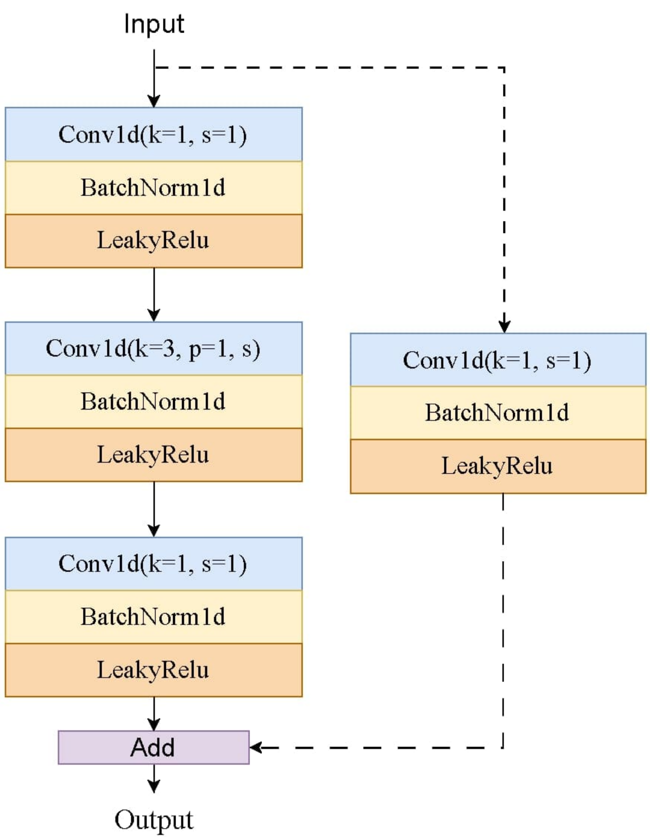


Figure 4. Block structure: This block contains three convolutional sub-blocks which together form an inverted residual module. In each inverted residual module, there is a dimension-increasing layer with a 1 × 1 convolution, a feature extraction layer with a 3 × 3 convolution and a dimension-decreasing layer with a 1 × 1 convolution. After each convolutional kernel, batchnorm and leakyrelu activation functions are employed. When stride is set to one, the block uses shortcut connection.

2.6. Domain Fusion Model

In order to be able to unite the time-frequency domain features, this paper ensembles the feature extraction backbone networks from the frequency domain and time domain models. The fusion method is illustrated in Figure 5. The fusion model first concatenates the feature vectors from both time and frequency domains along the channel dimension. Subsequently, the concatenated feature vector is fed into a classifier. This classifier employs multiple Squeeze-and-Excitation Blocks (SEBlock) [23] to perform channel-wise adaptive weighting on the input features. Finally, for the classification prediction output, the ArcFace function is used to calculate the loss value for model convergence. During the training process, neither the time domain model nor the frequency domain model participates in gradient updates; only the classifier participates in the back-propagation process. The aim is to ensure that the feature representation capabilities of the two models remain unchanged, thereby guaranteeing optimal feature modeling capability. In this paper, the channel attention-based fusion method is employed instead of directly utilizing time-frequency features. This is because the direct utilization of time-frequency domain features may not be able to specifically represent the feature distribution of the input samples. By using the channel attention mechanism, the model can automatically extract the correlation between each feature channel and automatically weight the channels according to the degree of importance, which can help the model focus on feature maps that are more critical to the target task than directly extracting features from the time-frequency domain, thus improving the model performance. The overall fusion classifier structure is shown in Table 2.

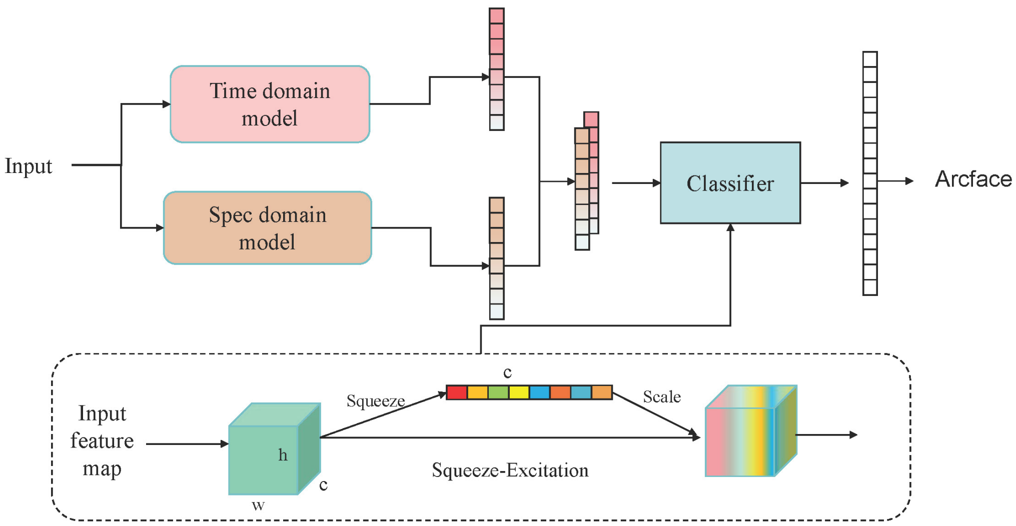


Figure 5. Time-Spec domain fusion model: w and h represent the shape of feature map, c represents the number of channels. During Squeeze-Excitation, the model first squeezes the feature map along the channel dimension, then derives weights for each channel and finally applies these weights to the corresponding channels of the original map.

Table 2. The fusion classifier structure: FC represents the fully connected layer, layer represents the convolutional layer and SEBlock represents the channel attention block. The batchnorm is used in each convolutional layer, and the PRELU activation function is used in the first two convolutional layers. In each SEBlock, the mid-channel factor is set to 0.5.

Layer	In Channel	Out Channel	Kernel Size
layer1	2	16	1
SEBlock	16	16	1
layer2	16	64	3
SEBlock	64	64	1
layer3	64	256	3
layer4	256	512	6
layer5	512	128	1
FC	128	16	-

2.7. Statistics Domain Model

In most anomaly detection tasks, models often encounter the issue of severe data imbalance. Even with the introduction of generators, it remains difficult to mitigate the model bias caused by the imbalance in the original data, which subsequently degrades the model's performance. Therefore, to address this issue, this paper proposes a method of using statistically weighted average frequency-domain features to correct feature bias. For the spectrogram of the input data, this paper calculates the global average value of each frequency along the time axis. Finally, a global weighting is applied based on the magnitude of each frequency's average value, meaning that frequencies with larger average values will receive greater weights, while those with smaller average values will receive smaller weights. Compared to statistical feature representation methods based on extreme values, the method used in this paper not only reflects the average performance of each frequency across the entire time axis but also highlights the importance of different frequencies based on their weighted weights. The average frequency index can be calculated using the following formula:

$$y_i = \frac{1}{N}\sum_{j=1}^{N} x_j, \tag{7}$$

where N represents the batch size number, i represents the frequency bin and x_j represents each sample bin. The global weighting factors are calculated as follows:

$$\lambda_i = \frac{y_i}{\sum_{f=1}^{M} y_f}, i \in (1, M), \tag{8}$$

where i represents the frequency bin, M represents the total frequency bin numbers and y_i represents the average frequency index of the current current frequency. The full process is shown in Figure 6.

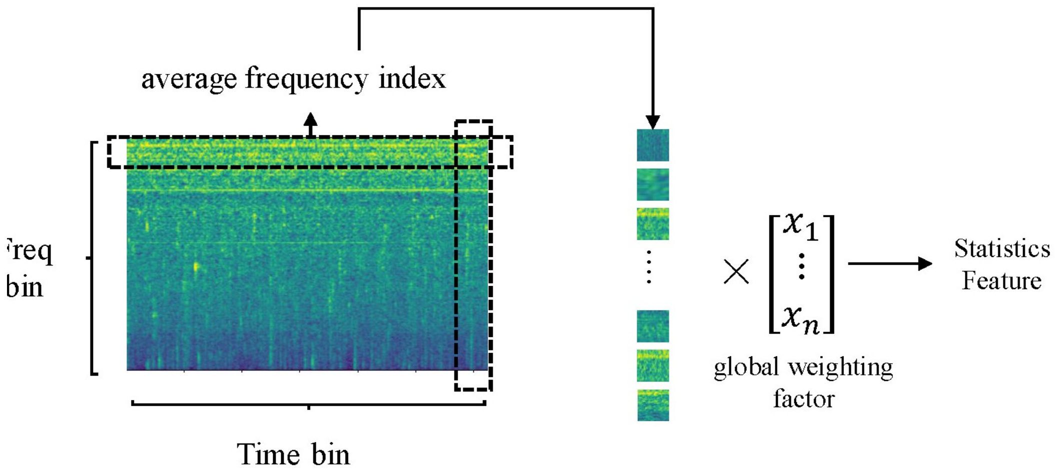


Figure 6. Global weight average pooling process.

2.8. Ensemble Evaluator Model

This paper employs an ensemble approach for anomaly assessment on the input data. The anomaly scores are obtained through a Gaussian Mixture Model (GMM), which combines multiple Gaussian distributions. By adjusting the parameters of the GMM, the model can approximate almost any data distribution. After evaluating both the statistical domain model and the time-frequency domain fusion mode, an ensemble method is utilized to obtain the final evaluation score. The calculation method is described as follows:

$$Score = \sum_{i=1}^{2} \lambda_i GMM_i, \tag{9}$$

where λ is weight parameter and i represents the i-th model. In this paper, the weight for the fusion model is set to 0.9, with the aim of fulling leveraging the discriminative performance of the model. For the statistical domain model, the weight is assigned as 0.1. This is because the role of the statistical model is merely to introduce statistical features and reduce the bias caused by data imbalance. Therefore, only a partial detection capability of this model is needed. Although this paper applies fixed discriminant weights to the two models, several tests indicate that better model performance can be achieved by adjusting the weights according to the characteristics of different test objects. However, since the interference noise in the original data is not adequately considered when using statistical factors, if the statistical model assigns a larger weight to some categories of data, it will instead reduce the model's detection ability.

3. Results

3.1. Dataset

To fully test the capability of the proposed model, the MIMII dataset [24], DCASE2020 task2 dataset and MIMII DUE dataset [25] are used for validation experiments. In ASD tasks, the primary official evaluation metric is the Area Under the Curve (AUC) value, which reflects the area under the ROC curve. In the MIMII dataset and the DCASE2020 task2 dataset, the primary focus of testing was the model's ability to detect anomalies without training on anomalous samples. In the MIMII DUE dataset, the main objective was to evaluate the model's capability for transfer anomaly sound detection without applying domain adaptation, thereby assessing the model's robustness and generalization ability. In the MIMII dataset, there are only four different categories. In the DCASE2020 task2 dataset, it contains the MIMII dataset and Toyadmos dataset, with six different categories. In the MIMII DUE dataset, it contains five different categories. Within each category, the normal data for the source domain and a small amount of data for the target domain are included.

In these datasets, the machine objects included are common types of machines found in industrial environments, such as slide rails, pumps, fans and so on. In the training set, only normal samples are included. In each training sample, the sound data are captured in a realistic manufacturing environment, which means that each audio datum contains not only the primary feature data of the machine being evaluated but also numerous disturbing noises from the surrounding environment, including the disruptive sounds of other industrial machines and parts of human voices. In the test set, the data consist of normal and abnormal sound samples, which are also derived from the real manufacturing system environment and covers a number of different types of anomalies in order to simulate the manufacturing environment under real-life conditions. In addition to this, in some test sets, to further satisfy the reality of the manufacturing environment, the test data also include the sound of the manufacturing machine under different production conditions, such as different temperatures, humidity and weather. Therefore, by considering these characteristics, these benchmark datasets are selected in this paper to validate the performance of the proposed model.

3.2. Implementation

During the data mixup augmentation, the mixing ratio factor is set to 0.4. In the process of spectrogram transformation, the number of Mel filters is set to 224, the hop length is set to 512 and the number of FFT points is set to 2048.

For the frequency domain network model, during the pre-training stage, the number of training epochs is 50, with a batch size of 128. The backbone network employs the SGD optimizer, with an initial learning rate of 0.05, a weight decay factor of 0.005 and a momentum factor of 0.9. The auxiliary network uses the AdamW optimizer [26] with an initial learning rate of 0.01. Additionally, to ensure good convergence stability, this paper applies cosine decay specifically for this network. During the fine-tuning comparison stage, the AdamW optimizer is used with an initial learning rate of 0.001, a weight decay factor of 0.0005, a batch size of 64 and a number of training epochs of 50.

For the time domain network model, the AdamW optimizer is used with an initial learning rate of 0.005, a weight decay factor of 0.0005, a batch size of 128 and a number of training epochs of 200.

For the time-frequency domain hybrid model, the SGD optimizer is used with a learning rate of 0.05, a weight decay factor of 0.005, a momentum factor of 0.9, a batch size of 128 and a number of training epochs of 20.

For the two GMM estimators, the number of mixing components is set to 1 and the number of EM iterations is set to 1000.

3.3. Experiment Results

The AUC test results for the model are presented in Table 3 below. Furthermore, to fully demonstrate the performance of the proposed model in this paper, we also calculated the partial-AUC (pAUC) value and recall score within the MIMII dataset and plotted the FAR-FRR curve. In this paper, the pAUC is calculated as the AUC over a low false-positive rate p, as this score reflects the reliability of the model. In this paper, the p is set to 0.1. The pAUC score and recall scores are shown in Table 4, and the FAR-FRR curve is shown in Figure 7. FAR represents False Acceptance Rate and FRR represents False Rejection Rate, while FRR is the likelihood that a legitimate user will be rejected by the system, and FAR is the likelihood that an impostor will be accepted by the system. In the evaluation of any classification model, the FAR-FRR curve can represent the model performance.

Table 3. AUC scores in three datasets: Since most open-source papers and code evaluation metrics only provide the AUC value, the AUC is primarily used as the main indicator for performance comparison. Meanwhile, to ensure fairness, all models only used the development dataset to train.

Dataset	Algorithm	Fan	Pump	Slider	Valve	Toycar	Toyconveyor	Gearbox
MIMII	AE [27]	63.24%	61.92%	66.74%	53.41%			
	IDNN [28]	64.64%	61.48%	69.8%	59.37%			
	MobilenetV2 [27]	80.61%	83.23%	96.26%	91.26%			
	AADCL [16]	80.11%	70.12%	77.43%	84.17%			
	SLFE-AE [29]	80.0%	88.5%	95.4%	93.1%			
	Complex-network [14]	89.55%	**96.4%**	98.75%	96.87%			
	This research	**95.56%**	93.56%	**98.86%**	**98.75%**			
DCASE2020	Self-encoder [30]	69.92%	59.57%	95.79%	94.31%	92.74%	80.85%	
	Reference [31]	59.64%	60.32%	76.29%	85.42%	79.30%	61.72%	
	Glow-aff [15]	75.90%	83.40%	94.60%	91.40%	92.20%	71.50%	
	STGram-MFN [32]	80.14%	80.4%	**97.09%**	82.5%	91.80%	70.90%	
	Reference [33]	90.07%	86.44%	91.84 %	**97.67%**	89.84%	67.64%	
	LMTnet [34]	88.81%	87.83%	94.27%	91.04%	90.05%	62.45%	
	This research	**94.47%**	**91.93%**	95.79%	95.31%	**92.74%**	**79.85%**	
MIMII DUE	AE [35]	64.36%	63.66%	58.09%	52.7%			66.7%
	MobilenetV2 [35]	63.65%	64.12%	64.62%	54.01%			68.4%
	UADA [36]	65.31%	62.32%	59.19%	**69.05%**			**71.05%**
	This research	**68.72%**	**64.62%**	**72.72%**	65.98%			70.07%

Table 4. The proposed model exhibits a high pAUC value and recall rate, which demonstrates its effectiveness in accurately detecting abnormal samples while avoiding the misclassification of positive samples. These results indicate the high reliability of the proposed model.

	Fan	**Pump**	**Slider**	**Valve**
pAUC	87.27%	78.91%	96.53%	96.3%
Recall	89.85%	86.25%	97.5%	97.75%

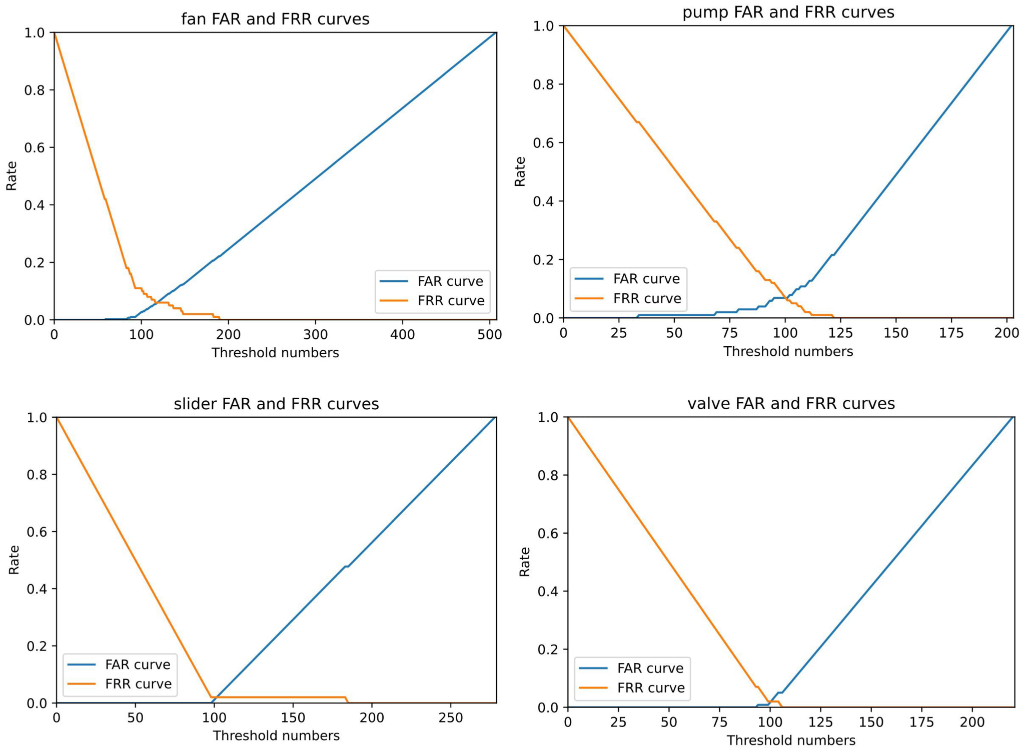


Figure 7. FAR-FRR curve within the MIMII datset.

3.4. Ablation Experiment

Meanwhile, to validate the positive effects of the time-frequency domain fusion model, this paper conducted experiments on the fusion model, with the experimental results presented in the following Table 5.

Table 5. AUC scores with/out fusion model within the MIMII dataset.

	Fan	**Pump**	**Slider**	**Valve**
With	**95.56%**	**93.56%**	**98.86%**	98.75%
Without	94.9%	91.57%	98.83%	**98.79%**

To verify the impact of models based on statistical features, this paper also compares the AUC results within the MIMII dataset when statistical features are utilized. The results are shown in Table 6.

Table 6. AUC scores with/out statistical features within the MIMII dataset.

	Fan	**Pump**	**Slider**	**Valve**
With	**95.56%**	**93.56%**	**98.86%**	**98.75%**
Without	93.38%	92.56%	94.86%	96.37%

4. Discussion

Based on the experimental results, the proposed model exhibits superior performance compared to traditional classification models and reconstruction methods. Within the MIMII dataset, compared with other existing methods, the test AUC results of the proposed method remained stable among the four machine categories, with the maximum difference in AUC within 5.1% while the other methods could reach from 9.2% to 15.4%. Based on the data of the fan, the proposed method is able to achieve the highest increase in AUC value. In the DCASE2020 task2 dataset, the model also exhibits excellent anomaly detection performance. In the MIMII DUE dataset, the proposed model can effectively perform anomaly detection under domain shift without using any domain adaptation operations and outperform several baseline models.

Meanwhile, based on the ablation results, it can be observed that the model utilizing the fusion method achieves better performance in terms of detection accuracy for certain categories. This is attributed to the model allocating greater weight to feature channels with more prominent feature expressions, thereby effectively enhancing the model's overall performance.

Based on the results of the above multiple experiments, it can be demonstrated that the proposed model exhibits excellent anomaly sound detection performance under multiple public datasets. The proposed time-frequency domain fusion model and statistical feature model can effectively conduct abnormal detection without being trained with real abnormal samples in reality, and avoid the model bias problem caused by data imbalance. Moreover, by learning the pseudo-anomaly samples generated by the abnormal sample generator, the model can also spontaneously explore the feature distribution of potential real anomaly samples. At the same time, considering the resource constraints that may exist in real manufacturing environments, the proposed model adopts multiple lightweight design approaches that enable it to resolve potential resource conflicts.

Despite the excellent ASD performance of the proposed model, there are still some issues that need to be addressed. For example, the proposed model employs a statistical-based approach to counteract the effect of data imbalance, but based on the result of the ablation experiment, it can be found that the approach is not effective in providing accurate feature induction information to help the model significantly improve its performance when facing some of the non-stationary signals. Meanwhile, since the input samples may contain noise information, the use of the statistical approach may surround too much disturbing information and affect the model performance instead. Furthermore, taking into account the influences of temperature and humidity on sound and production machines, when the manufacturing system is subjected to drastic weather variations, the informative features of the sound are changed, thereby leading to a decline in the model's performance.

Even though the statistical model is employed without fully accounting for the noise in the data, the issue of the feature space distribution becoming overly concentrated is seen as the data expand.However, the proposed method employs an ensemble discriminant-based approach in the inference stage, and the weight assigned to the statistical model is considerably lower than that of the neural network model. Therefore, although noise can affect the modeling capability of the statistical model, it does not affect the overall detection performance of the ensemble model.

5. Conclusions

This paper proposed an enhanced contrastive ensemble learning model for anomaly sound detection. Firstly, this paper proposes a new data augmentation method, which can effectively preserve the original audio feature information while adding diverse noises to fully simulate the disturbances in the real environment, in order to further improve the model performance and generalization capability. Secondly, this paper employs a cross-domain fusion model based on channel attention. This model integrates time-frequency domain feature vectors and utilizes a channel attention mechanism to enable the model to adaptively assign different weights to different feature channels, thus achieving feature fusion in the time-frequency domain and helping the model to learn the feature correlations between the time and frequency domains. This approach can effectively enhance model performance and improve its generalization capability. In addition, in order to reduce the data imbalance problem due to the lack of real anomaly samples, this paper designs an anomaly sample generator to help the model understand the potential anomaly distribution and gain the ability to model against potentially realistic anomaly samples. Finally, to further alleviate the impact of data imbalance, this paper adopts a statistical-based model. This model conducts feature modeling by directly generalizing the global characteristics of the data. As a result, this approach can directly avoid the model bias problem caused by data imbalance. At the same time, due to the lightweight design of the overall model, it can be quickly and easily deployed in resource-limited industrial manufacturing environments. The proposed method is tested on multiple benchmark datasets, and the results of the experiments demonstrate that the proposed model exhibits superior anomaly detection performance and domain adaptation capabilities.

Author Contributions: Conceptualization, J.L., F.Y. and X.L.; methodology, J.L., F.Y. and X.L.; software, J.L.; validation, J.L.; formal analysis, J.L.; investigation, J.L. and F.Y.; data curation, J.L.; writing—original draft preparation, J.L.; writing—review and editing, F.Y.; visualization, J.L.; supervision, F.Y. and X.L.; project administration, F.Y.; funding acquisition, F.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The MIMII dataset is is available at https://doi.org/10.5281/zenodo.3678171 (accessed on 2 February 2025). The DCASE2020 task2 dataset is available at https://doi.org/10.5281/zenodo.3678171 (accessed on 2 February 2025). The MIMII DUE dataset is available at https://doi.org/10.5281/zenodo.4562016 (accessed on 2 February 2025). The code presented in this study is available upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kulkarni, S.; Watanabe, H.; Homma, F. Self-Supervised Audio Encoder with Contrastive Pretraining for Respiratory Anomaly Detection. In Proceedings of the 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW), Rhodes, Greece, 4–10 June 2023; pp. 1–5. [CrossRef]
2. Suman, A.; Kumar, C.; Suman, P. Early detection of mechanical malfunctions in vehicles using sound signal processing. *Appl. Acoust.* **2022**, *188*, 108578. [CrossRef]
3. Koizumi, Y.; Yasuda, M.; Murata, S.; Saito, S.; Uematsu, H.; Harada, N. SPIDERnet: Attention Network For One-Shot Anomaly Detection In Sounds. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 281–285. [CrossRef]
4. Yamaguchi, M.; Koizumi, Y.; Harada, N. AdaFlow: Domain-adaptive Density Estimator with Application to Anomaly Detection and Unpaired Cross-domain Translation. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2018; pp. 3647–3651.
5. Shrivastava, A.; Vamsi, P.R. A Hybrid Method for Anomaly Detection Using Distance Deviation and Firefly Algorithm. In Proceedings of the 2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON), Raigarh, India, 8–10 February 2023; pp. 1–6. [CrossRef]

6. Jiang, Y.; Huang, T.; Wang, J.; Kang, C. Anomaly Detection of Argo Data using Variational Autoencoder and K-means Clustering. In Proceedings of the 2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 16–18 December 2022; Volume 5, pp. 1000–1004. [CrossRef]
7. Sharma, R.; Chaurasia, S. An Enhanced Approach to Fuzzy C-means Clustering for Anomaly Detection. In Proceedings of the First International Conference on Smart System, Innovations and Computing, Jaipur, India, 17–19 May; Somani, A.K., Srivastava, S., Mundra, A., Rawat, S., Eds.; Springer: Singapore, 2018; pp. 623–636.
8. Germain, F.G.; Wichern, G.; Roux, J.L. Hyperbolic Unsupervised Anomalous Sound Detection. In Proceedings of the 2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, NY, USA, 22–25 October 2023; pp. 1–5. [CrossRef]
9. Liu, Z.; Zhou, Y.; Xu, Y.; Wang, Z. SimpleNet: A Simple Network for Image Anomaly Detection and Localization. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 20402–20411. [CrossRef]
10. Sarwar, M.Z.; Cantero, D. Probabilistic autoencoder-based bridge damage assessment using train-induced responses. *Mech. Syst. Signal Process.* **2024**, *208*, 111046. [CrossRef]
11. Zhao, P.; Ding, Z.; Li, Y.; Zhang, X.; Zhao, Y.; Wang, H.; Yang, Y. SGAD-GAN: Simultaneous Generation and Anomaly Detection for time-series sensor data with Generative Adversarial Networks. *Mech. Syst. Signal Process.* **2024**, *210*, 111141. [CrossRef]
12. Bi, C.; He, S. Lightweight and Data-imbalance-aware Defect Detection Approach Based on Federated Learning in Industrial Edge Networks. In Proceedings of the 2023 IEEE 13th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 14–16 July 2023; pp. 60–64. [CrossRef]
13. Singh, J.; Gupta, S. Evaluating the Impact of Local Data Imbalance on Federated Learning Performance for IoT Anomaly Detection. In Proceedings of the 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), New Delhi, India, 6–8 July 2023; pp. 1–7. [CrossRef]
14. Kim, M.; Ho, M.T.; Kang, H.G. Self-supervised Complex Network for Machine Sound Anomaly Detection. In Proceedings of the 2021 29th European Signal Processing Conference (EUSIPCO), Dublin, Ireland, 23–27 August 2021; pp. 586–590.
15. Dohi, K.; Endo, T.; Purohit, H.; Tanabe, R.; Kawaguchi, Y. Flow-Based Self-Supervised Density Estimation for Anomalous Sound Detection. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 336–340.
16. Hojjati, H.; Armanfard, N. Self-Supervised Acoustic Anomaly Detection Via Contrastive Learning. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 22–27 May 2022; pp. 3253–3257.
17. Meghanani, A.; Anoop, C.S.; Ramakrishnan, A.G. An Exploration of Log-Mel Spectrogram and MFCC Features for Alzheimer's Dementia Recognition from Spontaneous Speech. In Proceedings of the 2021 IEEE Spoken Language Technology Workshop (SLT), Shenzhen, China, 19–22 January 2021; pp. 670–677. [CrossRef]
18. Zhang, H. mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.
19. DeVries, T. Improved Regularization of Convolutional Neural Networks with Cutout. *arXiv* **2017**, arXiv:1708.04552.
20. Chen, X.; He, K. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15750–15758.
21. Tang, Y.; Han, K.; Guo, J.; Xu, C.; Xu, C.; Wang, Y. GhostNetv2: Enhance cheap operation with long-range attention. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 9969–9982.
22. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4690–4699.
23. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
24. Purohit, H.; Tanabe, R.; Ichige, K.; Endo, T.; Nikaido, Y.; Suefusa, K.; Kawaguchi, Y. MIMII Dataset: Sound dataset for malfunctioning industrial machine investigation and inspection. *arXiv* **2019**, arXiv:1909.09347.
25. Tanabe, R.; Purohit, H.; Dohi, K.; Endo, T.; Nikaido, Y.; Nakamura, T.; Kawaguchi, Y. MIMII DUE: Sound dataset for malfunctioning industrial machine investigation and inspection with domain shifts due to changes in operational and environmental conditions. In Proceedings of the 2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, NY, USA, 17–20 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 21–25.
26. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
27. Koizumi, Y.; Kawaguchi, Y.; Imoto, K.; Nakamura, T.; Nikaido, Y.; Tanabe, R.; Purohit, H.; Suefusa, K.; Endo, T.; Yasuda, M.; et al. Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring. *arXiv* **2020**, arXiv:2006.05822.

28. Suefusa, K.; Nishida, T.; Purohit, H.; Tanabe, R.; Endo, T.; Kawaguchi, Y. Anomalous sound detection based on interpolation deep neural network. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–9 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 271–275.
29. Peng, Y.; Zhong, X.; Yang, X.; Hu, L. Detection of Abnormal Sound of Power Plant Equipment Fault based on Self-supervised Learning. In Proceedings of the 2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China, 29–31 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 174–178.
30. Dong, W.; Guo, F.; Cheng, T. Machine anomalous sound detection based on a multi-dimensional feature extraction self-encoder model. In Proceedings of the 2024 5th International Conference on Computer Engineering and Application (ICCEA), Hangzhou, China, 12–14 April 2024; pp. 1165–1169. [CrossRef]
31. Zhou, H.; Wang, K.; Yao, J.; Yang, W.; Chai, Y. Anomaly Sound Detection of Industrial Equipment Based on Incremental Learning. In Proceedings of the 2023 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS), Yibin, China, 22–24 September 2023; pp. 1–6. [CrossRef]
32. Liu, Y.; Guan, J.; Zhu, Q.; Wang, W. Anomalous Sound Detection Using Spectral-Temporal Information Fusion. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 22–27 May 2022; pp. 816–820. [CrossRef]
33. Geng, B.; Liao, Y.; Guo, L.; Feng, X.; Cui, K. Anomaly Detection in Rotating Machinery Sound Based on Time and Spectral Information Fusion. In Proceedings of the 2024 IEEE 13th Data Driven Control and Learning Systems Conference (DDCLS), Kaifeng, China, 17–19 May 2024; pp. 1434–1438. [CrossRef]
34. Ma, X.; Liao, Y.; Guo, L.; Geng, J.; Wang, G. Abnormal Sound Detection of Electrical Equipment Based on Time-Spectrum Information Fusion. In Proceedings of the 2023 IEEE 12th Data Driven Control and Learning Systems Conference (DDCLS), Xiangtan, China, 12–14 May 2023; pp. 985–989. [CrossRef]
35. Kawaguchi, Y.; Imoto, K.; Koizumi, Y.; Harada, N.; Niizumi, D.; Dohi, K.; Tanabe, R.; Purohit, H.; Endo, T. Description and Discussion on DCASE 2021 Challenge Task 2: Unsupervised Anomalous Sound Detection for Machine Condition Monitoring under Domain Shifted Conditions. *arXiv* **2021**, arXiv:abs/2106.04492.
36. Gu, X.; Li, R.; Kang, M.; Lu, F.; Tang, D.; Peng, J. Unsupervised adversarial domain adaptation abnormal sound detection for machine condition monitoring under domain shift conditions. In Proceedings of the 2021 IEEE 20th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), Banff, AB, Canada, 29–31 October 2021; pp. 139–146. [CrossRef]

applied sciences

Article

DVCW-YOLO for Printed Circuit Board Surface Defect Detection

Pei Shi [1,2,†], Yuyang Zhang [1,†], Yunqin Cao [1], Jiadong Sun [1], Deji Chen [2,3,*] and Liang Kuang [4]

1 School of IoT Engineering, Wuxi University, Wuxi 214105, China; ship@cwxu.edu.cn (P.S.); zyy1641302152@163.com (Y.Z.); cdquehxsra591023@126.com (Y.C.); sunjiadong@cwxu.edu.cn (J.S.)
2 Jiangsu Internet of Things Hyper-Converged Application and Security Engineering Research Center, Wuxi 214105, China
3 Jiangsu Foreign Expert Lab—Wuxi University School of Internet of Things Engineering Foreign Expert Lab, Wuxi 214105, China
4 School of Internet of Things Engineering, Jiangsu Vocational College of Information Technology, Wuxi 214153, China; kuangl@jsit.edu.cn
* Correspondence: dejichen@tongji.edu.cn
† These authors contributed equally to this work.

Abstract: The accurate and efficient detection of printed circuit board (PCB) surface defects is crucial to the electronic information manufacturing industry. However, current approaches to PCB defect detection face challenges, including large model sizes and difficulties in balancing detection accuracy with speed. To address these challenges, this paper proposes a novel PCB surface defect detection algorithm, named DVCW-YOLO. First, all standard convolutions in the backbone and neck networks of YOLOv8n are replaced with lightweight DWConv convolutions. In addition, a self-designed C2fCBAM module is introduced to the backbone network for extracting features. Next, within the neck structure, the C2f module is substituted with the more lightweight VOVGSCSP module, thereby reducing model redundancy, simplifying model complexity, and enhancing detection speed. By enhancing prominent features and suppressing less important ones, this modification allows the model to better focus on key regions, thereby improving feature representation capabilities. Finally, the WIoU loss function is implemented to replace the traditional CIoU function in YOLOv8n. This adjustment addresses issues related to low generalization and poor detection performance for small objects or complex backgrounds, while also mitigating the impact of low-quality or extreme samples on model accuracy. Experimental results demonstrate that the DVCW-YOLO model achieves a mean average precision (mAP) of 99.3% and a detection speed of 43.3 frames per second (FPS), which represent improvements of 4% and 4.08%, respectively, over the YOLOv8n model. These results confirm that the proposed model meets the real-time PCB defect detection requirements of small and medium-sized enterprises.

Keywords: PCB board defect detection; YOLOv8n model; DVCW-YOLO model; C2fCBAM structure; WIoU loss function

1. Introduction

Printed circuit boards (PCBs) are extensively employed in integrated circuits. However, the inherent instability of the production process, coupled with environmental factors and equipment conditions, frequently leads to the emergence of various defects, including missing holes, rat bites, and open circuits. These defects have a detrimental impact on both product performance and safety [1]. Consequently, there is a growing demand for efficient, accurate, and cost-effective PCB defect inspection systems.

In recent years, deep learning-based models for PCB surface defect detection have garnered significant attention. Current PCB defect detection methods using vision technology can be broadly classified into two categories: two-stage object detection algorithms and single-stage object detection algorithms [2]. Two-stage object detection algorithms first locate the target and generate detection boxes, followed by the classification of the detected boxes to refine the target categories and positions. Notable models include the region-based convolutional neural networks (R-CNNs) [3–5]. For example, Ren et al. [5] developed an object detection framework that combines the strengths of R-CNN and a new region proposal network (RPN) to improve the speed and accuracy of object detection. Kumar and Singh [6] address the challenge of detecting defects by introducing a two-stage object detection network based on region proposal network (RPN) for PCB defect detection. This type of two-stage detection algorithm presents several challenges, including limited inference speed and susceptibility to interference from background noise. In contrast, single-stage object detection algorithms combine feature extraction with prediction frame positioning, enabling the direct determination of target categories and detection frame positions. Examples include SSD [7] and YOLO [8]. Kumar and Singh [9] leveraged the SSD architecture for PCB defect detection, providing a detailed account of the training process, evaluation metrics, and the effectiveness of SSD in identifying various defects, including soldering issues and component misplacements. Liao et al. [10] introduced a cost-effective detection model based on YOLOv4, which improves the activation function in the backbone and neck prediction networks, reducing both the model size and the number of parameters. Qing et al. [11] proposed an improved PCB component detection model based on YOLOv8, which utilizes the C2Focal module as the core of the backbone network, combining fine-grained local features with coarse-grained global features. Chen et al. [12] introduced the W-YOLOv8 model, which uses a gradient gain allocation strategy with a dynamic non-monotonic focusing mechanism to direct the model's attention to anchor frames of ordinary mass, improving overall performance. While these studies have enhanced detection accuracy, a balance between detection performance and model size remains elusive.

In response to the challenges posed by small target sizes, missed detections, and excessive model parameters in PCB surface detection, this study introduces an enhanced detection model based on YOLOv8, called DVCW-YOLO. This model significantly mitigates parameter redundancy and model complexity through the integration of depthwise convolution (DWConv), VOVGSCSP modules, a self-constructed C2fCBAM module, and a weighted intersection over union (WIoU) loss function. The specific contributions of this work are outlined as follows:

1. The C2fCBAM module is developed by synergistically combining the C2f architecture with the convolutional block attention module (CBAM) mechanism. This integration enables the detection model to effectively prioritize critical regions by enhancing salient features while attenuating less relevant ones. Consequently, a novel backbone network is established to facilitate the extraction of key features.
2. A neck network is designed by incorporating the VOVGSCSP module and DWConv. The implementation of the WIoU loss function during training serves to alleviate the adverse effects of low-quality and extreme samples, thereby improving model accuracy, accelerating convergence, and reducing the high rate of missed detections commonly associated with PCB defect identification.
3. We augmented the dataset and executed comparative experiments involving nine detection algorithms, which substantiate that the proposed DVCW-YOLO consistently outperforms its counterparts in most cases.

The main contributions of this paper are outlined as follows. Section 2 introduces the architecture of the proposed DVCW-YOLO defect detection model. Section 3 presents the dataset construction, including data presentation, augmentation, and defect distribution characteristics. Section 4 details the experimental setup, parameter settings, detection results, and analysis. Finally, Section 5 concludes the paper with a summary of key findings.

2. Methodology

To achieve accurate PCB defect detection, we integrate the YOLOv8n framework with a self-designed C2fCBAM module, the VOVGSCSP module and DWConv to enhance detection performance. The following section details these approaches and their construction to address the challenges of PCB defect detection.

2.1. Proposed DVCW-YOLO Network

YOLOv8 [13] is a widely utilized detector in the field of target detection. It incorporates the cross stage partial (CSP) structure within backbone network, along with the spatial pyramid pooling with factorized convolutions (SPPF) module at its output stage. The neck network features a bidirectional path aggregation network (PAN-FPN) for efficient feature integration [14]. However, the backbone and neck networks of YOLOv8 exhibit several limitations, including the restricted feature extraction capability of C2f module, high computational complexity, and a substantial number of standard convolution parameters [15]. Additionally, the complete intersection over union (CIoU) loss function does not adequately account for the characteristics of the background region, which may lead to the neglect of surrounding contextual information, consequently diminishing detection performance [16].

To address these challenges, including the large model size in fire detection tasks and the difficulty in balancing detection accuracy with model efficiency, this paper achieves several improvements to the backbone network, neck network, and loss function based on the YOLOv8n architecture, and develops the DVCW-YOLO detection model. The overall architecture of the DVCW-YOLO model is illustrated in Figure 1. All standard convolutions in both the backbone and neck networks are replaced with the more lightweight DWConv. Additionally, we introduce the C2fCBAM module in the backbone network, which incorporates the CBAM attention mechanism at both the input and output ends of the C2f module. This new module replaces the original C2f module, enhancing the model's ability to extract features from key areas by emphasizing salient features and suppressing less important features. Secondly, the C2f module is replaced with global structure-aware convolution (GSConv) and the more efficient VOVGSCSP module in the neck structure, reducing both the number of model parameters and the overall model complexity. Finally, the WIoU loss function is used to replace the traditional CIoU loss function. This modification ensures that the improved model focuses more on normal and high-probability samples, thereby enhancing detection accuracy.

2.2. C2fCBAM Model

When the background on the surface of the PCB is complex, it can be easily confused with the actual defect areas, leading to false detections. To address this issue and enable the detection model to focus more on key information in the input features while minimizing attention to background noise, this paper optimizes the backbone network by designing and implementing the C2fCBAM module. The CBAM attention mechanism is added to both the input and output ends of the original C2f module [17], as shown in Figure 2. This structure enables the model to prioritize critical regions by enhancing salient features while suppressing less relevant ones, thereby facilitating the differentiation between normal and defective areas. This capability is particularly beneficial in scenarios where defects are subtle or when there is substantial background interference.

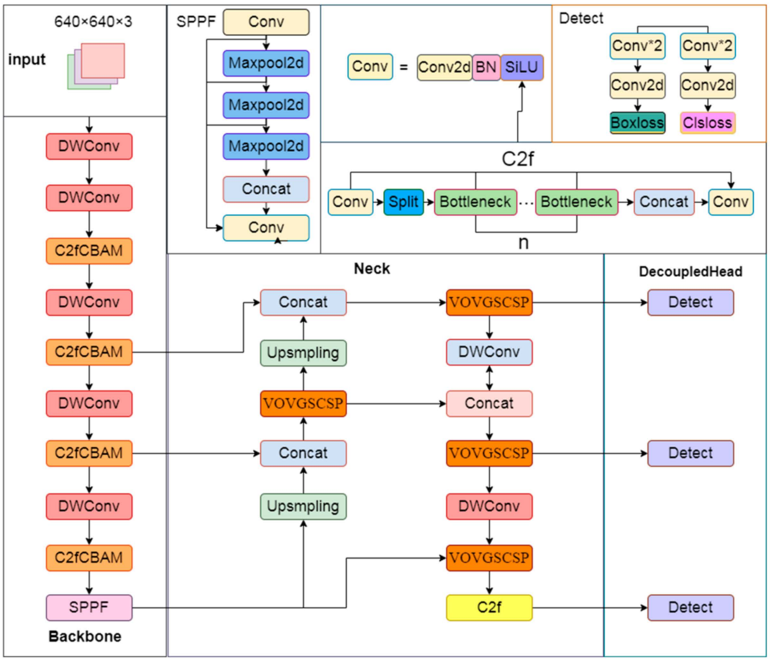


Figure 1. DVCW-YOLO model structure diagram.

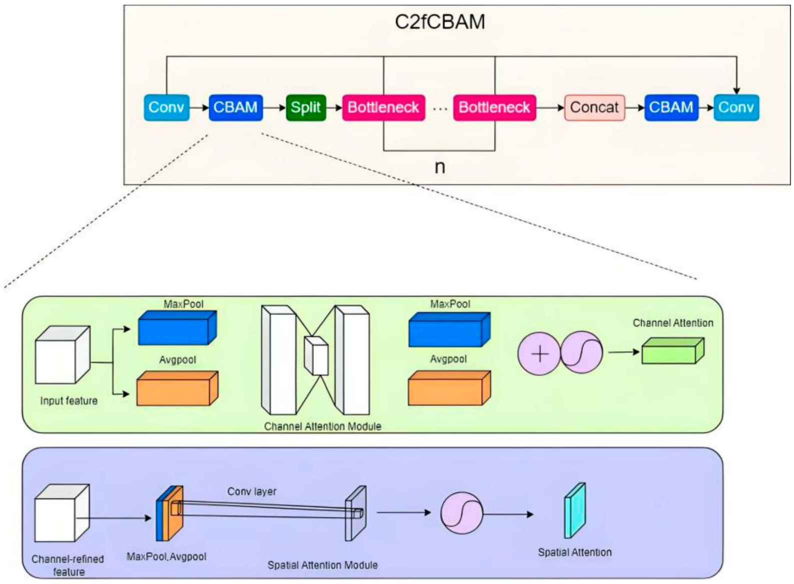


Figure 2. C2fCBAM module structure diagram.

The CBAM consists of two components: the channel attention module and the spatial attention module [17]. The channel attention module assigns weights to the feature channels, adaptively adjusting the importance of each channel to improve the network's ability to capture specific features. Conversely, the spatial attention module focuses on the spatial dimensions of the feature map, enhancing the network's capacity to perceive spatial locations by adaptively adjusting the importance of each spatial position. By incorporating CBAM, the network can better learn the distinguishing features of the input data, thereby improving its ability to differentiate between various objects and backgrounds.

2.3. Neck Network

In the context of PCB defect detection, the traditional YOLOv8n model suffers from large model parameters and limited detection accuracy. To address these limitations, this paper reduces the model parameters by integrating DWConv convolutions and VOVGSCSP modules, thereby improving detection speed. Specifically, all standard convolutions in the neck structure are replaced with lighter DWConv convolutions. Additionally, due to the redundant processing of input features in the C2f module, a significant residual redundancy issue arises. To mitigate this, the C2f module is replaced with the more efficient VOVGSCSP module, further reducing model parameters and enhancing detection speed.

2.3.1. DWConv Convolution

DWConv [18] is a specialized convolution technique, and its structure is illustrated in Figure 3. Unlike standard convolution, where all channels share a single convolutional kernel, DWConv assigns a unique convolutional kernel to each channel. This technique groups the input feature maps and applies convolution within each group independently. Assuming the current number of groups is N, the number of parameters required for DWConv is only $1/N$ of the parameters required for standard convolution. Consequently, the use of DWConv significantly reduces the number of model parameters, resulting in a more lightweight model.

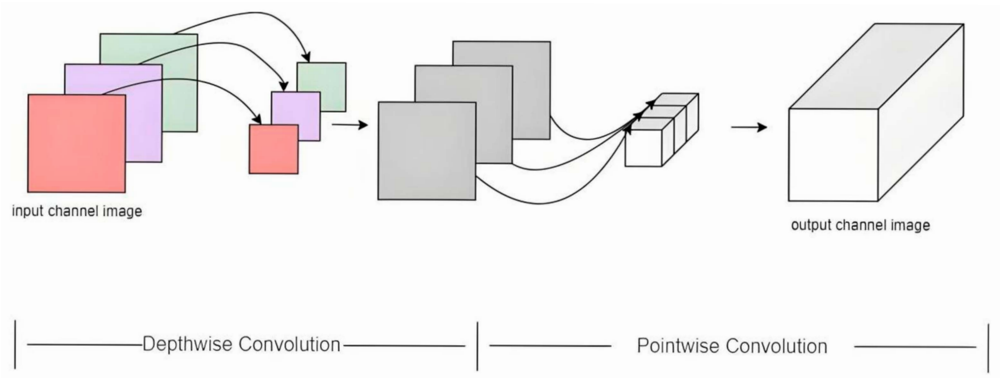


Figure 3. Depthwise separable convolution structure diagram.

2.3.2. VOVGSCSP Module

The VOVGSCSP module [19], in contrast to the C2f module, primarily consists of multiple GSConv and standard Conv convolutions, which are connected through residual links. The GSConv module integrates several techniques, including standard Conv, DWConv, and shuffle operations [20]. The shuffle operation enables the combination of standard Conv with DWConv by mixing features across channels, which helps to reduce computational costs. As the number of input feature channels increases, the computational cost of GSConv becomes significantly lower than that of standard Conv, approximately halving the computational burden, while still preserving the feature extraction capabilities of the original convolutional operation. This structure of the VOVGSCSP module is depicted in Figure 4. The computational costs associated with standard Conv and GSConv are outlined as follows:

$$GFLOP_{s1} = W \times H \times K \times K \times 1 \times C_{out} \tag{1}$$

$$GFLOP_{s2} = W \times H \times K \times K \times 1 \times \frac{C_{out}}{2}(C_{in} + 1) \tag{2}$$

For a given input feature map with size $W \times H$, K is the size of the convolution kernel, C_{in} is number of input channels, and C_{out} denotes the output channels. As the number of output channels increases, the computational advantage of DWConv over standard convolution becomes more pronounced. The VOVGSCSP structure effectively leverages these properties, achieving efficient feature extraction and processing.

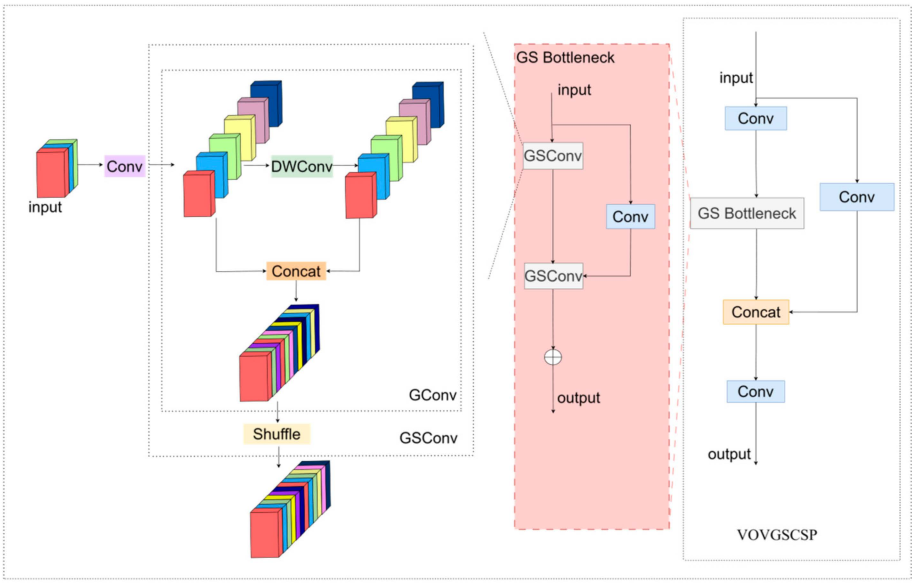


Figure 4. Structure diagram of VOVGSCSP module with GSConv.

2.4. WIoU Loss Function

The CIoU loss function [21] in the YOLOv8 model performs suboptimally when detecting low-quality samples, such as small defects, and exhibits limited generalization ability. To address these issues, the WIoU loss function [22,23] in the DVCW-YOLO model incorporates dynamic and non-monotonic focusing mechanisms. These mechanisms enhance the model's ability to handle small objects or complex backgrounds by improving generalization and detection quality. The WIoU function assesses anchor frame quality by considering outliers in the dataset and adjusting sample weights to reduce the impact of low-quality samples.

$$L_{WIoU} = R_{WIoU} L_{IoU} \tag{3}$$

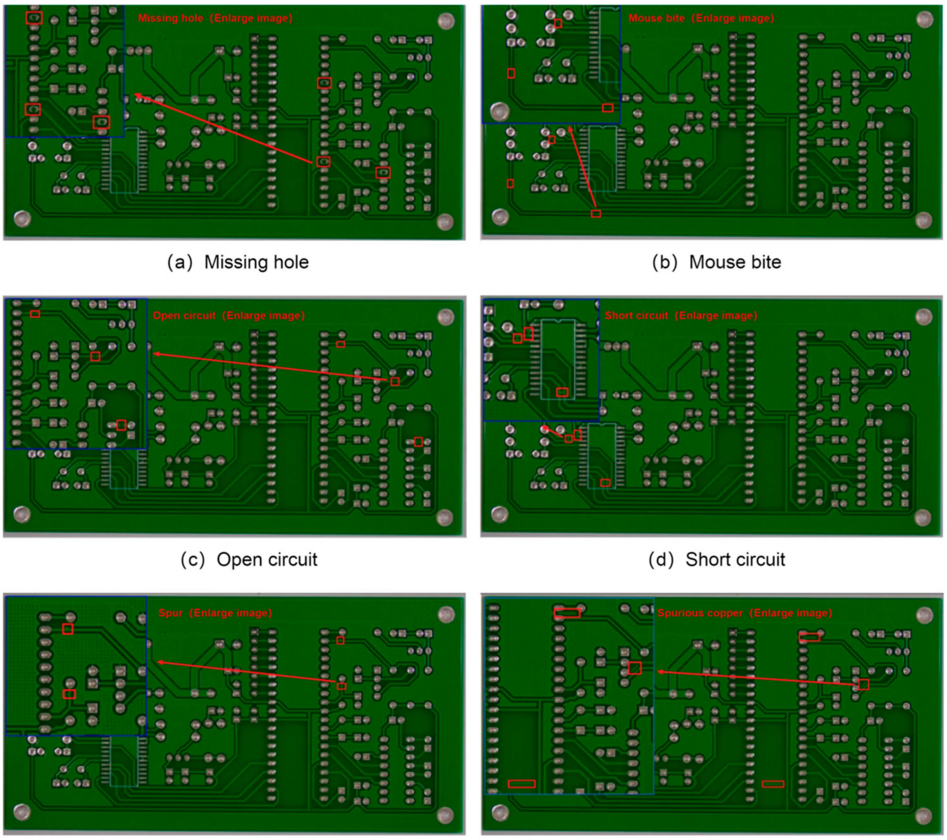
$$R_{WIoU} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(W_g^2 + H_g^2)}\right) \tag{4}$$

$$L_{IoU} = 1 - IoU \tag{5}$$

Specifically, $LWIoU$ denotes the modified loss function, while (X, Y) are the horizontal and vertical coordinates of the predicted bounding box center, respectively. (Xgt, Ygt) represent the center coordinates of the true bounding box, and (Hgt, Wgt) denote the height and width of the true bounding box, respectively. The loss $RWIoU$ is computed using an exponential function, which increases the weight of anchor boxes that are closer to the true bounding box while reducing the impact of those farther away. This loss $IIoU$ complements the IoU between the predicted bounding box and the true bounding box, focusing on improving the overlap quality.

3. Dataset Building

In this study, we utilize the open-source dataset from the Open Laboratory of Intelligent Robotics at Peking University, which contains six types of PCB board defects: missing hole, mouse bite, open circuit, short circuit, spurious copper, and spur. A subset of the dataset is shown in Figure 5. The dataset consists of 693 images with pixel dimensions of 2777×2138, with each image containing 3–6 defects. The defects occupy approximately 0.24% of the total pixel area in each image. Due to the limited size of this publicly available dataset, the small number of samples poses a challenge for effective PCB defect detection. To address this issue, we apply data augmentation techniques, including mosaic enhancement, flipping, random noise addition, and brightness adjustment, to randomly expand the dataset and improve the generalization ability of the detection model. After augmentation, the dataset contains a total of 10,688 images, and the ratio of the training set, validation set, and test set is 7:2:1. Some instances of these augmentations are shown in Figure 6.

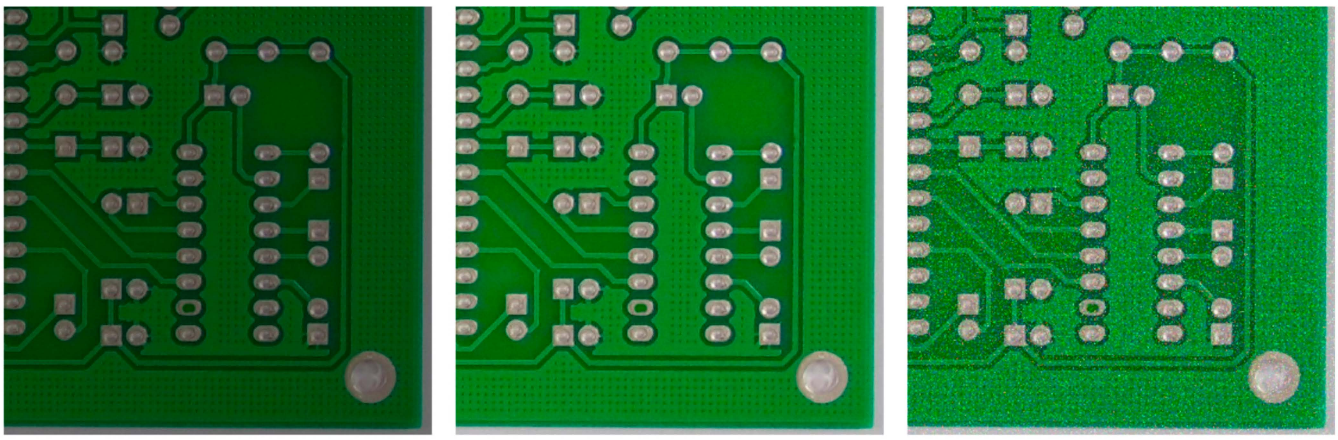


(a) Missing hole (b) Mouse bite

(c) Open circuit (d) Short circuit

(e) Spur (f) Spurious copper

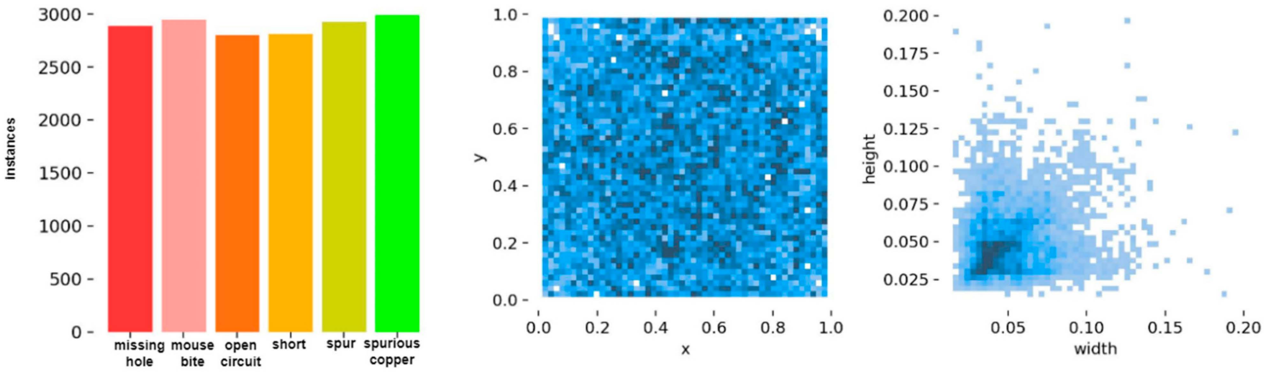
Figure 5. Instances of PCB surface defects.



(a) Original (b) Brightness enhancement (c) Gaussian noise

Figure 6. Data augmentation instances.

To gain an intuitive understanding of the defect information after data augmentation, we analyzed the overall distribution, size, and quantity of defects, as depicted in Figure 7. As depicted in Figure 7a, the augmented dataset generated through random data augmentation, the number of images across the six defect categories varies, although the differences are relatively small. The number of missing holes, mouse bites, open circuits, short circuits, spurs, and spurious copper are 2883, 2935, 2798, 2805, 2915, 3000, respectively. Figure 7b illustrates the positions of the center points of all defect detection bounding boxes within the augmented images. From Figure 7b, it is evident that the defect center points of all augmentations are widely dispersed, suggesting that the defects do not follow a clear spatial pattern. Figure 7c presents the width and height of all bounding boxes in the augmented dataset. As seen in Figure 7c, the majority of the bounding boxes are concentrated near the origin, with their dimensions primarily in the range of [0.025, 0.05]. This indicates that most of the detection bounding boxes are small, with the majority of the defects being small-sized targets.



(a) Dataset category distribution map (b) Map of the location of the target box (c) Target box size distribution map

Figure 7. Defect distribution information of augmentation.

4. Experiments and Analysis of Results

In the following section, we discuss the experiment and evaluate the proposed DVCW-YOLO detection model using the augmented dataset. We performed three validation experiments to assess the performance of the model and determine key evaluation metrics and model parameters. The first experiment tested the effectiveness of the C2fCBAM module in the optimized backbone network, comparing its performance with different attention mechanisms. The second experiment conducted ablation studies on the DWConv, C2fCBAM, VOVGSCSP, and WIoU components designed for the backbone network and neck network, respectively, to evaluate the impact of each optimization on model performance. The third experiment compared the proposed DVCW-YOLO detection model with several state-of-the-art models, including Faster R-CNN, SSD, YOLOv3-tiny, YOLOv4-tiny, YOLOv5s, YOLOv7, YOLO-G, and YOLOv8-PCB, to demonstrate the performance advantages of the DVCW-YOLO model.

4.1. Evaluation Indicators

To comprehensively evaluate the performance of the model, six evaluation metrics are employed: precision (P) [24], recall (R) [25], mAP [26], floating point operations (FLOPs) [27], parameters [28], and FPS [29]. Precision is defined as the ratio of correctly detected positive samples to the total number of detected positives, including false positives. Recall, on the other hand, measures the ratio of correctly detected positive samples to the total number of actual positive samples, including undetected positives. The mAP represents the average precision across multiple categories and recall levels. FLOPs and the number of parameters are indicators of model complexity. FLOPs measure the computational complexity, with a higher number of FLOPs indicating a greater computational cost and increased demand for computing resources. The parameters metric refers to the number of model parameters, which provides insight into the model's memory usage and overall size. Finally, FPS is used to assess the model's detection speed, indicating the timeliness of the model's performance. The formula for calculating these metrics is as follows:

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

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

$$AP = \int_0^1 P(R)dR \tag{8}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{9}$$

where TP (true positive) refers to the number of correctly identified positive targets. FN (false negative) represents the number of positive targets that were missed or incorrectly classified. FP (false positive) denotes the number of non-positive targets incorrectly identified as positive. N is the total number of categories. t indicates the average detection time for each image, which is used to calculate FPS.

4.2. Experimental Environment and Parameter Settings

The experiments were conducted on an Ubuntu 18.04 platform with the following specifications: Intel (R) Xeon (R) Platinum 8255C CPU @ 2.50GHz, NVIDIA RTX 2080 Ti (11 GB GPU), PyTorch version 1.8.1, Python 3.8.0, and the VSCode-integrated development environment. Anaconda 4.13.0 was used to create a virtual environment.

Model training parameters directly influence the accuracy and performance of the model. To ensure consistency and reliability, all models were tested under the same training conditions. The specific parameter settings are as follows: learning rate (LR) = 0.001; batch size = 32; momentum factor = 0.937; and weight decay = 0.0005. The dataset was split into training, validation, and test sets at a 7:2:1 ratio. All models were initialized using transfer learning with the pre-trained weights of YOLOv8n. The training process variation curves for the bounding box loss (Box_Loss) and the categorical loss (Cls_Loss) are shown in Figure 8.

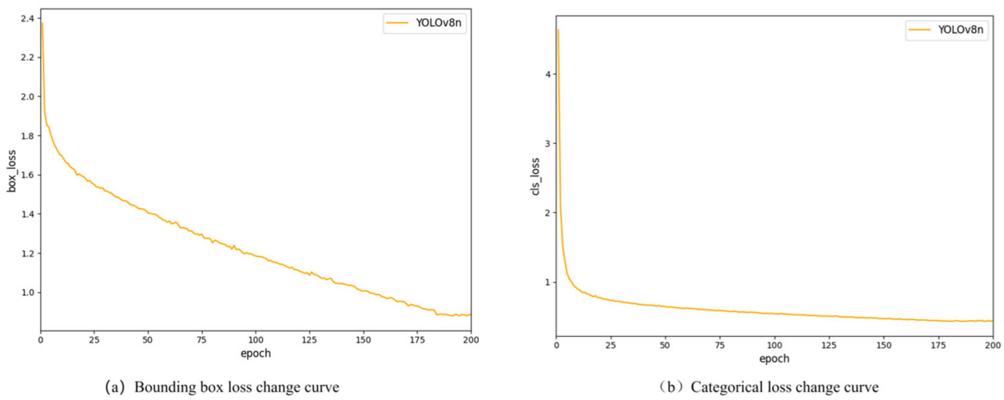


Figure 8. Loss curve during training.

As shown in Figure 8, both the bounding box loss and classification loss of the model gradually decrease as the number of training iterations increases. In the early stages of training, the learning rate is higher, leading to a faster reduction in loss. In Figure 8a, it illustrates that the bounding box loss curve begins to flatten after 180 epochs. Similarly, Figure 8b shows that the classification loss curve decreases more slowly after 100 epochs and becomes stable around 180 epochs. Based on these observations, this study sets the total number of training epochs to 200.

4.3. Experimental Comparative Analysis

4.3.1. Comparative Experiments on Attention Mechanisms

To evaluate the detection performance of the C2fCBAM module in the YOLOv8n backbone network, comparative experiments are conducted with the gated attention module (GAM) [30] and learned spatial kernel (LSK) attention modules [31]. The detection performance of each module is analyzed, and the experimental results are presented in Table 1.

Table 1. Performance comparison of three attention mechanisms.

Attention Mechanism of YOLOV8n Model Backbone Network	P/%	R/%	mAP0.5/%	FPS (f/s)	FLOPs/G	Parameters/M
NONE	96.1	95.0	95.3	41.6	8.9	3.15
+GAM	98.2	99.0	99.0	18.8	18.0	7.87
+LSK	98.2	99.0	99.1	35.7	9.3	3.35
+CBAM	98.4	99.4	99.3	41.2	8.9	3.17

From Table 1, it is evident that the C2fCBAM module, when utilizing the CBAM attention mechanism, demonstrates significant advantages in detection accuracy and model complexity. The mAP50 with the CBAM attention mechanism was 99.3%, which is 4.1%, 0.3%, and 0.2% higher than the mAP50 values for YOLOv8n, GAM, and LSK without an attention mechanism, respectively. Additionally, the P and R values for the CBAM attention mechanism were 98.4% and 99.4%, representing increases of 2.3%, 0.2%, and 0.2%, and 4.6%, 0.4%, and 0.4%, respectively, compared to YOLOv8n, GAM, and LSK without attention mechanisms. Moreover, adopting the CBAM attention mechanism resulted in a minimal increase of 0.02 M in model parameters, which is a negligible increment for the original model. In conclusion, the C2fCBAM module, based on the CBAM attention mechanism, effectively enhances the detection accuracy of the YOLOv8n backbone network without significantly increasing the model size, while also maintaining computational efficiency.

4.3.2. Optimize the Module Ablation Experiment

To verify the performance of each optimization operation, ablation experiments were conducted on DWConv, C2fCBAM, VOVGSCSP, and WIOU loss functions in the backbone and neck networks, respectively. Four sets of experiments were designed to evaluate different optimization strategies by incorporating various modules into the YOLOv8n model, with each experiment using identical training parameters. The experimental results are presented in Table 2.

An analysis of Table 2 reveals several key performance improvements when comparing the YOLOv8n-D model to YOLOv8n. Specifically, the mAP of YOLOv8n-D increased by 3.9%, the FPS improved by 0.5, the model's FLOPs decreased by 1.2 G, and the number of parameters was reduced by 0.57 M. These findings indicate that the incorporation of the DWConv convolution operation not only enhances detection accuracy but also reduces the model's parameter count, contributing to a lighter model. When comparing YOLOv8n-DV to YOLOv8n-D, the mAP of YOLOv8n-DV slightly decreased by 0.4%, but FPS increased by 1.5, FLOPs reduced by 0.7 G, and the number of parameters decreased by 0.12 M. This suggests that while the addition of VOVGSCSP leads to a minor reduction in detection accuracy, it effectively boosts detection speed and reduces model complexity. The further comparison of YOLOv8n-DVM with YOLOv8n-DV shows a 0.4% improvement in mAP, with no changes in detection speed or model complexity. These results indicate that the YOLOv8n-DVM model, optimized with the WIOU loss function, enhances the quality of detection anchor frames, improves the model's attention to small samples, and ultimately boosts detection accuracy. Finally, in comparison with YOLOv8n-DVW, the DVCW-YOLO model shows a 0.2% increase in mAP, a slight decrease of 0.3 in FPS, no change in FLOPs, and a marginal increase of 0.01 M in parameters. This suggests that the C2fCBAM structure used in DVCW-YOLO effectively improves detection accuracy, with only a minimal increase in model size.

Table 2. Performance comparison of ablation experiment.

Model	DWConv	VOVGSCSP	WIOU	C2fCBAM	P/%	R/%	mAP0.5/%	FPS (f/s)	FLOPs /G	Params /M
YOLOV8N	×	×	×	×	96.1	95.0	95.3	41.6	8.9	3.15
YOLOV8N-D	√	×	×	×	98.2	99.0	99.1	42.1	7.7	2.58
YOLOV8N-DV	√	√	×	×	98.5	98.5	98.7	43.6	7.0	2.46
YOLOV8N-DVW	√	√	√	×	98.6	98.7	99.1	43.6	7.0	2.46
DVCW-YOLO	√	√	√	√	99.1	99.4	99.3	43.3	7.0	2.47

To further analyze the impact of different modules on model performance, the trends of the detection loss value and mAP for each detection model were compared across training iterations, as shown in Figure 9. Under the same threshold, the DVCW-YOLO model outperforms the other models. As seen in Figure 9a, after 50 iterations, the loss curves for all models fluctuate within a small range, but the DVCW-YOLO model consistently exhibits the lowest loss, indicating that the model's predictions are closer to the actual labels in the training data. Additionally, the DVCW-YOLO model and the YOLOv8n-DVW model converge significantly faster than YOLOv8n-DV, YOLOv8n-D, and YOLOv8n, suggesting that the C2fCBAM and WIOU loss functions facilitate optimization efficiency. Figure 9b illustrates that the mAP value of DVCW-YOLO model converges more quickly than the other four models and tends to stabilize after training 50 epochs. This confirms that the DVCW-YOLO model effectively performs PCB defect detection and exhibits clear overall performance advantages.

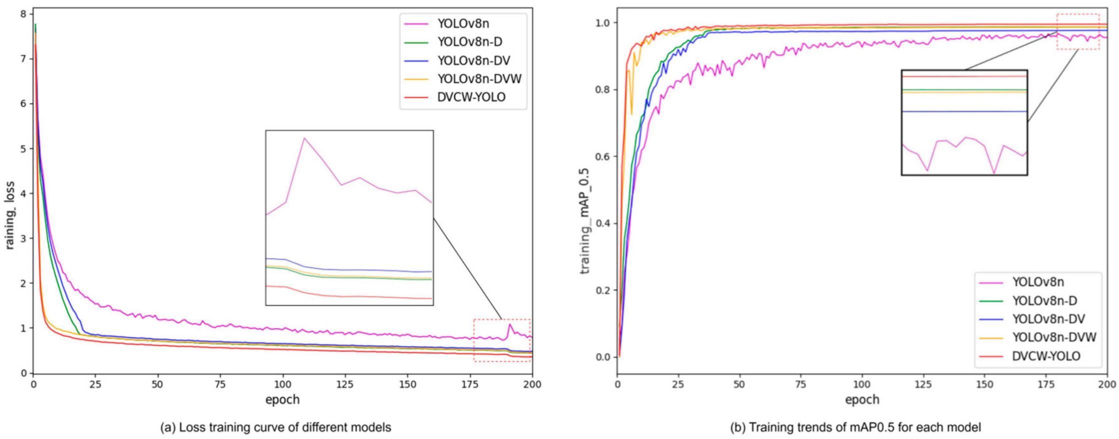


Figure 9. Comparison curve of different models.

4.3.3. Performance Comparison of Different Detection Algorithms

To further assess the object detection performance of the DVCW-YOLO model proposed in this study, it was compared with several state-of-the-art models, including Faster R-CNN, SSD, YOLOv3-tiny, YOLOv4-tiny, YOLOv5s, YOLOv7, YOLO-G [32], and YOLOv8-PCB [33]. The detailed comparison results are presented in Table 3.

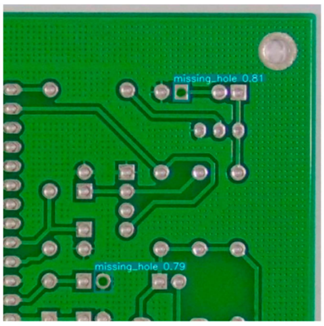
Table 3. Performance comparative results of different models.

Model	mAP0.5/a%	FPS (f/s)	FLOPs/G
Faster-RCNN	89.1	1.2	129
SSD	76.8	23.3	31.5
YOLOv3-tiny	79.6	26.7	26.4
YOLOv4-tiny	85.9	30.1	21.2
YOLOv5s	93.6	32.1	16.4
YOLOv7	94.6	32.5	9.8
YOLO-G [32]	99.0	-	13.2
YOLOv8-PCB [33]	98.37	8.8	-
YOLOv8n	95.3	41.6	8.9
DVCW-YOLO	**99.3**	**43.3**	**7.0**

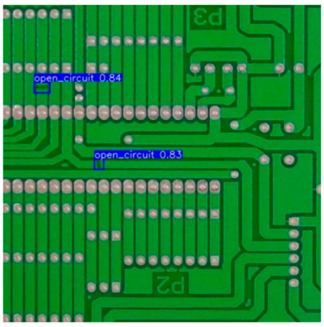
The DVCW-YOLO model achieved the highest mAP@0.5 of 99.3% among all the models evaluated. This represents improvements of 11.4%, 29.2%, 24.7%, 15.5%, 6%, 4.9%, 0.3%, 0.9%, and 4.1% over faster R-CNN, SSD, YOLOv3-tiny, YOLOv4-tiny, YOLOv5s, YOLOv7, YOLO-G, and YOLOv8n, respectively. In addition to its superior accuracy, the DVCW-YOLO model also demonstrates a remarkable efficiency, with a FLOPs value of only 7.0 G. In comparison, it shows a significant advantage in computational efficiency, reducing FLOPs by 122 G, 24.5 G, 19.4 G, 14.2 G, 9.4 G, 2.8 G, and 6.2 G over faster R-CNN, SSD, YOLOv3-tiny, YOLOv4-tiny, YOLOv5s, YOLOv7, YOLO-G, and YOLOv8n, respectively. DVCW-YOLO also had significant advantage in detection speed, and its FPS value was greater by 42.1, 20, 16.6, 13.2, 11.2, 10.8, 34.5, 1.7 than those of faster R-CNN, SSD, YOLOv3-tiny, YOLOv4-tiny, YOLOv5s, YOLOv7, YOLOv8-PCB and YOLOv8ns. Overall, the results indicate that the DVCW-YOLO model offers clear advantages in both detection accuracy and speed compared to other models, demonstrating its superior performance in object detection tasks.

4.3.4. The Analysis of Detection Results

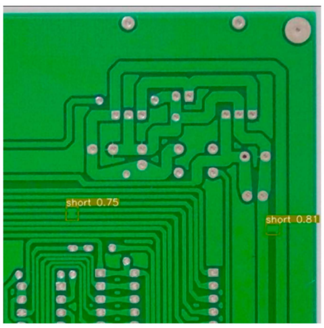
To further analyze the detection effect of proposed DVCW-YOLO model, we classify the detection results for various defects in the test set, including missing holes, open circuits, short circuits, and spurious copper. The detection outcomes of the DVCW-YOLO model and the original YOLOv8n model are presented in Figures 10 and 11.



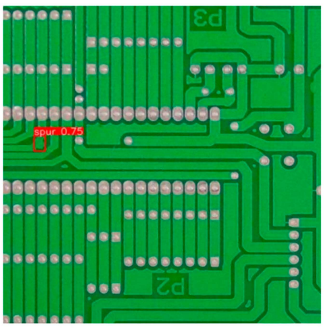
(a) Missing hole detection effect diagram



(b) Renderings of open circuit detection

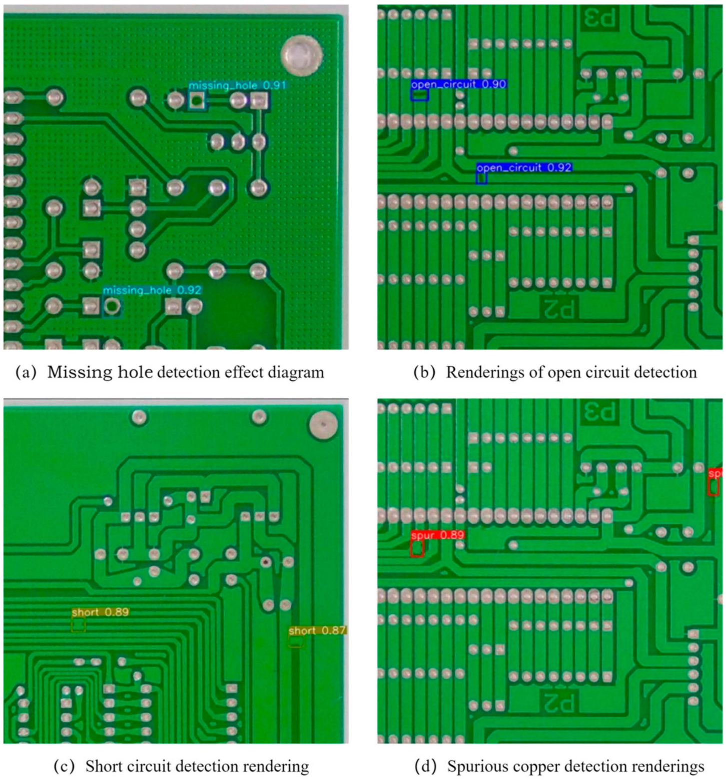


(c) Short circuit detection rendering



(d) Spurious copper detection renderings

Figure 10. Detection results of YOLOv8n.



(a) Missing hole detection effect diagram (b) Renderings of open circuit detection
(c) Short circuit detection rendering (d) Spurious copper detection renderings

Figure 11. Detection results of DVCW-YOLO.

As illustrated in Figures 10a–c and 11a–c, the DVCW-YOLO model shows higher confidence in detecting defects such as PCB missing holes, open circuits, and short circuits compared to the YOLOv8n model. The mentioned four defects are detectable using the Yolov8n with a confidence score around 0.8. The use of DVCW-YOLO can not only effectively detect these defects but can also achieve a detection confidence of about 0.9. Notably, in Figure 10d, YOLOv8n misses a defect, while in Figure 11d, the DVCW-YOLO model accurately detects all defects. These results highlight the significant improvement in detection accuracy achieved by the DVCW-YOLO model, as well as its effectiveness in reducing the missed detection rate. This makes the DVCW-YOLO model more suitable for practical PCB defect detection applications.

5. Conclusions

To tackle the challenges of balancing large model sizes and high missed detection rates that commonly afflict traditional detection methods, we propose an advanced defect detection model for PCBs based on the YOLOv8n architecture. Our approach involves replacing all conventional convolutional layers in the YOLOv8n backbone with lightweight DWConv. Additionally, we introduce the C2fCBAM module into the backbone network to enhance the extraction of salient features while reducing the influence of secondary features. This modification allows the model to focus more effectively on critical regions, thereby improving its ability to accurately characterize and extract features of the detected objects, particularly addressing the challenge of high false detection rates for small targets. In the neck network, we substitute the C2f module with a more efficient VOVGSCSP module to optimize the trade-off between model accuracy, detection speed, and overall model size. Moreover, we implement the WIoU loss function instead of the traditional CIoU loss function, which helps alleviate the impact of low-quality and extreme samples, thereby enhancing model accuracy, accelerating convergence, and reducing the high missed detection rate typically encountered in PCB defect detection. Our experimental results on self-augmented datasets indicate that the DVCW-YOLO model achieves mAP of 99.3%, along with a detection speed of 43.3 FPS.

In future work, we plan to expand the dataset to include a broader range of defect types and various inspection objects. This initiative aims to improve the generalization and applicability of the detection model, thereby providing a solid theoretical framework for real-time industrial defect detection.

Author Contributions: Conceptualization, P.S. and Y.Z.; methodology, P.S. and Y.Z.; validation, P.S., Y.Z. and Y.C.; formal analysis, P.S. and Y.Z.; investigation, D.C.; resources, J.S.; writing—original draft preparation, P.S. and Y.Z.; writing—review and editing, P.S., Y.Z., Y.C., J.S., D.C. and L.K.; supervision, P.S. and Y.Z.; project administration, P.S., Y.Z. and L.K.; funding acquisition, P.S., Y.Z. and D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Jiangsu Province Natural Science Project of Institution (grant no. 21KJB520020), the Wuxi Science and Technology Plan Project (grant no. K20221044), the Qing Lan Project of Jiangsu Province, the Wuxi Science and Technology Plan Project (K20231011), the Wuxi University Research Start-up Fund for Introduced Talents (grant no. 550223005), and the Wuxi "Xishan Talent Plan" Innovation Leader Talent Project (2022xsyc002), to which we would like to express our heartfelt thanks.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in this experiment is publicly available.

Acknowledgments: All authors would like to thank the reviewers for their valuable contributions that have enhanced the quality of our research.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhou, Y.; Yuan, M.; Zhang, J.; Ding, G.; Qin, S. Review of vision-based defect detection research and its perspectives for printed circuit board. *J. Manuf. Syst.* **2023**, *70*, 557–578. [CrossRef]
2. Ma, Y.; Yin, J.; Huang, F.; Li, Q. Surface defect inspection of industrial products with object detection deep networks: A systematic review. *Artif. Intell. Rev.* **2024**, *57*, 333. [CrossRef]
3. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
4. Girshick, R.B. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
5. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
6. Kumar, A.; Singh, K. PCB Defect Detection Using Two-Stage Object Detection Network. *J. Electron. Test.* **2021**, *37*, 265–275.
7. Qian, H.; Wang, H.; Feng, S.; Yan, S. FESSD: SSD target detection based on feature fusion and feature enhancement. *J. Real-Time Image Process.* **2023**, *20*, 2. [CrossRef]
8. Diwan, T.; Anirudh, G.; Tembhurne, J.V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimed. Tools Appl.* **2023**, *82*, 9243–9275. [CrossRef]
9. Kumar, A.; Singh, R. PCB Defect Detection Using SSD-Based Deep Learning Approach. *Sensors* **2022**, *22*, 1234.
10. Liao, X.; Lv, S.; Li, D.; Jiang, C. YOLOv4-MN3 for PCB surface defect detection. *Appl. Sci.* **2021**, *11*, 11701. [CrossRef]
11. Ling, Q.; Isa, N.A.M.; Asaari, M.S.M. Precise Detection for Dense PCB Components Based on Modified YOLOv8. *IEEE Access* **2023**, *11*, 116545–116560. [CrossRef]
12. Chen, P.; Xie, F. A machine learning approach for automated detection of critical PCB flaws in optical sensing systems. *Photonics* **2023**, *10*, 984. [CrossRef]
13. Lou, H.; Duan, X.; Guo, J.; Liu, H.; Gu, J.; Bi, L.; Chen, H. DC-YOLOv8: Small-size object detection algorithm based on camera sensor. *Electronics* **2023**, *12*, 2323. [CrossRef]

14. Dang, L.; Huangfu, P.; Hou, Y.E.; Liu, Y.; Han, H. A path aggregation network based on residual feature enhancement for object detection in remote sensing imagery. *Remote Sens. Lett.* **2023**, *14*, 598–608. [CrossRef]
15. Chen, J.; Ji, C.; Zhang, J.; Feng, Q.; Li, Y.; Ma, B. A method for multi-target segmentation of bud-stage apple trees based on improved YOLOv8. *Comput. Electron. Agric.* **2024**, *220*, 108876. [CrossRef]
16. Xiao, Q.; Huang, J.; Huang, Z.; Li, C.; Xu, J. Transparent Component Defect Detection Method Based on Improved YOLOv7 Algorithm. *Int. J. Pattern Recognit. Artif. Intell.* **2023**, *37*, 2350030. [CrossRef]
17. Wang, Q.; Nie, Z.; Liu, H. Assessment of Power System Stability During Transients Using Deep Residual Shrinkage Network and CBAM Integration. *J. Circuits Syst. Comput.* **2024**, *33*, 2450246. [CrossRef]
18. Cao, Y.; Pang, D.; Zhao, Q.; Yan, Y.; Jiang, Y.; Tian, C.; Wang, F.; Li, J. Improved YOLOv8-GD deep learning model for defect detection in electroluminescence images of solar photovoltaic modules. *Eng. Appl. Artif. Intell.* **2024**, *131*, 107866. [CrossRef]
19. Zhang, G.; Liu, S.; Nie, S.; Yun, L. YOLO-RDP: Lightweight Steel Defect Detection through Improved YOLOv7-Tiny and Model Pruning. *Symmetry* **2024**, *16*, 458. [CrossRef]
20. Cai, S.; Zhang, X.; Mo, Y. A Lightweight underwater detector enhanced by Attention mechanism, GSConv and WIoU on YOLOv8. *Sci. Rep.* **2024**, *14*, 9558. [CrossRef]
21. Yan, J.; Zeng, Y.; Lin, J.; Pei, Z.; Fan, J.; Fang, C.; Cai, Y. Enhanced Object Detection in Pediatric Bronchoscopy Images Using YOLO-Based Algorithms with CBAM Attention Mechanism. *Heliyon* **2024**, *10*, e32678. [CrossRef]
22. Yang, D.; Solihin, M.I.; Ardiyanto, I.; Zhao, Y.; Li, W.; Cai, B.; Chen, C. Author Correction: A streamlined approach for intelligent ship object detection using EL-YOLO algorithm. *Sci. Rep.* **2024**, *14*, 15254. [CrossRef]
23. Hu, D.; Yu, M.; Wu, X.; Hu, J. DGW-YOLOv8: A small insulator target detection algorithm based on deformable attention backbone and WIoU loss function. *IET Image Process.* **2024**, *18*, 1096–1108. [CrossRef]
24. Alnaggar, O.A.M.F.; Jagadale, B.N.; Saif, M.A.N.; Ghaleb, O.A.M.; Ahmed, A.A.Q.; Aqlan, H.A.A.; Al-Ariki, H.D.E. Efficient artificial intelligence approaches for medical image processing in healthcare: Comprehensive review, taxonomy, and analysis. *Artif. Intell. Rev.* **2024**, *57*, 221. [CrossRef]
25. Ali, A.; Khan, M.; Khan, K.; Khan, R.U.; Aloraini, A. Sentiment Analysis of Low-Resource Language Literature Using Data Processing and Deep Learning. *Comput. Mater. Contin.* **2024**, *79*, 713–733. [CrossRef]
26. Li, Y.; Zhang, X. Multi-modal deep learning networks for rgb-d pavement waste detection and recognition. *Waste Manag.* **2024**, *177*, 125–134. [CrossRef]
27. Fan, J.; Gao, B.; Ge, Q.; Ran, Y.; Zhang, J.; Chu, H. SegTransConv: Transformer and CNN Hybrid Method for Real-Time Semantic Segmentation of Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 1586–1601. [CrossRef]
28. Kennedy, A.C.; Douglass, M.J.J.; Santos, A.M.C. Being certain about uncertainties: A robust evaluation method for high-dose-rate prostate brachytherapy treatment plans including the combination of uncertainties. *Phys. Eng. Sci. Med.* **2023**, *46*, 1115–1130. [CrossRef]
29. Gong, L.; Yang, Y.; Feng, S.; Dai, W.; Liang, B.; Xiong, J. Solar active regions detection and tracking based on deep learning. *Sol. Phys.* **2024**, *299*, 121. [CrossRef]
30. Yang, Y.; Jiao, G.; Liu, J.; Zhao, W.; Zheng, J. A lightweight rice disease identification network based on attention mechanism and dynamic convolution. *Ecol. Inform.* **2023**, *78*, 102320. [CrossRef]
31. Yue, G.; Liu, Y.; Niu, T.; Liu, L.; An, L.; Wang, Z.; Duan, M. Glu-yolov8: An improved pest and disease target detection algorithm based on yolov8. *Forests* **2024**, *15*, 1486. [CrossRef]
32. Jia, S.; Du, L.; Xinyu, J.; Ben, L.; Kangkang, F. PCB defect detection algorithm based on YOLO-G. *Microelectron. Comput.* **2024**, *41*, 35–44.
33. Yan, W.; Jian, L.; Jin, T.; Hong, P.; Siyi, C. Defect detection of printed circuit boards based on YOLOv8-PCB. *Laser Optoelectron. Prog.* **2024**, *14*. Available online: https://kns.cnki.net/kcms/detail/31.1690.TN.20240624.2254.060.html (accessed on 25 June 2024).

applied sciences

Article

Application of Machine Learning Algorithms in Real-Time Monitoring of Conveyor Belt Damage

Damian Bzinkowski [1], Miroslaw Rucki [2,*], Leszek Chalko [1], Arturas Kilikevicius [2], Jonas Matijosius [2], Lenka Cepova [3] and Tomasz Ryba [1]

1 Faculty of Mechanical Engineering, Casimir Pulaski Radom University, Stasieckiego 54, 26-600 Radom, Poland; leszek.chalko@urad.edu.pl (L.C.)
2 Institute of Mechanical Science, Vilnius Gediminas Technical University, Sauletekio al. 11, LT-10223 Vilnius, Lithuania; jonas.matijosius@vilniustech.lt (J.M.)
3 Faculty of Mechanical Engineering, VSB-Technical University of Ostrava, 17. listopadu 2172/15, 70800 Ostrava, Czech Republic
* Correspondence: m.rucki@urad.edu.pl

Featured Application: This work can potentially be applied to industrial belt conveyors of any type. The tested system can be used for real-time monitoring in order to identify and prevent overloads, misalignments, growing damage to the belt in the early stages, and other trends that may cause failure.

Abstract: This paper is devoted to the real-time monitoring of close transportation devices, namely, belt conveyors. It presents a novel measurement system based on the linear strain gauges placed on the tail pulley surface. These gauges enable the monitoring and continuous collection and processing of data related to the process. An initial assessment of the machine learning application to the load identification was made. Among the tested algorithms that utilized machine learning, some exhibited a classification accuracy as high as 100% when identifying the load placed on the moving belt. Similarly, identification of the preset damage was possible using machine learning algorithms, demonstrating the feasibility of the system for fault diagnosis and predictive maintenance.

Keywords: machine learning; real-time monitoring; belt conveyor; fault diagnosis; predictive maintenance

1. Introduction

It can be stated that belt conveyors are the most common type of conveyor since they are relatively cheap, versatile, and easy to maintain [1,2]. There are systems consisting of a single belt conveyor, but there are also long transport lines with a number of conveyors [3]. Belts are a crucial component of conveyor systems [4], which are subject to degradation and failure under normal work conditions due to the dynamic contact between the transported material on the belt's surface and between the belt and its supporting components [5]. Degradation is increased and accelerated under multiple physical, chemical, or biological factors [6], but also under thermal shocks, humidity, or aging [7]. Apart from safety hazards, unplanned shutdowns lead to significant losses [8]. Thus, degradation must be continuously monitored by collecting real-time data from various sensors [9]. This task is not trivial since belts are one of the most complex and most difficult components to diagnose [10]. In general, there are two methods for the detection of conveyor belt damage: (i) contact methods with an extra mechanical component embedded into the conveyor belt's components and (ii) non-contact methods like the ones based on machine vision [11] or acoustic emission analysis [12]. Monitoring the state and condition of the conveyor belt during its work is of great importance because it enables one to discover the appearance of possible damage and failure early on [13]. From the published data [14], it is known that expenses on conveyor monitoring equipment were, in 2018, as high as USD 200 million, and they are expected to exceed USD 0.25 trillion in 2024.

However, apart from collecting the signal from a monitoring system conveying the belt's actual condition, a proper analysis of the registered signals should be performed, and the relevant decision should follow [15]. Among others, machine learning (ML) algorithms are widely applied for this purpose. For instance, Andrejiova and co-authors attempted to identify the correlations between significant damage to the rubber–textile material of the conveyor belt and some parameters, such as the type of material and the impact height [16]. Using multispectral imaging, Zhou et al. assessed a conveyor belt's wear condition, classifying it into three wear states using a deep learning approach [17]. In order to prevent the belt from flipping over, Rumin, together with the team, proposed an algorithm based on machine learning and analytical algorithms to determine the forces acting on the individual components of the conveyor and subsequently make a decision on classifying the category of the obtained results [18]. Zhang and co-authors reported the application of a new detection method able to simultaneously detect multiple faults using a special dataset for various types of damage to the conveyor belt [19]. Guo and colleagues proposed a novel method for conveyor belt damage detection based on fusion knowledge distillation and a deep neural network using the data collected via an image acquisition module [20]. Signals obtained from both machine vision [21] and acoustic systems [22] were reported to be analyzed with a machine learning algorithm. Pulcini and Modoni proposed a digital twin to perform a predictive maintenance approach through the analysis of data collected from various sensors placed along an operating conveyor belt, exploiting a machine learning algorithm [23]. Also, Soares and co-authors reported the application of a combination of wavelet packet decomposition and gradient boosting decision tree for the diagnosis of failure modes, including the surface wear of laboratory belt conveyor idlers [24].

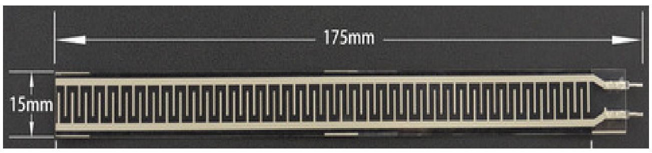
In this paper, a novel system for conveyor belt monitoring is presented based on strain gauges. Strain gauges provide a continuous signal corresponding to the belt's tension in different work conditions. Considering some of the features of the registered signals, an attempt was undertaken to identify them using several machine learning algorithms.

2. Materials and Methods

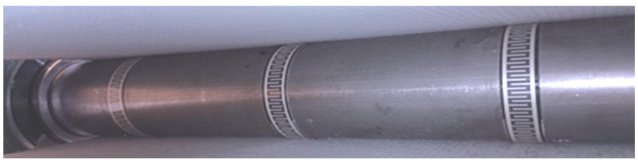
The materials and methods consisted of a model of the belt conveyor, a measurement system for real-time monitoring, and machine learning algorithms that helped to classify the loads and damage types of the belt.

2.1. Monitoring System

The proposed monitoring system is a type of contact method for the detection of conveyor belt damage. It is based on strain gauges, exploiting the principle described in [25]. However, for the current research, the system was modified and patented [26]. In this solution, the long, flexible RP-L-170-type thin-film sensor [27] produced by DFRobot (Chengdu, China) was applied. Figure 1a illustrates the strain gauge, and Figure 1b shows its position on the tail pulley surface.



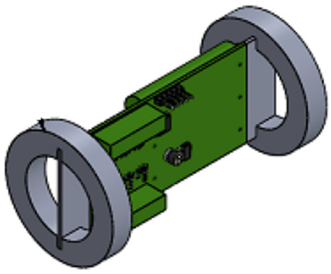
(**a**)



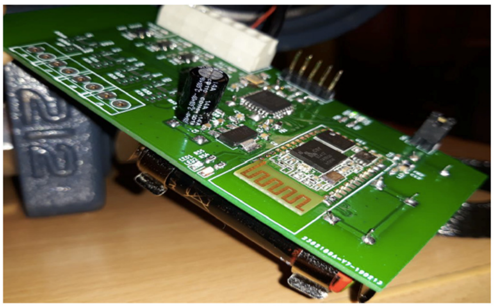
(**b**)

Figure 1. The strain gauges used in the conveyor belt monitoring system: (**a**) view and dimensions; (**b**) the positioning of three strain gauges on the tail pulley surface.

The resistance R_T of a strain gauge is directly dependent on the pressing force applied by the belt $R_T = (F_n)$. To register the signal from the strain gauges, a special electronic unit was projected and built into the hollow tail pulley. Figure 2a shows the holder of the printed circuit board (PCB) fitted to the inside of the pulley. Figure 2b shows the PCB designed to collect, transmit, and process the signals initially so that the results can be presented in a remote laptop in the form of graphs and data tables for further analysis.



(**a**)



(**b**)

Figure 2. Details of the built-in electronic system: (**a**) a drawing of the PCB holder placed inside the tail pulley; (**b**) the printed circuit ready to be mounted into the pulley.

In principle, the electronic system was similar to the one described in [25]. Its tasks were to collect signals generated by the force F between the conveyor belt and the strain gauges placed on the pulley surface, to transmit the signal through the Bluetooth port, and to process collected data with a dedicated LabView program. The entire system underwent calibration as described in [28], and the Type A expanded uncertainty was below 1% of the measured value for the 99% gradient boosting decision tree, while the approximation error was ca. 8%. These results cover all of the uncertainty sources, including the electronic unit. The monitoring system was expected to support a predictive maintenance strategy able to safeguard the continuous operation of the conveyor and to minimize supply chain disruptions in agreement with the Industry 4.0 concept [29].

2.2. The Test Model of the Belt Conveyor

In order to perform the necessary experiments, the test rig was built. It consisted of a driving pulley with an adjustable electric motor and a tail pulley with a built-in strain gauge measuring system. Figure 3 demonstrates the end part of the test rig with the belt placed on the tail pulley.

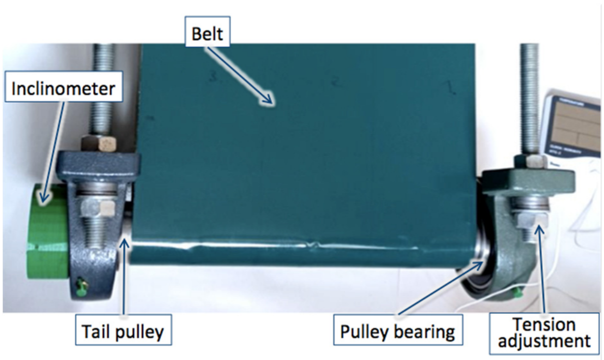


Figure 3. A view of the test rig with the model of a belt conveyor.

The rubber belt enabled us to imitate the work conditions of a belt conveyor with different speeds, tensions, loads, and even damages. The inclinometer allowed for the registration of the rotational angle of the pulley, which was important because only part of the strain gauge underwent pressure.

In the experiments, a typical belt material, EDV08PB-AS 2.0, was used, made by Enitra Sp. z o.o. (Wałbrzych, Poland). Before the experiments, the material underwent a detailed analysis reported elsewhere [30]. Its important feature is the anisotropy of the strength characteristics, especially the ones related to damages. In particular, the samples with transverse preset damages exhibited ca. 60% lower breaking force than the ones with longitudinal preset damages.

2.3. Damages Identification

The work of the belt conveyor was imitated using the abovementioned test rig. The test campaign first included the work of the belt without loads and damages, and then the loaded work, and, finally, the work with intentionally made damages. The tests were performed at three different rotary speeds of the tail pulley, namely, n_1 = 159 rpm, n_2 = 318 rpm, and n_3 = 520 rpm. Finally, the machine learning (ML) algorithms were tested for the rotary speed of n_1 = 159 rpm, which corresponded with a linear belt speed of v_1 = 0.5 m/s.

Since it is widely recognized that longitudinal and transverse tears belong to the main types of catastrophic failures, and the longitudinal ones appear more commonly than transverse ones [31], the damages were made in two stages as follows:

- First, three cuts were made, namely, two longitudinal cuts through UW I and UW II, which were 50 mm and 70 mm long, respectively, and one partial longitudinal cut UW III, which was 45 mm long and 1 mm deep.
- Then, two more cuts were added, namely, a longitudinal partial cut through UW IV which was 50 mm long and 1.5 mm deep, and a transverse cut through UP I, which was 10 mm long in the middle of the belt.

Thus, three different states of the belt were analyzed: the undamaged one, the one with three cuts, and the one with five cuts.

In order to avoid additional uncertainty propagation due to the calculations, rounding, and approximations, the data were analyzed in analog-to-digital units (ADUs) obtained from the analog-to-digital converter. It should be noted that a higher value in ADUs corresponded with smaller pressure on the strain gauge.

To train a machine learning (ML) model, usually a batch of data are used [32]. The respective data collected from the electronic system were analyzed from the perspective of belt state monitoring. Statistical parameters were calculated, including standard deviation and kurtosis, and the data were imported to MatLAb (R2024b). From the *Classification Lerner* application, 10 algorithms were used, and a typical number of folds for the cross-validation was chosen, namely, k_v = 5 [33].

Then, the analysis of 18 features was performed with the *Diagnostic Feature Designer* application. The purpose of the test was to determine which features are the most significant for machine learning, and a one-way analysis of variance (ANOVA) model [34] was found to be useful for this analysis.

3. Results and Discussion

The results were analyzed in two stages. First, the graphical visualization of the registered signals in time t was studied, comparing the simultaneous indications of three strain gauges in order to assess the features related directly to the loads and damages. Then, the ML algorithms were tested, assessing their ability to classify the loads and damages.

3.1. Registered Monitoring Signal

An example of a registered signal is shown in Figure 4. It corresponds with the tail pulley rotary speed of n_1 = 159 rpm. On the left, the strain gauge indications are expressed in ADUs, while on the right, they are recalculated to the force units N. However, it should be noted that these are the values of the force pressing directly on the strain gauge surface, not to be confused with the belt tension or the belt pressure on the pulley.

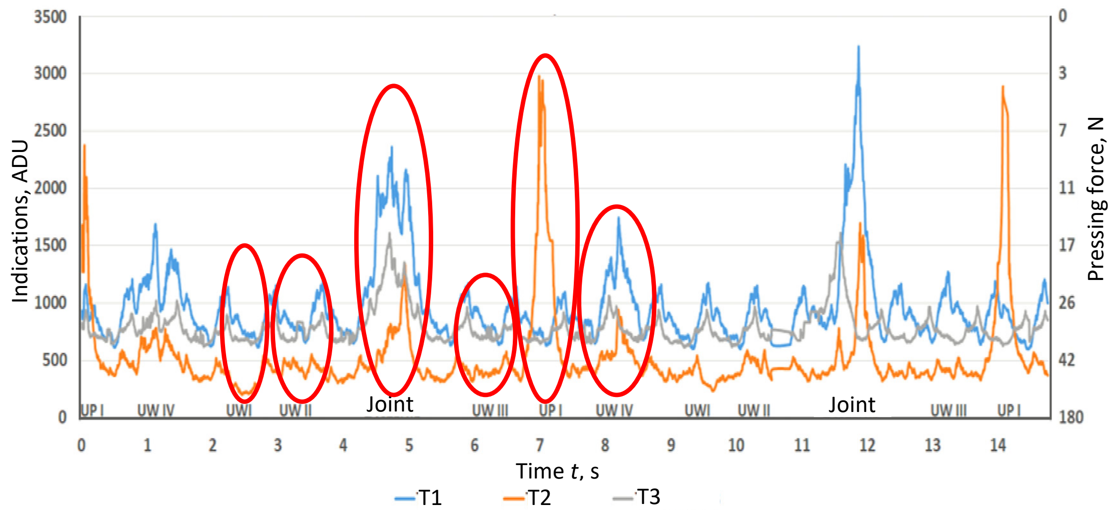


Figure 4. An example of the registered signal from strain gauges T1, T2, and T3.

During the measurement time, some typical registered characteristics appeared, which can be seen in the graph in relation to the damages. The most noticeable one is the "jump" of all three strain gauge indications when the belt joint was in contact with the tail pulley after the fourth second of measurement. At that moment, the indication of strain gauge T1 jumped from ca. 700 ADU to ca. 2400 ADU, which corresponded with the drop in the pressing force. Quite predictably, the central strain gauge, T2, experienced a smaller drop in the pressing force, which corresponded with the increase in indications from 400 ADU to 1400 ADU.

Another obvious characteristic is the one related to the cut UP I in the middle of the belt width. This one reached the pulley at the seventh second of the experiment and caused a significant drop in the pressing force on the central strain gauge, T2, from ca. 42 N to almost 3 N. Gauge T2 was directly under the cut, so the pressing force was very small. Other preset damages had different impacts on each strain gauge, making it possible to identify their presence from a comparison of the three signals.

3.2. The Application of the Machine Learning Algorithms

The possibility of damage identification was tested using statistical data from the registered signal of each strain gauge, namely, T1, T2, and T3, for the belt without damage, the one with three defects, and the one with five defects, respectively, as described in Section 2.3. The statistical parameters were as follows: The mean value for each strain gauge is

$$\overline{x} = \frac{1}{N}\sum\nolimits_{n=1}^{N}|x(n)|, \tag{1}$$

where $x(n)$ is the amplitude of each sampling, and N is the number of samples.

The respective equations for the mean square value X_{ms}, standard deviation σ, and kurtosis X_k for each strain gauge were used as follows:

$$X_{ms} = \sqrt{\frac{1}{N}\sum\nolimits_{n=1}^{N}[x(n)]^2}, \tag{2}$$

$$\sigma = \sqrt{\frac{1}{N}\sum\nolimits_{n=1}^{N}[x(n)-\overline{x}]^2}, \tag{3}$$

$$X_k = \frac{\frac{1}{N}\sum_{n=1}^{N}[x(n)-\overline{x}]^4}{\left(\frac{1}{N}\sum_{n=1}^{N}[x(n)-\overline{x}]^2\right)^2}. \tag{4}$$

Thus, with six statistical parameters for three strain gauges, 20 repetitions provided 360 statistical datapoints covering three states of the belt. Table 1 contains the data for repetitions No. 1, 19, and 20 for each parameter, calculated for each strain gauge. Table 2 presents the list of the algorithms tested, specifying the accuracy of damage identification and the respective numbers of the cases erroneously classified.

Table 1. The characterization of the collected signals from three strain gauges, namely, T1, T2, and T3, after 20 repetitions, ADU.

Repetition		1	...	19	20	1	...	19	20	1	...	19	20
State		**Undamaged Belt**				**3 Defects on the Belt**				**5 Defects on the Belt**			
Mean $\overline{x}$	T1	862.6	...	831.3	765.2	1044.6	...	959.3	1021.4	1440.3	...	1331.3	1419.4
	T2	406.3	...	317.5	316.2	442.1	...	385.3	430.0	668.1	...	580.9	668.3
	T3	735.9	...	750.3	761.4	787.6	...	751.5	773.8	856.0	...	843.7	863.4
Min	T1	569	...	520	487	619	...	637	642	828	...	799	848
	T2	223	...	213	222	221	...	193	144	247	...	239	209
	T3	580	...	602	615	660	...	625	642	670	...	592	671
Max	T1	1330	...	1298	1376	2209	...	1553	1907	2987	...	2306	2505
	T2	845	...	519	538	1301	...	1119	1087	3429	...	3052	3073
	T3	1136	...	1066	1140	1030	...	1069	1081	1322	...	1337	1371
Mean square X_{ms}	T1	876.4	...	772.9	555.9	923.7	...	932.2	835.4	1460.1	...	1294.9	1423.5
	T2	414.7	...	293.6	226.0	411.7	...	391.8	368.7	784.7	...	629.3	755.8
	T3	744.8	...	690.8	543.6	679.5	...	725.3	626.1	850.0	...	815.0	860.8
Std. dev. σ	T1	155.0	...	160.4	182.3	253.7	...	154.2	192.6	361.1	...	253.5	275.3
	T2	82.9	...	51.1	47.5	184.8	...	135.5	159.8	436.5	...	310.5	377.9
	T3	114.6	...	98.6	109.0	70.7	...	82.3	89.8	120.5	...	125.2	137.7
Kurtosis X_k	T1	−0.5	...	−0.7	−0.1	4.3	...	1.0	2.5	2.2	...	1.9	0.3
	T2	3.6	...	0.6	2.3	3.3	...	5.4	1.5	13.2	...	18.3	10.2
	T3	0.1	...	0.1	0.5	0.4	...	1.0	0.8	1.2	...	0.4	0.1

Table 2. Machine learning results for different algorithms.

Algorithm	Classification Accuracy	Number of Wrongly Classified Cases
Fine Tree	100.00%	0
Medium Tree	100.00%	0
Coarse Tree	100.00%	0
Ensemble Bagged Trees	100.00%	0
Ensemble Subspace KNN	98.33%	1
Linear Discriminant	96.67%	2
Coarse Gaussian SVM	96.67%	2
Coarse KNN	33.33%	40
Ensemble Boosted Trees	33.33%	40
Ensemble RUSBoosted trees	33.33%	40

Notably, four algorithms based on a decision tree reached 100% of damage identification. It is known that "A decision tree is a hierarchical model composed of discriminant functions, or decision rules, that are applied recursively to partition the feature space of a dataset into pure, single class subspaces" [35].

As shown in Figure 5, decision trees have branch nodes represented by triangles, and each of them is the parent to two children leaf nodes represented by circles. A line between the parent and children nodes represents a decision pathway. In the present research, the single feature was T1_mean, with the boundary values of the features being 880.604 and 1194.04 ADU. The inequality expressions given close to the branch nodes describe the decision boundaries.

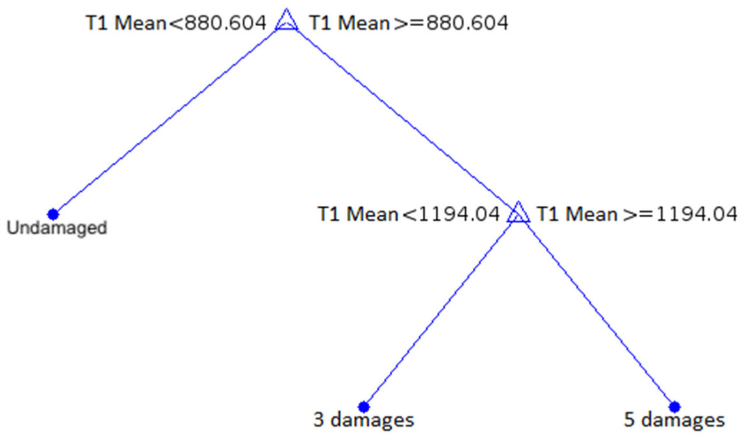


Figure 5. Decision tree.

Thus, if the value of the new analyzed sample is smaller than 880.604 ADU, the sample travels the pathway and is directed to the leaf node to be classified as "undamaged" based on the set of training samples. Usually, "the class represented by the majority of training samples within the leaf node subspace dictates sample classification" [35].

The analysis of 18 features was performed using the *Diagnostic Feature Designer* application and a one-way ANOVA. This sort of test requires the assumptions of a normal distribution and homogeneity of variance for all groups. To test the effect of a single factor on the results, it assumes the presence of one independent variable, represented by the groups studied in the experiments, and one dependent variable, represented by the results of the study. Hypotheses regarding the equality of means are tested if there are more than two analyzed groups [36].

When the abovementioned assumptions are not met, the result may become unreliable. Data transformation can be a method that allows for data normalization and variance stabilization. However, when such actions do not provide a positive result, non-parametric tests become an alternative, e.g., the Kruskal–Wallis test, which is equivalent to a one-way analysis of variance [37]. The non-parametric Kruskal–Wallis test corresponds with the parametric one-way ANOVA test and allows for post hoc testing when the null hypothesis of equality of means in the analyzed groups is rejected. When the assumptions of a parametric test are met and the distribution is normal, parametric tests appear to be more powerful than the comparable non-parametric ones, i.e., a non-parametric test is less likely to correctly define a statistically significant result [38]. Thus, it is recommended to first apply a parametric test, and if it appears improper, then a non-parametric test should be used [37].

In this research, we decided to use the Laplacian Score, variance, and monotonicity tests. The Laplacian Score used the nearest neighboring graph to determine the local data structure and the respective value for each feature. A feature with a higher Laplacian Score indicated its importance for localization [39]. Variance is the classical method of differentiation for its evaluation and has no statistical interpretation. However, it can be used for the construction of other parameters, such as central moments or the standard deviation. It is expressed as the arithmetic mean of the squares of deviations of individual values of the feature from their arithmetic mean [40]. Monotonicity is the rating method available in *Diagnostic Feature Designer*, with values ranging from 0 to 1 to describe the trend of the given function during the system's evolution.

The results of the tests obtained from the five different abovementioned models are shown in Figure 6, ranked in decreasing order for different features according to the results of the one-way ANOVA. Generally, all of the tests indicated similar trends in significance, even though the particular results differed greatly from one model to another.

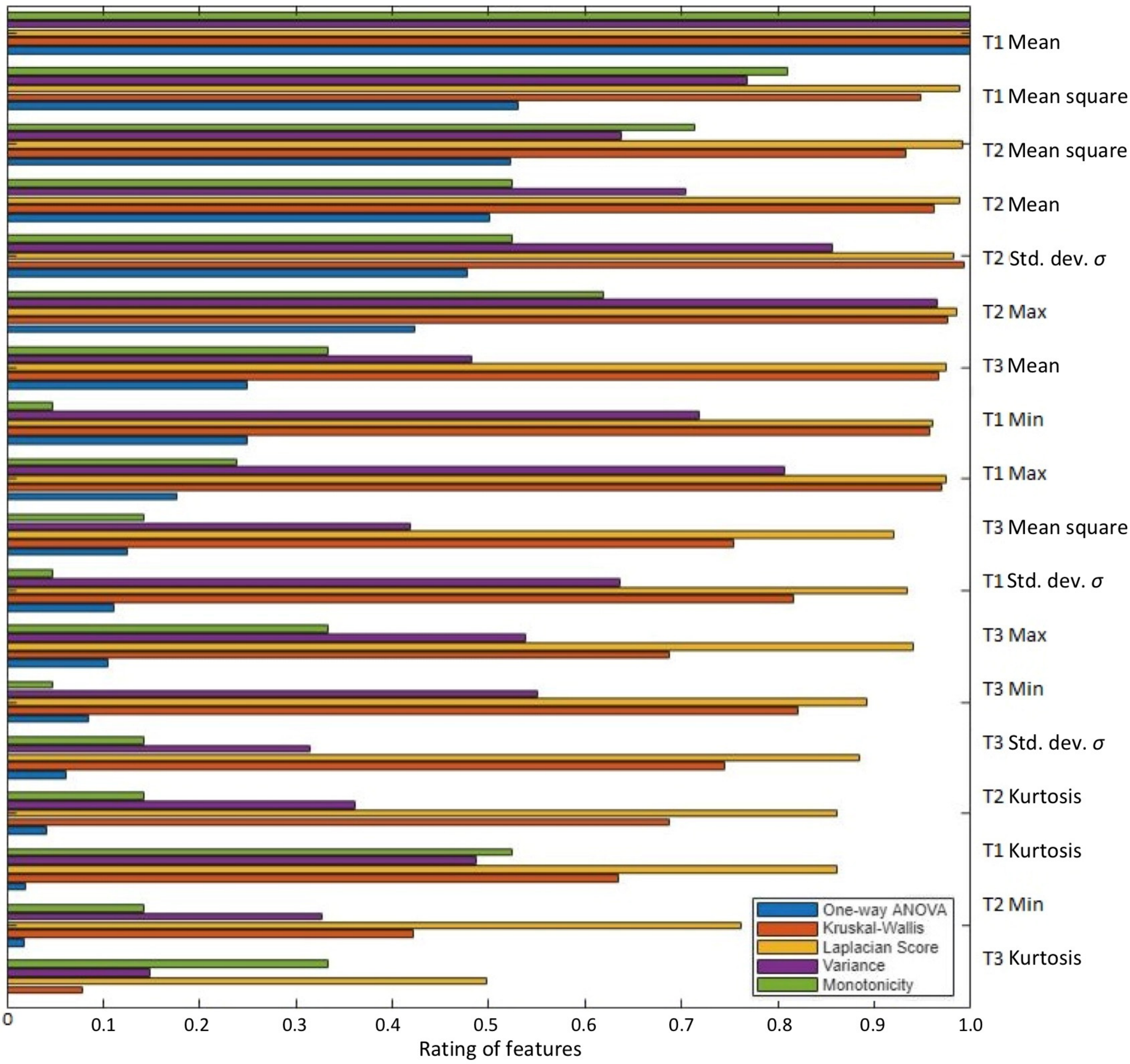
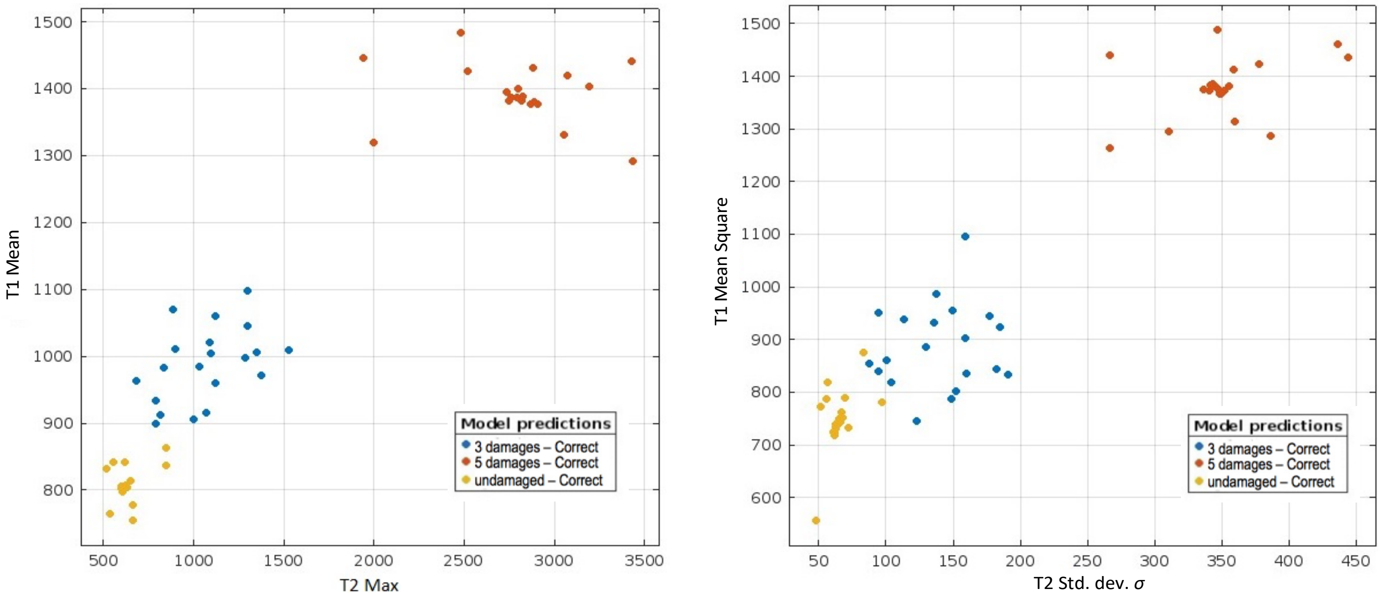


Figure 6. The ratings of the results for different features evaluated with different models.

It is seen in Figure 6 that the mean value from the T1 strain gauge was the most significant feature, which was confirmed by all four models applied. In turn, the kurtosis of the T3 strain gauge indications was found to be the least significant for the identification of the conveyor belt state. On the other hand, Laplacian Score method provided the highest values for test results, with one exception being the T2 Std.dev., while the ANOVA method gave the lowest values for the tested parameters with three exceptions.

Figure 7a,b show examples of distribution graphs for the features found to be highly significant. In Figure 7a, the results for the T2 Max and T1 Mean are collected, while in Figure 7b, the T2 standard deviation and T1 Mean Square points are presented. In both diagrams, the areas of concentration of the points can be clearly distinguished. These points were used for state identification and damage classification for the analyzed belt.



(**a**) (**b**)

Figure 7. Examples of dispersion diagrams for significant features: (**a**) T1 Mean and T2 Max; (**b**) T1 Mean Square and T2 standard deviation.

For better visualization of the performance of the tested algorithms, the results are also presented in the form of a confusion matrix in Figure 8. It represents the models that reached 100% of classification for all three cases, including the undamaged belt, the one with three defects, and the one with five defects. Each row represents the results of actual classification, and each column corresponds with a predicted class. For the cross-validation, the confusion matrix was calculated using the predictions for the validation observations.

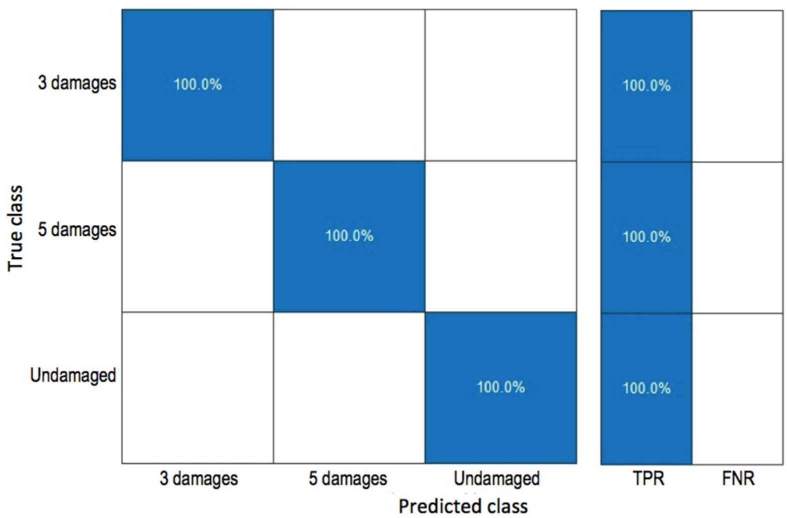


Figure 8. Confusion matrix for models that reached 100% classification.

The cells located diagonally in the matrix indicate where the actual and predicted classes match. In order to check the performance of the classifier in individual classes, the function was selected, showing the true positive rates (TPRs) and false negative rates (FNRs). The TPR represents the percentage of correctly classified observations per true class, while the FNR shows the percentage of incorrectly classified observations per true class. The obtained confusion matrix confirmed the possibility of identifying the state of the conveyor belt from the analysis of signals generated by the strain gauge-based monitoring system.

In order to compare the features of the obtained belt state observations and to illustrate the relationships between them, a parallel coordinate plot (PCP) is presented in Figure 9. PCPs are widely recognized as a powerful technique allowing for the visual analysis of large sets reaching some millions of datapoints with multiple parameters associated with them [41]. The range scaling method was used to make the data fit into a certain pre-defined interval with the same minimum and maximum limits. The graph shows predictions for the *Coarse Tree* model using eight features that exhibited the highest significance.

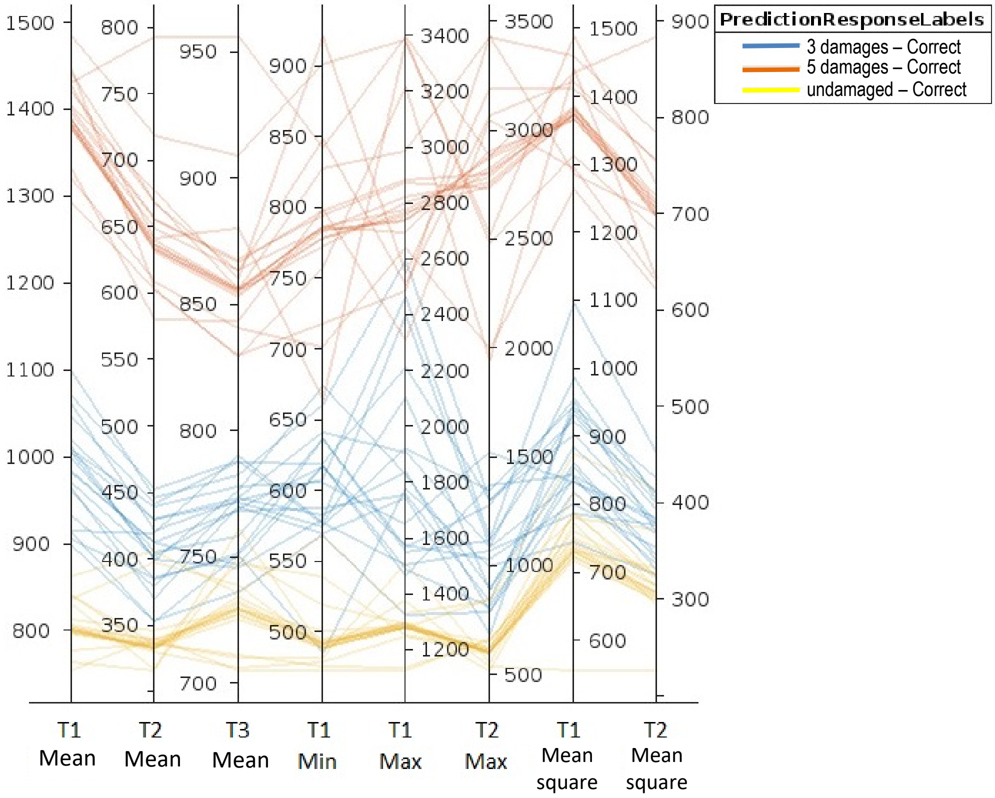


Figure 9. The parallel coordinate plot for the *Coarse Tree* model.

The most significant features were specified based on Figure 9. These features precisely show the ranges of values for three different states of the conveyor belt. The variables indicating a belt with five defects are marked in red. Apart from a dozen values, one distinctive area can be seen in the upper part of the graph. The values in the middle part have the greatest dispersion, and this area corresponds with the belt with three defects. The belt without damages is characterized by the variables in the lower part of the graph, which are marked in yellow and exhibited the smallest dispersion. In the graph, three areas corresponding with the states of the conveyor belt can be distinguished clearly, which overlapped only to a small extent.

The results indicate that the data collected from the strain gauge-based monitoring system are closely related to the actual state of the conveyor and the reflected condition of the rubber belt. As such, these data can also be analyzed for the speed and quality aspects of the manufacturing process in terms of a production efficiency analysis [42]. From the collection of records, major faults and critical failures can be assessed using the Fault Tree Analysis (FTA) method through a graphical representation of the faults' causes and potential countermeasures [43]. Moreover, continually collected data can be used for the prediction of the remaining useful life and for appropriate optimized maintenance decisions [44].

4. Conclusions

The analysis proved that the strain gauge-based measurement system provided a reliable sequence of signals for the real-time monitoring of the conveyor belt work. As a result of continuous monitoring of the belt operation, the system generated a significant amount of data that underwent an extended analysis based on the machine learning (ML) algorithms. The signals were collected from three strain gauges, namely, T1, T2, and T3, after 20 repetitions, and statistical parameters were calculated accordingly. It was found that for damage monitoring, the most significant features were the arithmetic mean value from the T1 strain gauge, followed by the T1 Root Mean Square, T2 Root Mean Square T2 Mean, and T2 standard deviation. The results indicate the existence of an identifiable correlation between the actual condition of the rubber belt and the signal recorded by the strain gauge system. So, it was experimentally proven that the strain gauge-based system could be successfully used for the real-time monitoring of the conveyor belt operation.

A further analysis of 18 features was performed using 10 ML algorithms. Four of them were able to correctly classify the undamaged belt, the belt with three preset defects, and the one with five defects. Thus, it can be concluded that the system provided real-time information enabling the detection of the belt damages that appeared during the conveyor operation. It is planned to test more ML algorithms in future research. Moreover, it is important to investigate the possibility of detecting damages appearing and dynamically developing during the belt's operation in order to prevent failures.

5. Patents

The work reported in this manuscript resulted in the Polish patent No. P.447569 Method and a device for the supervision of the tension and wear of conveyor rubber belts.

Author Contributions: Conceptualization, D.B., T.R. and A.K.; methodology, D.B., T.R., M.R. and L.C. (Lenka Cepova); software, L.C. (Leszek Chalko); validation, L.C. (Leszek Chalko), A.K. and L.C. (Lenka Cepova); formal analysis, D.B., L.C. (Leszek Chalko), M.R. and L.C. (Lenka Cepova); investigation, T.R. and J.M.; resources, T.R. and J.M.; data curation, L.C. (Leszek Chalko); writing—original draft preparation, M.R.; writing—review and editing, all authors; visualization, M.R., A.K. and J.M.; supervision, D.B.; project administration, M.R.; funding acquisition, D.B. and T.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Moran, S. *Process Plant Layout*, 2nd ed.; Butterworth-Heinemann: Amsterdam, The Netherlands, 2017; pp. 471–481. [CrossRef]
2. Borucka, A.; Kozłowski, E.; Parczewski, R.; Antosz, K.; Gil, L.; Pieniak, D. Supply Sequence Modelling Using Hidden Markov Models. *Appl. Sci.* **2023**, *13*, 231. [CrossRef]
3. Subba Rao, D.V. *The Belt Conveyor: A Concise Basic Course*; CRC Press: London, UK, 2021.
4. Hou, C.; Qiao, T.; Dong, H.; Wu, H. Coal flow volume detection method for conveyor belt based on TOF vision. *Measurement* **2024**, *229*, 114468. [CrossRef]
5. Ilanković, N.; Živanić, D.; Zuber, N. The Influence of Fatigue Loading on the Durability of the Conveyor Belt. *Appl. Sci.* **2023**, *13*, 3277. [CrossRef]
6. Bortnowski, P.; Kawalec, W.; Król, R.; Ozdoba, M. Types and causes of damage to the conveyor belt—Review, classification and mutual relations. *Eng. Fail. Anal.* **2022**, *140*, 106520. [CrossRef]
7. Rudawska, A.; Madleňák, R.; Madleňáková, L.; Droździel, P. Investigation of the Effect of Operational Factors on Conveyor Belt Mechanical Properties. *Appl. Sci.* **2020**, *10*, 4201. [CrossRef]

8. Zheng, H.; Wu, H.; Yin, H.; Wang, Y.; Shen, X.; Fang, Z.; Ma, D.; Miao, Y.; Zhou, L.; Yan, M.; et al. Novel mining conveyor monitoring system based on quasi-distributed optical fiber accelerometer array and self-supervised learning. *Mech. Syst. Signal Process.* **2024**, *221*, 111697. [CrossRef]
9. Chamorro, J.; Vallejo, L.; Maynard, C.; Guevara, S.; Solorio, J.A.; Soto, N.; Singh, K.V.; Bhate, U.; Ravi Kumar, G.V.V.; Garcia, J.; et al. Health monitoring of a conveyor belt system using machine vision and real-time sensor data. *CIRP J. Manuf. Sci. Technol.* **2022**, *38*, 38–50. [CrossRef]
10. Kozłowski, T.; Wodecki, J.; Zimroz, R.; Błażej, R.; Hardygóra, M. A Diagnostics of Conveyor Belt Splices. *Appl. Sci.* **2020**, *10*, 6259. [CrossRef]
11. Zeng, F.; Zhang, S.; Wang, T.; Wu, Q. Mini-Crack Detection of Conveyor Belt Based on Laser Excited Thermography. *Appl. Sci.* **2021**, *11*, 10766. [CrossRef]
12. Wang, Y.; Miao, C.; Liu, Y.; Meng, D. Research on a sound-based method for belt conveyor longitudinal tear detection. *Measurement* **2022**, *190*, 110787. [CrossRef]
13. Fedorko, G. Application possibilities of virtual reality in failure analysis of conveyor belts. *Eng. Fail. Anal.* **2021**, *128*, 105615. [CrossRef]
14. Kirjanów-Błażej, A.; Jurdziak, L.; Błażej, R.; Rzeszowska, A. Calibration procedure for ultrasonic sensors for precise thickness measurement. *Measurement* **2023**, *214*, 112744. [CrossRef]
15. Zhang, M.; Jiang, K.; Cao, Y.; Li, M.; Wang, Q.; Li, D.; Zhang, Y. A new paradigm for intelligent status detection of belt conveyors based on deep learning. *Measurement* **2023**, *213*, 112735. [CrossRef]
16. Andrejiova, M.; Grincova, A.; Marasova, D. Identification with machine learning techniques of a classification model for the degree of damage to rubber-textile conveyor belts with the aim to achieve sustainability. *Eng. Fail. Anal.* **2021**, *127*, 105564. [CrossRef]
17. Zhou, M.; Chen, Y.; Hu, F.; Lai, W.; Gao, L. A deep learning approach for accurate assessment of conveyor belt wear state based on multispectral imaging. *Opt. Laser Technol.* **2025**, *181*, 111782. [CrossRef]
18. Rumin, P.; Kotowicz, J.; Hogg, D.; Zastawna-Rumin, A. Utilization of measurements, machine learning, and analytical calculation for preventing belt flip over on conveyor belts. *Measurement* **2023**, *218*, 113157. [CrossRef]
19. Zhang, M.; Shi, H.; Zhang, Y.; Yu, Y.; Zhou, M. Deep learning-based damage detection of mining conveyor belt. *Measurement* **2021**, *175*, 109130. [CrossRef]
20. Guo, X.; Liu, X.; Gardoni, P.; Glowacz, A.; Królczyk, G.; Incecik, A.; Li, Z. Machine vision based damage detection for conveyor belt safety using Fusion knowledge distillation. *Alex. Eng. J.* **2023**, *71*, 161–172. [CrossRef]
21. Gao, Y.; Qiao, Y.; Zhang, H.; Yang, Y.; Pang, Y.; Wei, H. A contactless measuring speed system of belt conveyor based on machine vision and machine learning. *Measurement* **2019**, *139*, 127–133. [CrossRef]
22. Liu, X.; Pei, D.; Lodewijks, G.; Zhao, Z.; Mei, J. Acoustic signal based fault detection on belt conveyor idlers using machine learning. *Adv. Powder Technol.* **2020**, *31*, 2689–2698. [CrossRef]
23. Pulcini, V.; Modoni, G. Machine learning-based digital twin of a conveyor belt for predictive maintenance. *Int. J. Adv. Manuf. Technol.* **2024**, *133*, 6095–6110. [CrossRef]
24. Soares, J.L.L.; Costa, T.B.; Moura, L.S.; Sousa, W.S.; Mesquita, A.L.A.; Mesquita, A.L.A.; de Figueiredo, J.M.S.; Braga, D.S. Fault diagnosis of belt conveyor idlers based on gradient boosting decision tree. *Int. J. Adv. Manuf. Technol.* **2024**, *132*, 3479–3488. [CrossRef]
25. Bzinkowski, D.; Ryba, T.; Siemiatkowski, Z.; Rucki, M. Real-time monitoring of the rubber belt tension in an industrial conveyor. *Rep. Mech. Eng.* **2022**, *3*, 1–10. [CrossRef]
26. Ryba, T.; Bzinkowski, D.; Rucki, M. Method and Device for Supervision of the Tension and Wear of the Conveyor Rubber Belts. Polish Patent No. P.447569, 22 January 2024. (In Polish).
27. RP-L-170 Thin Film Pressure Sensor. Available online: https://www.dfrobot.com/product-1843.html (accessed on 22 October 2024).
28. Ryba, T.; Rucki, M.; Siemiatkowski, Z.; Bzinkowski, D.; Solecki, M. Design and calibration of the system supervising belt tension and wear in an industrial feeder. *Facta Univ. Ser. Mech. Eng.* **2022**, *20*, 167–176. [CrossRef]
29. Mallioris, P.; Aivazidou, E.; Bechtsis, D. Predictive maintenance in Industry 4.0: A systematic multi-sector mapping. *CIRP J. Manuf. Sci. Technol.* **2024**, *50*, 80–103. [CrossRef]
30. Ryba, T.; Bzinkowski, D.; Siemiatkowski, Z.; Rucki, M.; Stawarz, S.; Caban, J.; Samociuk, W. Monitoring of Rubber Belt Material Performance and Damage. *Materials* **2024**, *17*, 765. [CrossRef]
31. Wang, G.; Wang, Y.; Sun, H.; Yue, Q.; Zhou, Q. Study on Visual Detection Method of Multi-scale Damage to Conveyor Belt Under Complex Background. *J. Fail. Anal. Prev.* **2024**, *24*, 896–908. [CrossRef]
32. Le-Nguyen, M.H.; Turgis, F.; Fayemi, P.E.; Bifet, A. Real-time learning for real-time data: Online machine learning for predictive maintenance of railway systems. *Transp. Res. Procedia* **2023**, *72*, 171–178. [CrossRef]
33. Beyerer, J.; Richter, M.; Nagel, M. *Pattern Recognition: Introduction, Features, Classifiers and Principles*; De Gruyter: Berlin, Germany, 2017.
34. Fox, W.P.; Sturdivant, R.X. *Probability and Statistics for Engineering and the Sciences with Modeling Using R*; CRC Press: Boca Raton, FL, USA, 2023.

35. Myles, A.J.; Feudale, R.N.; Liu, Y.; Woodyand, N.A.; Brown, S.D. An introduction to decision tree modeling. *J. Chemom.* **2004**, *18*, 275–285. [CrossRef]
36. Nowakowski, M. The ANOVA method as a popular research tool. *Stud. Pr. WNEiZ* **2019**, *55*, 67–77. [CrossRef]
37. Sandurska, E.; Szulc, A. A method of statistical analysis in the field of sports science when assumptions of parametric tests are not violated. *J. Educ. Health Sport* **2016**, *6*, 275–287. [CrossRef]
38. Politi, M.T.; Ferreira, J.C.; Patino, C.M. Nonparametric statistical tests: Friend or foe? *J. Bras. Pneumol.* **2021**, *47*, e20210292. [CrossRef] [PubMed] [PubMed Central]
39. Chandra, B. Gene Selection Methods for Microarray Data. In *Applied Computing in Medicine and Health*; Al-Jumeily, D., Hussain, A., Mallucci, C., Oliver, C., Eds.; Morgan Kaufmann: Waltham, MA, USA, 2016; pp. 45–78.
40. Ręklewski, M. *Descriptive Statistics: Theory and Examples*; Państwowa Uczelnia Zawodowa we Włocławku: Włocławek, Poland, 2020. (In Polish)
41. Stumpfegger, J.; Höhlein, K.; Craig, G.; Westermann, R. GPU accelerated scalable parallel coordinates plots. *Comput. Graph.* **2022**, *109*, 111–120. [CrossRef]
42. Chen, Z.; Xiao, Z.; Sun, Y.; Dong, Y.; Zhong, R.Y. Production efficiency analysis based on the RFID-collected manufacturing big data. *Manuf. Lett.* **2024**, *41*, 81–90. [CrossRef]
43. Bujna, M.; Prístavka, M.; Lee, C.K.; Borusiewicz, A.; Samociuk, W.; Beloev, I.; Malaga-Toboła, U. Reducing the Probability of Failure in Manufacturing Equipment by Quantitative FTA Analysis. *Agric. Eng.* **2023**, *27*, 255–272. [CrossRef]
44. Wang, L.; Li, B.; Zhao, X. Multi-objective predictive maintenance scheduling models integrating remaining useful life prediction and maintenance decisions. *Comput. Ind. Eng.* **2024**, *197*, 110581. [CrossRef]

applied sciences

Article

Early Fault Detection and Operator-Based MIMO Fault-Tolerant Temperature Control of Microreactor

Yuma Morita and Mingcong Deng *

Department of Electrical Engineering and Computer Science, Graduate School of Engineering, Tokyo University of Agriculture and Technology, 2-24-16 Nakacho, Koganei-shi 184-8588, Tokyo, Japan; s244940t@st.go.tuat.ac.jp
* Correspondence: deng@cc.tuat.ac.jp; Tel.: +81-42-388-7134

Abstract: A microreactor is a chemical reaction device that mixes liquids in a very narrow channel and continuously generates reactions. They are attracting attention as next-generation chemical reaction devices because of their ability to achieve small-scale and highly efficient reactions compared to the conventional badge method. However, the challenge is to design a control system that is tolerant of faults in some of the enormous number of sensors in order to achieve parallel production by numbering up. In a previous study, a simultaneous control system for two different temperatures was proposed in an experimental system that imitated the microreactor cooled by Peltier devices. In addition, a fault-tolerant control system for one area has also been proposed. However, the fault-tolerant control system could not be applied to the control system of two temperatures in the previous study. In this paper, we extend it to a two-input, two-output fault-tolerant control system. We also use a fault detection system that combines ChangeFinder, a time-series data analysis method, and One-Class SVM, an unsupervised learning method. Finally, the effectiveness of the proposed method is confirmed by experiments.

Keywords: microreactor; fault detection method; operator theory; nonlinear control; right coprime factorization; fault-tolerant control

1. Introduction

In recent years, microreactors, in which reactions are continuously carried out in tiny liquid channels, have attracted attention as a next-generation chemical reaction device based on flow methods. Because microreactors have a large surface area ratio where liquids come into contact with each other, they can facilitate chemical reactions with greater efficiency and faster reaction rates than conventional batch methods. In addition, microreactors contribute to the miniaturization of production facilities and are expected to be applied to high-mix, low-volume production processes. Furthermore, by numbering up multiple microreactors in parallel, large-scale production is possible [1,2]. Experiments have been conducted with the made microreactors, and the effectiveness of their hydrodynamic and chemical reaction properties has been confirmed [3,4]. Microreactors have attracted attention for their ability to achieve precise temperature control because of the advantages of their extremely narrow liquid flow paths and their ability to maintain uniform temperatures of the reaction liquid. The experimental equipment used in this study uses a Peltier device as the actuator that performs the cooling. Although nonlinear characteristics have been confirmed to exist in the heat absorption of Peltier devices [5,6], precise temperature control of the aluminum plate by electric current has been confirmed [7,8].

In recent years, industrial plants have become larger and larger, and more and more attention is being paid to technologies that enable safe and economical plant operation. The effectiveness of model-based studies of fault detection and isolation has been reported in the field of control engineering [9,10]. When microreactors are used for continuous flow production, the number of sensors and actuators is expected to increase due to numbering up. As the number of devices in a system grows, the likelihood of individual component failure also increases, but it is impractical to repair each time a fault occurs. To avoid such economic losses, it has become common for redundant sensors and actuators to be implemented in plants. Therefore, even if some of the systems fail, such redundant equipment can be used to realize the control system with redundant operation against faults. A control system with redundancy against fault is called a fault-tolerant control system. However, from the viewpoint of safety, the system that can quickly identify the equipment in which the fault has occurred is required [11,12].

In addition, the use of time-series data to detect faults in production plants has been attracting attention. As production plants become larger and include more sophisticated systems as a result of advances in science and technology, the labor and cost involved in monitoring them has become an issue. Therefore, the construction of a system that senses each condition and automatically analyzes the data will enable real-time fault detection without human intervention [13]. ChangeFinder is a data analysis method that has been attracting attention for its ability to capture changes in behavior (change-points) in a time series rather than the data values themselves, and to calculate a score for the likelihood of a change-point in the time-series data. If time-series data show stationarity during normal conditions, the change-points detected by ChangeFinder may be equipment anomalies [14].

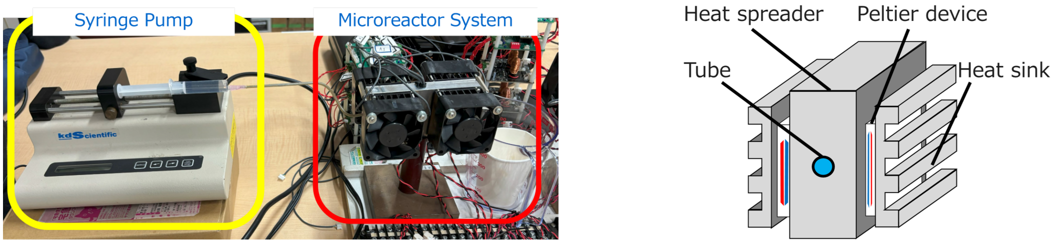
In a previous study, the fault-tolerant control system based on operator theory considering Bounded-Input Bounded-Output (BIBO) stability was proposed for minor temperature sensor faults in microreactors, aiming to continue control even in the event of the fault [15]. The fault-tolerant control system based on operator theory that takes into account BIBO stability has been proposed for the purpose of maintaining control even in the event of minor faults in temperature sensors of microreactors. Operator theory is a nonlinear control theory that expresses inputs and outputs in terms of maps (operators) [16]. Nonlinear control theory makes it possible to design control systems without linear approximation.

However, fault detection is required in general fault-tolerant control systems because the control loop is switched between faulty and normal conditions. Prior research has applied One-Class SVM, which is unsupervised learning, and proposed fault detection that does not require a signal at fault in advance. However, the challenge is to accurately determine whether the cause of fluctuations in the signal output from the sensor is a fault or a characteristic of the control target. Early fault detection is a trade-off with the incidence of false detection of faults due to noise and other factors, and from the standpoint of safety, even more accurate and early fault detection is a challenge [15].

Therefore, in this study, we will perform two tasks: detecting faults in temperature sensors and constructing a fault-tolerant control system using the microreactor system cooled by Peltier devices as the control target. For fault detection, the unsupervised fault detection system combining ChangeFinder and One-Class SVM was used. This is because we aimed to improve detection performance by providing explicit features from data analysis to the input of the One-Class SVM. A supervised classification system combining ChangeFinder and SVM was proposed in previous research [17]. Since the hyperparameters are not optimized in the previous method, the fault classification performance will be improved by optimizing the hyperparameters using a real-coded genetic algorithm. In addition, the fault detection system is effective for only one temperature sensor in the microreactor system in the conventional method. On the other hand, in this paper, a fault detection system is constructed for multiple sensors. Similarly, in designing the fault-tolerant control system, we will design a fault-tolerant control system that is effective for multiple sensors. The effectiveness of this system is confirmed by simulation and actual experiments, assuming that a sensor fault occurs during temperature control such that the sensor is damaged and shows a measured value that is different from the original temperature.

2. Experimental System

The experimental apparatus used in this study is shown in Figure 1a. A schematic diagram of the microreactor system is shown in Figure 1b. The microreactor system was made in Okayama University, Okayama, Japan. The Peltier devices used were TES1-12739-T100-SS-TF01-AIO from Thermonamic Electronics (Jiangxi) Corp, Jiangxi, China. The syringe pump was a kds101 made in the U.S. by KD Scientific Inc. (Holliston, MA, USA). The experimental apparatus consists of the microreactor system and a syringe pump for water flow, as shown in Figure 1a. The microreactor system consists of an aluminum heat spreader and a tube passing through the center, which is the water flow path. A Peltier device is installed to cool the heat spreader. The goal of this research is to control the temperature of the tube. A syringe pump is used to supply a constant flow of water to the tubes. The flow rate of room temperature water in this study is 0.5 [mL/min], which is the maximum flow rate.



(**a**) (**b**)

Figure 1. Experimental system: (**a**) external view. (**b**) Schematic diagram of the microreactor system.

3. Problem Statement

This chapter describes the problem setup for this study. In this study, we assume a case in which a tube temperature sensor fails during temperature control of the microreactor system. In this case, we aim to construct a system that allows temperature control to continue without shutting down the entire plant, even during the temperature sensor fault. The following two steps are required:

1. A system that determines in real time whether the temperature sensor of a tube is normal or abnormal (fault detection system).
2. A control system that compensates for tracking to the reference input in the event of a tube temperature sensor fault (fault-tolerant control systemm)

The fault detection system consists of a One-Class SVM with some of its inputs added to ChangeFinder's outputs. The discriminant model trained only on normal data learns the relationship between the control input, temperature sensor measurements, and the change-point scores of the temperature sensor measurements by ChangeFinder. If the observed data do not fit the learned relationship, the model discriminates abnormal conditions. This enables fault detection without requiring prior fault data and without specializing in a particular pattern of faults.

The fault-tolerant control system is a nonlinear control system that considers Bounded-Input Bounded-Output (BIBO) stability based on operator theory. The system assumes that an unknown fault factor is added to the normal measured value of the temperature sensor that has failed, and the system operates in such a way that the fault factor is removed. If the control system can remove the fault elements, the temperature of the controlled object can follow the reference input even in the event of an abnormality. However, fault-tolerant control systems require the fault detection system to be activated. Early fault detection is necessary from the standpoint of stability.

Assumed Temperature Sensor Fault Conditions

The following is an explanation of the temperature sensor fault situation assumed in this study. As it is difficult to intentionally cause a sensor fault, the fault is emulated using the measured values of a normal sensor. Various types of sensor faults have been reported [18]. In this study, we consider the case where the sensor deviates from its predetermined position due to a situation such as the sensor being unglued, and shows a measured value that is different from the original temperature of the controlled object. This situation is shown in Figure 2. The fault is represented as a step-like fault element d_w added to the original temperature y for the measured value y_f, as shown in Equations (1) and (2). Note that A is the amplitude of the fault signal, B is the time constant of the fault signal, and T_F is the fault time. The reason for this is explained below. When the position of the sensor suddenly changes as shown in Figure 2b, the ambient temperature of the sensor changes. The Fourier law and thermal radiation create a thermal gradient in the surrounding area. Therefore, a step-like offset change generally occurs while the sensor is moving. Step-like changes also occur due to the thermal inertia of the sensor itself. In temperature, where heat transfer and heat conduction are dominant compared to thermal radiation, the offset change during the fault is expected to be a first-order delay due to the thermal inertia of the sensor. Only fault patterns with such offsets will be considered. The original measured value y is not used in the experiment and y_f is used in all control systems.

$$y_f = y + d_w \tag{1}$$

$$\begin{cases} d_w = 0 & (T < T_F) \\ d_w = A(1 - \exp(-B(T - T_F))) & (T \geq T_F) \end{cases} \tag{2}$$

An example of the normal and faulty measured deviation at that time is shown in Figure 2.

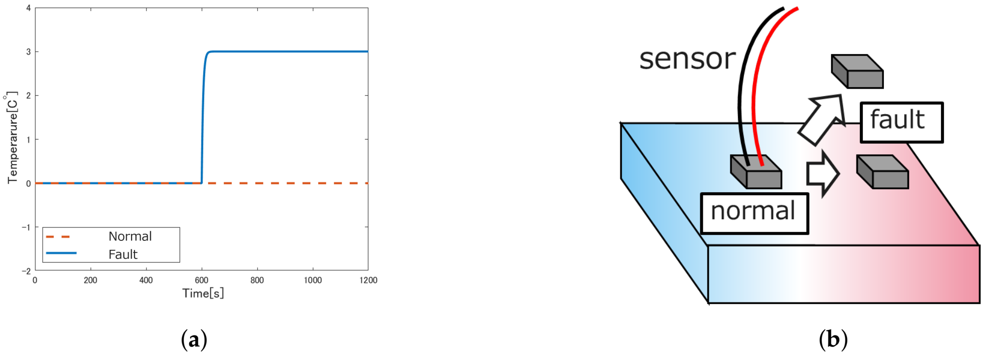


Figure 2. Assumed temperature sensor fault conditions: (**a**) deviation of normal and faulty sensor readings from the correct values. (**b**) Temperature sensor fault condition.

4. Modeling

This chapter describes the modeling of the entire microreactor system, including the microreactor(tube) and auxiliary cooling system [19]. Define the multiple area (Part) in the device, as shown in Figure 3. Three Parts are defined for each tube and heat spreader, and the sensor to measure the temperature is installed. The parameters used in the modeling of heat spreaders and tubes are listed in Table 1. The corresponding area parameters in Figure 3 are shown in Table 2. The parameters in Tables 1 and 2 refer to previous study [19], with some modifications.

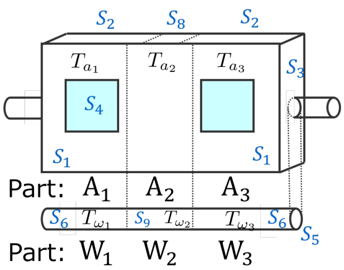


Figure 3. Area (Part) definition of microreactor system.

Table 1. Parameters of modeling.

Symbol	Description	Value/Unit
T_0	Initial temperature (outside temperature)	[K]
T_c	Heat-absorbing surface temperature of Peltier device	[K]
T_h	Heat dissipation surface temperature of Peltier device	[K]
T_{a_n}	Temperature of Part A_n	[K]
T_{w_k}	Temperature of Part W_k	[K]
i	Input current of Peltier device	[A]
S	Seebeck coefficient of Peltier device	0.053 V/K
K	Thermal conductivity of Peltier device	0.63 W/K
R	Resistance of Peltier device	2.0 Ω
ϵ_a	Emissivity of Aluminum	0.2
ϵ_w	Emissivity of water	0.93
σ	Stefan–Boltzmann constant	5.67×10^{-8} W/(m$^2 \cdot$ K^4)
α	Heat transfer coefficient of air	180 W/(m$^2 \cdot$ K)
α_w	Heat transfer coefficient of water	W/(m$^2 \cdot$ K)
λ_a	Thermal conductivity of aluminum	135 W/(m · K)
λ_w	Thermal conductivity of water	0.602 W/(m · K)
c_a	Specific heat of aluminum	900 J/(kg · K)
c_w	Specific heat of water	4180 J/(kg · K)
m_a	Mass of aluminum	[kg]
m_w	Mass of water	[kg]

Table 2. Parameters of area.

Symbol	Value	Unit	Symbol	Value	Unit
S_1	2.6×10^{-3}	m^2	S_2	7.0×10^{-4}	m^2
S_3	9.8×10^{-3}	m^2	S_4	9.0×10^{-4}	m^2
S_5	$9.0\pi \times 10^{-6}$	m^2	S_6	$3.0\pi \times 10^{-4}$	m^2
S_7	1.4×10^{-3}	m^2	S_8	2.8×10^{-4}	m^2
S_9	$1.2\pi \times 10^{-4}$	m^2			

4.1. Thermal Models of Heat Spreader

In this section, heat spreaders are modeled based on the laws of heat transfer. The three heat transfer laws used are Fourier's law, Newton's cooling law, and Stefan–Boltzmann's law. The relational equation for heat conduction in Part $A_n (n = 1, 2, 3)$ is obtained using the heat transfer laws and Tables 1 and 2 as follows. For the left side of the model equation, it is the time derivative of the thermal energy of the Part. The left-hand side shows the total amount of heat energy exchanged by external factors, indicating the heat energy transfer per unit time. Note that since the defined Parts are adjacent to each other, there is interference due to heat transfer. Note that $S' = 2S_1 + 2S_2 + S_3 - S_5$. In addition, u_{d_1} and u_{d_3} are the heat absorbed from the Peltier device and are discussed in Section 4.3.

$$\text{Part A}_1: \frac{d(T_0 - T_{a_1})m_a c_a}{dt} = 2u_{d_1} - \alpha S'(T_0 - T_{a_1}) + \alpha_\omega S_6 (T_{a_1} - T_{\omega_1}) + \frac{\lambda_a S_3 (T_{a_1} - T_{a2})}{dx} + \epsilon_a \sigma S' (T_{a_1}^4 - T_0^4) \tag{3}$$

$$\begin{aligned}\text{Part A}_2: \frac{d(T_0 - T_{a_2})m_a c_a}{dt} &= -\alpha(T_0 - T_{a_2})(2S_7 + 2S_8) + \alpha_\omega S_9 (T_{a_2} - T_{\omega_2}) - \frac{\lambda_a S_3 (T_{a_1} - T_{a_2})}{dx} \\ &\quad - \frac{\lambda_a S_3 (T_{a_3} - T_{a_2})}{dx} + \epsilon_a \sigma (T_{a_2}^4 - T_0^4)(2S_7 + 2S_8)\end{aligned} \tag{4}$$

$$\text{Part A}_3: \frac{d(T_0 - T_{a_3})m_a c_a}{dt} = 2u_{d_3} - \alpha S'(T_0 - T_{a_3}) + \alpha_\omega S_6 (T_{a_3} - T_{\omega_3}) + \frac{\lambda_a S_3 (T_{a_3} - T_{a_2})}{dx} + \epsilon_a \sigma S' (T_{a_3}^4 - T_0^4) \tag{5}$$

4.2. Thermal Model of Tube

Similarly, the thermal model in Part $\mathrm{W}_n (n = 1, 2, 3)$ is as follows.

$$\text{Part W}_1 : \frac{d(T_0 - T_{\omega_1}) m_\omega c_\omega}{dt} = -\frac{\lambda_\omega S_5((T_{\omega_0} - T_{\omega_1}) - (T_{\omega_0} - T_{\omega_2}))}{dx} + \alpha_\omega S_6 (T_{\omega_1} - T_{a_1}) \tag{6}$$

$$\text{Part W}_2 : \frac{d(T_0 - T_{\omega_2}) m_\omega c_\omega}{dt} = -\frac{\lambda_\omega S_5(2(T_{\omega_0} - T_{\omega_2}) - (T_{\omega_0} - T_{\omega_1}) - (T_{\omega_0} - T_{\omega_3}))}{dx} + \alpha_\omega S_9 (T_{\omega_2} - T_{a_2}) \tag{7}$$

$$\text{Part W}_3 : \frac{d(T_0 - T_{\omega_3}) m_\omega c_\omega}{dt} = -\frac{\lambda_\omega S_5((T_{\omega_0} - T_{\omega_3}) - (T_{\omega_0} - T_{\omega_2}))}{dx} + \alpha_\omega S_6 (T_{\omega_3} - T_{a_3}) \tag{8}$$

4.3. Heat Absorption by Peltier Device

In this study, the control target is cooled by heat absorption from the Peltier device. For the modeling of the heat spreader performed in Section 3, the control input is the heat absorption u_{d_n} from the Peltier device. In this section, the thermal model of the Peltier device is presented and the calculation method of the current input to the experimental device is described.

The heat absorption of the Peltier device is defined as positive. The model of the heat absorption u_d[W] is shown in Equation (9). Similarly, the model of heat dissipation u_h[W] when heat dissipation is defined as positive is shown in Equation (10).

$$u_\mathrm{d} = ST_\mathrm{c} i - K(T_\mathrm{h} - T_\mathrm{c}) - \frac{1}{2} R i^2 \tag{9}$$

$$u_\mathrm{h} = ST_\mathrm{h} i - K(T_\mathrm{h} - T_\mathrm{c}) + \frac{1}{2} R i^2 \tag{10}$$

The command value given to the experimental apparatus is the current i_n. When expressed as above, we consider determining the current flowing through the Peltier device based on the heat absorption u_{d_n} to be referred to. In this case, the current i_n is obtained by solving the Equation (9) for i_n as follows.

$$i = \frac{ST_\mathrm{c} \pm \sqrt{(ST_\mathrm{c})^2 - R(2K(T_\mathrm{h} - T_\mathrm{c}) + 2u_\mathrm{d})}}{R} \tag{11}$$

Thermal Model Deformation

Equivalent transformations are performed for the models shown in Section 3, and the transformation is made into an easy-to-handle form. Defining the temperature drop of the controlled object from the outside temperature $y_{a_n}(t) = T_0 - T_{a_n} (n = 1, 2, 3)$, the model for the heat spreader is as follows.

$$\text{Part A}_n : \frac{dy_{a_n}(t)}{dt} = \omega_{a_n}(t) + \sum_{m=1}^{4} (-1)^m A_{a_{nm}} y_{a_n}^m(t) \tag{12}$$

In this case, $\omega_{a_n}(t)$ (n = 1, 2, 3) is as follows. The A_{11} to A_{34} are constants obtained by Equation transformation.

$$\omega_{a_1}(t) = \frac{2u_{d_1}(t) + \alpha_w S_6 y_1(t) + \frac{\lambda_a S_3}{dx} y_{a_2}(t)}{m_a c_a} \tag{13}$$

$$\omega_{a_2}(t) = \frac{\alpha_w S_9 y_2(t) + \frac{\lambda_a S_3}{dx} y_{a_1}(t) + \frac{\lambda_a S_3}{dx} y_{a_3}(t)}{m_a c_a} \tag{14}$$

$$\omega_{a_3}(t) = \frac{2u_{d_1}(t) + \alpha_w S_6 y_3(t) + \frac{\lambda_a S_3}{dx} y_{a_2}(t)}{m_a c_a} \tag{15}$$

Similar to the Section 4.2, the equation transformation is performed for the model for the tube Part. Defining the temperature $y_k = T_0 - T_{w_k} (k = 1, 2, 3)$ of the drop from the outside temperature to be controlled, the model for the tube is as follows.

$$\text{Part W}_k : \frac{dy_k(t)}{dt} = \omega_{w_k}(t) - A_{w_k} y_k(t) \qquad (k = 1, 2, 3) \tag{16}$$

In this case, $\omega_{w_k} (k = 1, 2, 3)$ are expressed as follows. A_{w_1} to A_{w_3} are constants obtained by equation transformation.

$$\omega_{w_1}(t) = \frac{\alpha_w S_6 y_{a_1}(t)}{m_{w_1} c_w} \qquad \omega_{w_2}(t) = \frac{\alpha_w S_9 y_{a_2}(t)}{m_{w_2} c_w} \qquad \omega_{w_3}(t) = \frac{\alpha_w S_6 y_{a_3}(t)}{m_{w_3} c_w} \tag{17}$$

5. Control Design

In this chapter, we describe the design of a two-input, two-output temperature control system for the three-division model.

5.1. Right Factorization

Right factorization [16] of the model of the control target described above is performed based on operator theory. Define the heat input including interference in the heat spreader as z_{a_n} and the heat input of the tube as z_{w_k}. The derived model is subjected to right factorization based on operator theory. First, from Equations (12)–(16), ω_{a_n}, ω_{w_k} are shown.

$$\omega_{a_n}(t) = \frac{z_{a_n}(t)}{m_a c_a} \qquad \omega_{w_{k}(t)} = \frac{z_{w_k}(t)}{m_w c_w} \tag{18}$$

Defining z_{a_n} and z_{w_k} in Equation (18), Part A$_n$ and Part W$_k$ can be shown as in Equations (19) and (20).

$$\text{Part A}_n : \dot{y}_{a_n}(t) = \frac{z_{a_n}(t)}{m_a c_a} + \sum_{m=1}^{4} (-1)^m A_{a_{nm}} y_{a_n}^m(t) \tag{19}$$

$$\text{Part W}_k : \dot{y}_k(t) = \frac{z_{w_k}(t)}{m_w c_w} - A_{w_k} y_k(t) \tag{20}$$

Here, y_{A_n} is the temperature of the heat spreader Part A$_n$. Also, y_k is the temperature of the tube Part W$_k$. y_{A_n} and y_k can be observed from the installed temperature sensors.

Equations (21)–(23) are shown for the heat input z_{a_n} and z_{ω_n} including the interference of each plant.

$$z_{a_1}(t) = 2u_{d_1}(t) + \frac{\lambda_a S_3}{dx} y_{a_2}(t) + \alpha_w S_6 y_1(t) \qquad z_{w_1} = \alpha_w S_6 y_{a_1}(t) + \frac{\lambda_w S_5}{dx} y_2(t) \tag{21}$$

$$z_{a_2}(t) = \frac{\lambda_a S_3}{dx} y_{a_1}(t) + \frac{\lambda_a S_3}{dx} y_{a_2}(t) + \alpha_w S_6 y_2(t) \qquad z_{w_2}(t) = \alpha_w S_6 y_{a_2}(t) + \frac{\lambda_w S_5}{dx}(y_1(t) + y_2(t)) \tag{22}$$

$$z_{a_3}(t) = 2u_{d_3}(t) + \frac{\lambda_a S_3}{dx} y_{a_2}(t) + \alpha_w S_6 y_3(t) \qquad z_{w_3}(t) = \alpha_w S_6 y_{a_3}(t) + \frac{\lambda_w S_5}{dx} y_2(t) \tag{23}$$

Based on operator theory, the plant under control is right factorized into invertible operator D_n^{-1} and stable operator $\tilde{N}_{a_n}$. The heat spreader Part A_n is expressed as the following operator.

$$D_{a_n} : \{z_{a_n}(t) = m_a c_a \omega_{a_n}(t) \tag{24}$$

$$N_{a_n} : \begin{cases} \dot{x}_{a_n}(t) = \omega_{a_n}(t) + \sum_{m=1}^{4} (-1)^m A_{a_{nm}} y_{a_n}^m(t) \\ y_{a_n}(t) = x_{a_n}(t) \end{cases} \tag{25}$$

5.2. Control Design with Right-Coprime Factorization

In this section, the stability of the entire control system is guaranteed by designing a controller based on operator theory for the control target with right factorization [16]. In addition, a controller that compensates for tracking to the reference input in the outer loop is designed to guarantee tracking. The designed controller is shown in Figure 4. Figure 4 shows the control system in vector, for example $\mathrm{R}_{\boldsymbol{w}}^{\boldsymbol{-1}} = (R_{w_1}^{-1}, R_{w_2}^{-1}, R_{w_3}^{-1})$. Here, the plants to be controlled are $D_{a_n}^{-1}$ and N_{a_n} representing heat spreaders and P_{w_n} representing tubes. We further define a new operator $\tilde{N}_{w_n} = P_{w_n} N_{a_n}$.

In designing the control system, the controller design is performed under the conditions of $n = (1.3)$, $k = (1, 3)$, and $n = k$. This is because there are only two regions where the amount of heat absorption can be input from the Peltier device: Part $\mathrm{A_1}$ and Part $\mathrm{A_3}$. Therefore, the system cools and controls Part $\mathrm{A_n}$ by heat transfer from Part $\mathrm{W_n}$. All other heat transfers that occur between the other Part are interferences. In this section, the design is performed for the nominal model.

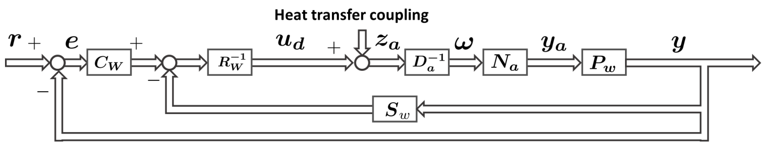


Figure 4. Control system based on operator theory.

Design the operators according to the Bezout identity in Equation (A12). Designing the operators S_n and R_n so that they satisfy the Beozut identity compensates for the BIBO stability of the nonlinear system. Then S_n and R_n become right-coprime factorizations. In Equation (A12), I is the identity operator and outputs the input signal as it is.

$$S_{w_n} \tilde{N}_{w_n} + R_{w_n} D_{a_n} = I \tag{26}$$

The designed operators are shown in Equations (27) and (28). However, a constant $B_k (0 < B_k < 1)$ is used as the design parameter.

$$S_{w_n} = (1 - B_k) \tilde{N}_{w_n}^{-1} \tag{27}$$

$$R_{w_n} = \frac{B_k}{m_a c_a} \tag{28}$$

From the above, the BIBO stability of the inner loop is guaranteed. However, the design of Equation (A12) cannot perform reference input tracking. Therefore, the reference input tracking is achieved by designing a tracking compensation controller as in Equation (29). Here, K_{P_n} and K_{I_n} are the design gains.

$$C_{w_n}(e_n)(t) = K_{P_n} e_n(t) + K_{I_n} \int_0^t e_n(\tau) d\tau \tag{29}$$

5.3. Fault-Tolerant Control System

In this section, we design a control system that can follow a reference input even in the event of a tube-mounted temperature sensor fault by activating the sensor in the event of the fault. The fault-tolerant control system in this study consists of the fault detection system and the fault signal compensation control system. The control system described is the fault-tolerant control system that is extended from the one-input, one-output fault-tolerant control system of the two-partition model of the previous study [15] to a three-partition model. The fault signal compensation control system used in this study is shown in Figure 5. As in the Section 5.2, only $n = k = (1, 3)$ is explained. Here, FD in Figure 5 denotes the fault detection system, which switches to the fault-tolerant control system. Operator Q_n is an operator that outputs a signal to compensate for faults and is enabled by the fault detection system.

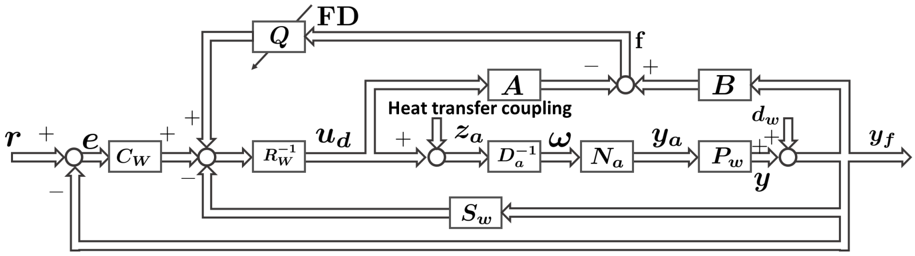


Figure 5. Fault-tolerant control system.

In the case of a sensor fault, the fault signal f_n is shown as in Equation (30) using the fault element d_{ω_n}.

$$f_n(t) = B_n(y_n + d_{\omega_n})(t) - A_n(u_{d_n})(t) \tag{30}$$

Here, when Q_n is enabled, the output of the tracking compensation controller C_{w_k} is organized as in Equation (31).

$$\begin{aligned}&(S_{w_k}\tilde{N}_{w_k} - QB_n\tilde{N}_{w_k})\tilde{N}_{w_k}^{-1}(y_n)(t)\\ &=C_{w_k}(e_k)(t) - (R_{w_k}D_{a_n} + QA_nD_{a_n})\tilde{N}_{w_k}^{-1}(y_n + d_{w_n})(t)\end{aligned} \tag{31}$$

In this case, design A_n and B_n so that the left side is $\tilde{N}_{w_k}$ and the term including the fault element on the right side is O operator, respectively. Since the model equation for A_n is the same as for Part A_n, the variable $A_n(u_{d_n})$ uses the temperature sensor measurements of Part A_n.

$$A_n(u_{d_n})(t) = N_{a_n}D_{a_n}^{-1}(u_{d_n})(t) \tag{32}$$

$$B_n(y_n + d_{w_n})(t) = P_{w_k}^{-1}(y_n + d_{w_n})(t) \tag{33}$$

When operators A_n, B_n satisfying the conditions are designed as in Equations (32) and (33), operator Q_n is shown as in Equation (34).

$$Q_n(f_n)(t) = -\tilde{N}_{\omega_k}^{-1}R_{w_k}D_{a_n}P_{w_k}(f_n)(t) \tag{34}$$

Substituting Equations (32)–(34) into Equation (31) again, a relational expression such as Equation (35) is shown.

$$(S_{w_k}\tilde{N}_{w_k} + R_{w_k}D_{a_n})\tilde{N}_{w_k}^{-1}(y_n)(t) = C_{w_k}(e_k)(t) \tag{35}$$

The design of the operator as in Equation (A12) also shows the relationship as in Equation (36).

$$\tilde{N}_{w_k}^{-1}(y_n)(t) = C_{w_k}(e_k)(t) \tag{36}$$

The relationship in Equation (36) transforms the operator Q_n into an operator with a tracking compensation operator Cw_k. However, to account for thermal disturbances caused by modeling errors of the control target and thermal disturbances caused by thermal transfer with the liquid flowing in the channel, it is transformed like an operator with Mw_k whose gain of Cw_k is adjusted separately.

$$Q_n(f_n)(t) = -M_{w_k}R_{w_k}D_{a_n}P_{w_k}(f_n)(t) \tag{37}$$

The relationship in Equation (36) removes the fault element from the output, allowing it to follow the reference input even in the event of a fault. Note that $Q_n(f_n)(t) = 0$ when the sensor is determined to be normal.

5.4. Consideration of Coupling Effects

The design of the Sections 5.2 and 5.3 assumes that the two-input two-output control system is $n = k = (1, 3)$ interference-free independent two subsystems. In reality, however, there is interference due to heat transfer between adjacent parts. We consider removing the effect of interference by compensating for this interference with the input from the actuator. The control system considering the removal of couplings is shown in Figure 6.

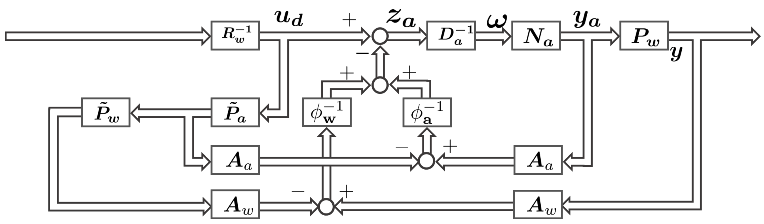


Figure 6. Control system to remove coupling effects.

Here, $\tilde{\boldsymbol{P}}_{\boldsymbol{a}} = (\tilde{P}_{a_1}, \tilde{P}_{a_2}, \dots \tilde{P}_{a_n})$ is the heat transfer model of the heat spreader, $\tilde{\boldsymbol{P}}_{\boldsymbol{w}} = (\tilde{P}_{w_1}, \tilde{P}_{w_2}, \dots \tilde{P}_{w_k})$ is the tube heat transfer model. When there is no effect of coupling, $z_i = u_{d_i}$ is valid, and the control system design is performed with this as a target. The operators designed in Figure 6 are shown in Equations (38) and (39) below. Here, each operator is assumed to have a bounded signal input and is defined as α_{a_i} and α_{w_i}. Here, n_a and n_w are design parameters and $\boldsymbol{\phi_a} = (\phi_{a_1}, \phi_{a_2} \dots, \phi_{a_n})$, $\boldsymbol{\phi_w} = (\phi_{w_1}, \phi_{w_2} \dots, \phi_{w_k})$ is a linear operator such that $\phi_i(\alpha_i(t)) \to 0$ for any bounded input signal. I is the identity operator.

$$\phi_{a_i}^{-1}(\alpha_{a_i})(t) = \frac{1}{n_a}\alpha_{a_i} \qquad \phi_{w_i}^{-1}(\alpha_{w_i})(t) = \frac{1}{n_w}\alpha_{w_i} \tag{38}$$

$$A_a = I \qquad A_w = I \tag{39}$$

The equation for z_i is rearranged as in Equation (40).

$$\begin{aligned} z_i(t) =& u_{d_i}(t) + d_i(t) + \phi_{a_i}^{-1} A_{a_i} \tilde{P}_{a_i}(u_{d_i})(t) \\ &+ \phi_{w_i}^{-1} A_{w_i} \tilde{P}_{w_i} \tilde{P}_{a_i}(u_{d_i})(t) - \phi_{a_i}^{-1} A_{a_i} P_{a_i}(z_i)(t) - \phi_{w_i}^{-1} A_{w_i} P_{w_i} P_{a_i}(z_i)(t) \end{aligned} \tag{40}$$

Furthermore, a transformation of Equation (40) leads to Equation (41).

$$\begin{aligned} &\phi_{a_i}^{-1}(\phi_{a_i} + A_{a_i} P_{a_i})(z_i)(t) + \phi_{w_i}^{-1}(\phi_{w_i} + A_{w_i} P_{w_i} P_{a_i})(z_i)(t) \\ &= d_i(t) + \phi_{a_i}^{-1}(\phi_{a_i} + A_{a_i} \tilde{P}_{a_i})(u_{d_i})(t) + \phi_{w_i}^{-1}(\phi_{w_i} + A_{w_i} \tilde{P}_{w_i} \tilde{P}_{a_i})(u_{d_i})(t) \end{aligned} \tag{41}$$

Since $\phi_{a_i}(\alpha_{a_i}(t)) \to 0$ and $\phi_{w_i}(\alpha_{w_i}(t)) \to 0$, the Equation (42) is achieved. This eliminates the effect of coupling.

$$z_i(t) \to u_{d_i}(t) \tag{42}$$

However, if the tube's temperature sensor is determined to have failed, the output of the nominal heat transfer model is used instead of the measured value from the temperature sensor for y_i.

5.5. Overall Control System

To summarize the above, the block diagram of the entire control system is shown in Figure 7. Here, in Figure 7, G_a is the operator indicating $G_a = \frac{\lambda_a S_3}{dx}$ and G_{aw} is the operator indicating $G_{aw} = \frac{\lambda_a S_3}{dx}$ and G_w is the operator indicating $G_w = \alpha_\omega S_6$.

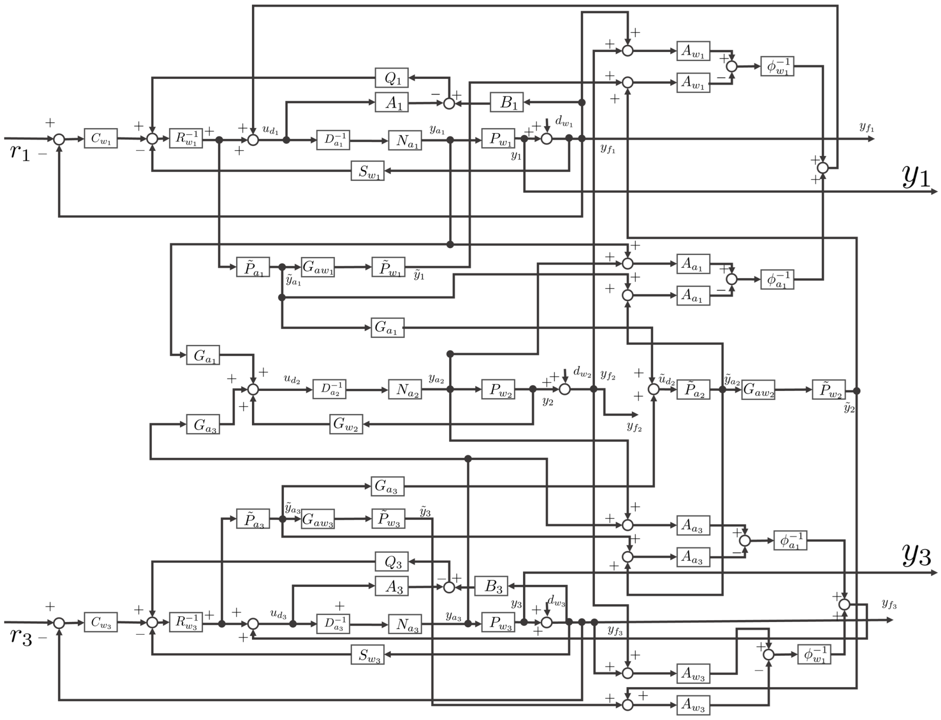


Figure 7. Overall control system.

6. Fault Detection System

This chapter presents the fault detection system for detecting tube temperature sensor faults. Here, the fault is represented by the control system designed in Section 5.2, shown in Figure 8. The purpose of fault detection in this study is to switch the control loop to the fault-signal-compensating control system described in Section 5.3 in the event of the fault. The goal is to detect the temperature sensor fault in the tube represented by y_f in Figure 8. The using fault detection system is a combination of ChangeFinder and One-Class SVM. One-Class SVM is described in Appendix A and ChangeFinder is described in Appendix B. If the change-points detected by ChangeFinder are considered as anomalies, the system has the advantage of reliably detecting change-points even for minor faults. On the other hand, if the reference input changes during control, it is also detected as a change-point. This has the disadvantage that an abnormality cannot be detected during the transient period until it is tracked again. Abnormality detection by One-Class SVM alone has the advantage that detection is effective even for unknown anomalies. It has the disadvantage that it is difficult to detect small faults; for example, when the distance between the normal data during training and the observed anomaly data is small. Therefore, we aim to construct a system that complements each other's weak points by adding the output of ChangeFinder to the input of One-Class SVM. The following two processes are required for anomaly detection.

1. Anomaly classification modeling with One-Class SVM. An anomaly discrimination model is constructed by preparing training data and having One-Class SVM learn them. The explanation is given in Sections 6.1 and 6.2.
2. Anomaly detection by anomaly discrimination model. Anomaly detection of acquired real-time data is performed using a discriminant model created in advance. The explanation is given in Section 6.3.

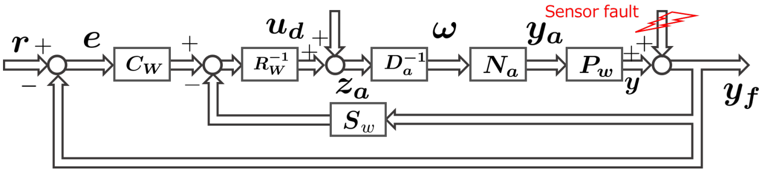


Figure 8. Representation of temperature sensor faults in control systems.

6.1. Creation of Classification Models by Learning

The fault detection system uses a combination of One-Class SVM and ChangeFinder to perform fault detection. In this section, we explain how to construct a classification model for One-Class SVM, which classifies observed data as normal or faulty. One-Class SVM can classify normal classes and abnormal classes by using an abnormality classification model learned using only previously acquired normal data. This makes it effective for plants for which faulty data are difficult to obtain or for which the faulty pattern is unknown. Therefore, training data that have the same dimensions as the data to be input during classification are required during training.

The discriminant model is constructed by training the One-Class SVM with previously observed training data. An overview of the process is shown in Figure 9.

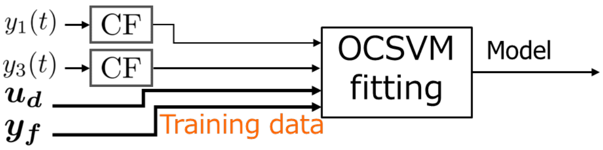


Figure 9. Training data learning.

As shown in Figure 9, the training data include the input $\boldsymbol{u_d}$ of the heat absorption of the Peltier device, the measured temperature of the tube $\boldsymbol{y_f}$, and the change-point score of $\boldsymbol{y_f}$ calculated by ChangeFinder. Since the thermal model of the microreactor is a dynamic system, past values affect the current values. In a dynamic system, if a classification model is built using only the current information, the plant characteristics will not be learned correctly and there is a risk of false positives. Therefore, by including time-shifted past data in the learning vector, it is possible to construct the classification model that takes the dynamic system into account. The variable to be trained is as in Equation (43). Note that $\mathrm{CF}(\boldsymbol{x})$ denotes the change-point score of time-series data $\boldsymbol{x}$ by ChangeFinder.

$$X = [\boldsymbol{u_{d_1}}(t), \boldsymbol{u_{d_3}}(t), \boldsymbol{y_1}(t), \boldsymbol{y_2}(t), \ldots, \boldsymbol{y_3}(t-l), \mathrm{CF}(\boldsymbol{y_1}(t)), \mathrm{CF}(\boldsymbol{y_3}(t))] \in \mathbb{R}^{N\times(1+(l+1))} \tag{43}$$

$$\boldsymbol{u_{d_1}}(t) = [u_{d_1}(1), u_{d_1}(2), \ldots, u_{d_1}(N)]^{\mathrm{T}} \in \mathbb{R}^N \tag{44}$$

$$\boldsymbol{u_{d_3}}(t) = [u_{d_3}(1), u_{d_3}(2), \ldots, u_{d_3}(N)]^{\mathrm{T}} \in \mathbb{R}^N \tag{45}$$

$$\boldsymbol{y_1}(t) = [y_1(1), y_1(2), \ldots, y_1(N)]^{\mathrm{T}} \in \mathbb{R}^N \tag{46}$$

$$\boldsymbol{y_3}(t) = [y_3(1), y_3(2), \ldots, y_3(N)]^{\mathrm{T}} \in \mathbb{R}^N \tag{47}$$

Here, N is the number of training data and $l = 2$ is the number of time shifts. In other words, four types of training data are used: input, output, output before one step, and output before two steps. The proposed method detects temperature sensors in Part W_1 and Part W_3, whereas the conventional method detects a single temperature sensor located at the center of the tube. In the thermal model of the microreactor, there is interference between adjacent parts, and this is the reason for this detection method.

6.2. Hyperparameter Optimization of One-Class SVM to Improve Classification Performance

One-Class SVM have parameters that must be specified as design parameters during training, called hyperparameters. The two hyperparameters in One-Class SVM are the kernel parameter γ and the normalization parameter ν. Hyperparameters have a strong influence on the classification performance of the constructed model, and thus need to be optimized in order to construct a more accurate classification model.

This parameter optimization problem can be treated as an objective function maximization problem with the performance of the constructed classification model as the objective function. In this study, we use a real-coded genetic algorithm [20,21] as one of the optimization methods on a continuous search space. The real-coded genetic algorithm is an optimization algorithm inspired by the evolutionary process of living organisms, in which a real-coded vector is an individual and the value of the objective function is the strength of the individual. Real-coded genetic algorithms are characterized by the fact that they do not explicitly use the gradient of the objective function. Therefore, optimization is possible even for functions for which the gradient of the objective function cannot be obtained, such as the classification performance evaluation in this study. Figure 10 shows the process of hyperparameter optimization using the real-coded genetic algorithm.

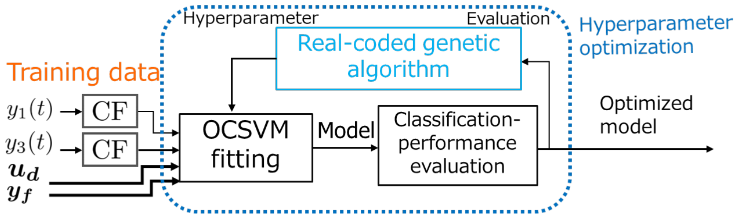


Figure 10. Hyperparameter optimization process using the real-coded genetic algorithm.

The process of Figure 10 is described in detail as follows.

1. Obtain input/output data in advance by experiment as training data.
2. Add the expected fault elements to some of the previously acquired input/output data to create test data in the event of the fault.
3. Obtain the hyperparameters to be tested using the real-coded genetic algorithm.
4. Construct a tentative classification model by training using the training data and the hyperparameters obtained.
5. Calculate the classification performance evaluation value from the classification model and test data.
6. The real-coded genetic algorithm determines the hyperparameters to try next.
7. Repeat the process from 3 to 6 for the product of the number of individuals and the number of generations to obtain the optimized hyperparameters.

Let z be the hyperparameter to be tried. The maximization problem for improving the classification performance of this study is shown in Equation (48). p_{pre} is the precision. It is the pattern ratio in which the correct answer is anomalous when it is determined to be anomalous. Similarly, P_{rec} is the recall. It is the proportion of patterns for which the discriminant model was able to discriminate an abnormality out of the patterns for which the correct answer was abnormality. Therefore, the penalty for false positives increases as the value of r is increased.

$$\begin{aligned} &\max_{z} -\frac{1}{rp_{pre} + (1-r)p_{rec}} \\ &\text{s.t. } 0 < r < 1 \end{aligned} \tag{48}$$

The above process improves the performance of the classification model by optimizing the hyperparameters of the One-Class SVM.

6.3. Fault and Normal Classification with One-Class SVM and ChangeFinder

In this section, we show how to perform anomaly detection using the discriminant model constructed in Section 6.1. Fault detection is achieved in real time by performing fault detection for each control step. The process for each control step is shown below.

1. The temperature is measured by sensors.
2. Control inputs and change-point scores are calculated from temperature measurements.
3. Similar to Equation (43), a variable X is created for anomaly determination, to be input to OCSVM.
4. If classified as faulty in 3, the fault tolerance control system is activated.

7. Simulation Results

In this chapter, the effectiveness of the proposed control and fault detection systems is verified by simulation. Simulations were performed using MATLAB 2023b. One-Class SVM used ocsvm function. ChangeFinder and real-coded genetic algorithms were self-developed. In the simulation, a normally distributed noise with mean 0 and variance 0.0005 is added to y_n assuming the measurement noise of the temperature sensor. The target temperature is (outside temperature - reference input). Simulations were performed for the following two cases. The parameters are also shown in Table 3.

1. Part W_1 increased 3.2 °C from 500 s and Part W_3 increased 3.6 °C from 500 s.
2. Part W_3 increased 0.2 °C from 500 s.

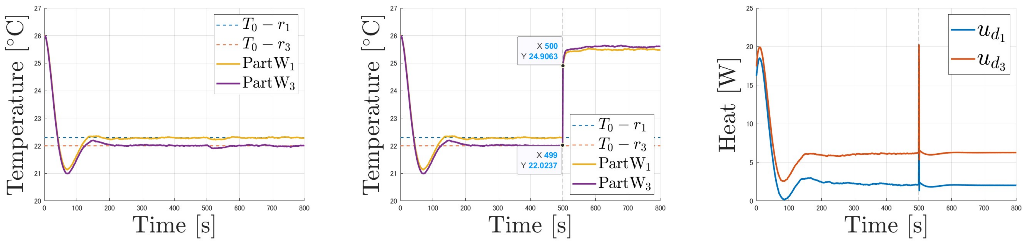
Table 3. Condition parameters of simulations.

Symbol	Description	Value	Unit
T_n	smoothing scale	10	–
T_0	Outside temperature (initial temperature)	26	°C
dt	Control cycle	1	s
T_F	Fault occurrence time	500	s
B	Time constant of a fault element	0.2	s
A_1	Fault element amplitude of PartW_1	–	°C
A_3	Fault element amplitude of PartW_3	–	°C
r	Oblivion coefficient	0.03	–
r_1	Reference input of PartW_1	3.7	°C
r_2	Reference input of PartW_3	4	°C
k	SDAR model order	1	–
B_n	Design parameter	0.9999	–
n_{w_n}	Design parameter	50	–
k_{C_p}	Proportional gain of C_{w_n}	0.0140	–
k_{C_i}	Integral gain of C_{w_n}	0.0003	–
k_{M_p}	Proportional gain of M_{w_n}	0.0140	–
k_{M_i}	Integral gain of A M_{w_n}	0.0003	–
n_{a_n}	Design parameter	50	–

7.1. Part W_1 Increased 3.2 °C from 500 s and Part W_3 Increased 3.6 °C from 500 s

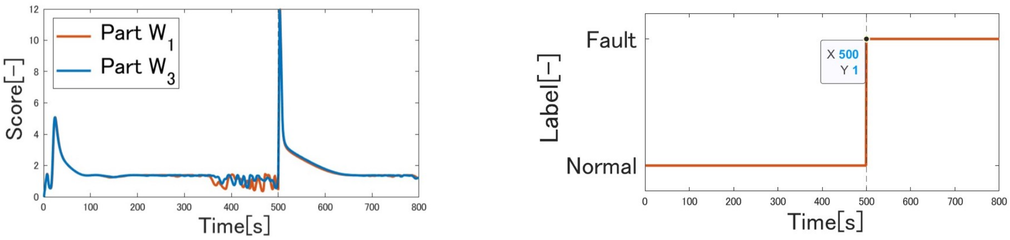
Figure 11 shows the results of temperature control using the proposed control system. Figure 12 also shows the results of the real-time fault classification. Figure 11b shows that the value of the fault sensor fluctuates due to the faults that occurred during the 500 s. However, as Figure 11a shows, the temperature of the controlled object continues to track the reference input. The control input jumps up at 500 s due to the sensor fault. However, it returns to the appropriate input by switching to the fault-tolerant control system.

Discussing the results for Figure 12a, ChangeFinder's change-point score jumped at 503 s, 3 s after the fault. Figure 12b shows that One-Class SVM detected the fault at 500 s, the control cycle in which the fault data were first measured; One-Class SVM detected the fault before ChangeFinder's score jumped. Thus, ChangeFinder contributed little to fault detection. In addition, there were no false positives.



(**a**) (**b**) (**c**)

Figure 11. Simulation results under condition where Part W_1 increased 3.2 °C from 500 s and Part W_3 increased 3.6 °C from 500 s: (**a**) temperature. (**b**) Measurements of faulty sensors. (**c**) Control input.

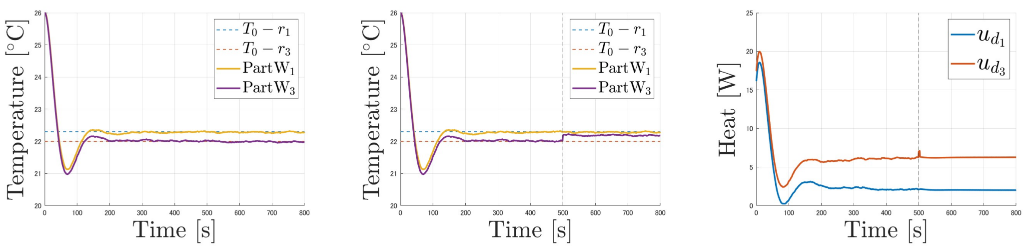


(**a**) (**b**)

Figure 12. Simulation results under condition where Part W_1 increased 3.2 °C from 500 s and Part W_3 increased 3.6 °C from 500 s: (**a**) change-point scores. (**b**) Classification of fault detection.

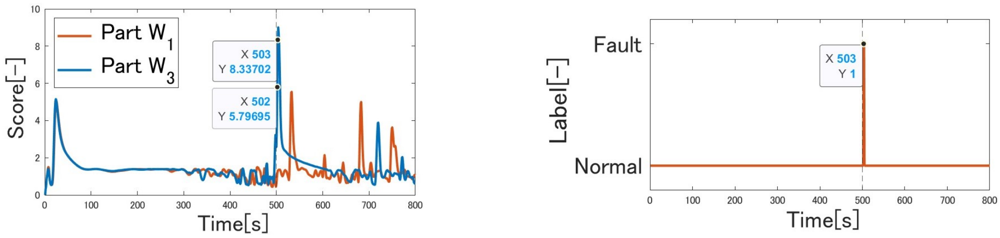
7.2. Part W_3 *Increased 3 °C from 500 s*

Similarly, Figure 13 shows the results of temperature control and Figure 14 shows the results of real-time fault detection. In Figure 13a, the temperature of the controlled object continues to track the reference input. In Figure 13b, the temperature change due to the fault was small. However, ChangeFinder detected the change-point and increased its score at 503 s. One-Class SVM also detected a fault at 503 s, 3 s after the fault occurred, coinciding in time with the increase in ChangeFinder's change-point score. This confirms the effectiveness of the fault detection system that combines the One-Class SVM input with the ChangeFinder output. No detection was made until the change-point score increased. Thus, it indicates that One-Class SVM alone could not detect faults of this small amplitude. However, as the ChangeFinder change-point score decreased, the fault detection system classified the fault condition as normal.



(**a**) (**b**) (**c**)

Figure 13. Simulation results under condition where Part W_3 increased 0.2 °C from 500 s: (**a**) temperature. (**b**) Measurements of faulty sensors. (**c**) Control input.



(**a**) (**b**)

Figure 14. Simulation results under condition where Part W_3 increased 0.2 °C from 500 s: (**a**) change-point scores. (**b**) Classification of fault detection.

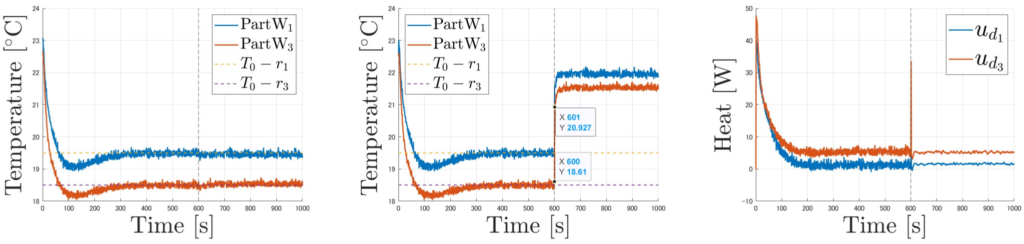
8. Experimental Results

In this chapter, we present the results of an experiment on an actual machine. In the experiment, the outside temperature was 22.591 °C. The measured temperature of Part A_2 immediately before the start of the experiment is defined as the constant outside temperature. The goal was to cool Part W_1 by 3 °C and Part W_3 by 4 °C from the outside temperature. The fault element was added to the temperature sensor values, assuming that the fault occurred in 601 s. Table 4 shows the differences in the parameters during the experiment compared to the simulation run. Figure 15 shows the results of the temperature control with the proposed control system. Figure 16 shows the results of the fault classification performed in real time.

As in the experiment, it was confirmed that the system could follow the reference input by switching to the fault-tolerant control system. In addition, fault detection during the 601 s was possible for faults that occurred in the 601 s.

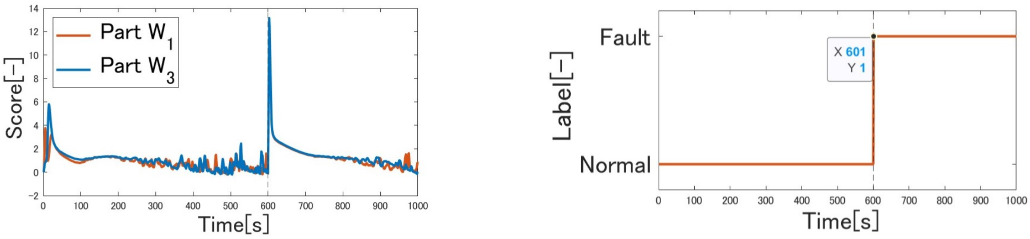
Table 4. Parameters of experiment.

Symbol	Description	Value	Unit
r	Oblivion coefficient	0.03	–
T_0	Outside temperature (initial temperature)	22.591	°C
k	SDAR model order	1	–
dt	Control cycle	1	s
T_F	Fault occurrence time	601	s
B	Time constant of a fault element	0.2	s
A_1	Fault element amplitude of PartW_1	3	°C
A_3	Fault element amplitude of PartW_3	3.5	°C
r_1	Reference input of PartW_1	3.0	°C
r_2	Reference input of PartW_3	4.0	°C
B_n	Design parameter	0.9999	–
n_{a_n}	Design parameter	50	–
n_{w_n}	Design parameter	150	–
k_{C_p}	Proportional gain of C_{w_n}	0.0185	–
k_{C_i}	Integral gain of C_{w_n}	0.00025	–
$k_{M_{p_1}}$	Proportional gain of M_{w_1}	0.0185	–
$k_{M_{p_3}}$	Proportional gain of M_{w_3}	0.0185	–
$k_{M_{i_1}}$	Integral gain of M_{w_1}	0.000249	–
$k_{M_{i_3}}$	Integral gain of M_{w_3}	0.000241	–



(**a**) (**b**) (**c**)

Figure 15. Experimental results under condition where Part W_1 increased 3.0 °C from 601 s and Part W_3 increased 3.5 °C from 601 s: (**a**) temperature. (**b**) Measurements of faulty sensors. (**c**) Control input.



(**a**) (**b**)

Figure 16. Experimental results under condition where Part W_1 increased 3.0 °C from 601 s and Part W_3 increased 3.5 °C from 601 s: (**a**) change-point scores. (**b**) Classification of fault detection.

Comparison of Simulation and Experiment

The simulation results shown in Section 7.1 are compared with the experiment results. However, since the target temperatures and fault requirements are different, the comparison is made based on the average absolute error from the reference input after the time of the fault. Specifically, the simulation is calculated using temperatures from 500 to 800 s, while the experiment is calculated using results from 601 to 1000 s. As shown in Table 5, the mean absolute error is smaller for the simulation. Therefore, the simulation results are superior in this comparison.

Table 5. Comparison of simulation and experiment.

Description	Value/Unit	Description	Value/Unit
Mean absolute error of Part W_1 (Simulation)	0.0181 °C	Mean absolute error of Part W_3 (Simulation)	0.0173 °C
Mean absolute error of Part W_1 (Experiment)	0.0735 °C	Mean absolute error of Part W_3 (Experiment)	0.0700 °C

9. Conclusions

In this study, the fault-tolerant control system was designed for the MIMO system that controls the temperature of two regions of the microreactor system. This extends the fault-tolerant control system, which was effective for only one region in previous studies. In addition, the fault detection system combining One-Class SVM and ChangeFinder was applied to detect faults; for training the One-Class SVM, the evaluation function for the classification model was created, and the hyperparameters were optimized by the real-coded genetic algorithm. Finally, the effectiveness of the system was verified through simulations and experiments. The fault detection system combining One-Class SVM and ChangeFinder detected faults even when the amplitude of the fault elements was small. The results show that the system is more effective than One-Class SVM alone in detecting failures in microreactor systems.

Author Contributions: Y.M. used a combination of One-Class SVM and ChangeFinder for training to create the fault classification model for temperature sensors. Parameter optimization of the fault classification model was performed using a real-coded genetic algorithm. In addition, we proposed a method to extend the fault-tolerant temperature control system of the microreactor to a MIMO fault-tolerant temperature control system, and proposed a control system combined with fault classification. Simulation and actual experiments were also conducted and this paper was written. M.D. suggested technical support and gave overall guidance on the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. One-Class SVM

One-Class SVM, a one-class classification method [22,23], is described. The algorithm is described below. First, the training data $\boldsymbol{x}_{i\in[l]}$ are prepared, and the inner product of the projection data by $\boldsymbol{\phi}$ is defined as a kernel function as in Equation (A1). In this study, the RBF kernel is used.

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\phi}(\boldsymbol{x}_i)^T \boldsymbol{\phi}(\boldsymbol{x}_j) \tag{A1}$$

Assuming that the normal and abnormal data are linearly separable in the high-dimensional space after mapping, the separation plane is expressed as Equation (A2) using the weight coefficient $\boldsymbol{\omega}$ and bias ρ.

$$\boldsymbol{\omega}^T \boldsymbol{\phi}(\boldsymbol{x}) - \rho = 0 \tag{A2}$$

To allow for misclassification, a slack variable $\xi_i \geq 0$ is introduced. In this case, we consider maximizing the distance from the origin with respect to the classification plane. Converting the maximization problem into a minimization problem, it is shown as follows, including the constraints using the normalization parameter $\nu \in (0, 1)$.

$$\begin{aligned} &\min_{\omega, \xi_i, \rho} \frac{1}{2} \|\boldsymbol{\omega}\|^2 - \rho + \frac{1}{l\nu} \sum_{i\in[l]} \xi_i \\ &\text{s.t. } (\boldsymbol{\omega}^T \cdot \boldsymbol{\phi}(\boldsymbol{x_i})) \geq (\rho - \xi_i), \quad \xi_i \geq 0, \forall i \in [l] \end{aligned} \tag{A3}$$

Also, introducing the new variables $\alpha_i, \beta_i \geq 0, \quad i \in [l]$ and Equation for the minimization problem, the Lagrangian function is shown as follows.

$$\begin{aligned} L(\boldsymbol{\omega}, \boldsymbol{\xi}, \rho, \boldsymbol{\alpha}, \boldsymbol{\beta}) =& \frac{1}{2} \|\boldsymbol{\omega}\|^2 - \rho + \frac{1}{l\nu} \sum_{i\in[l]} \xi_i \\ & - \sum_{i\in[l]} \alpha_i ((\boldsymbol{\omega}^T \cdot \boldsymbol{\phi}(\boldsymbol{x_i})) - \rho + \xi_i) - \sum_{i\in[l]} \beta_i \xi_i \end{aligned} \tag{A4}$$

The minimization problem becomes a $\boldsymbol{\alpha}$-only problem, as follows.

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j k\big(\boldsymbol{x}_i, \boldsymbol{x}_j\big) \\ &\text{s.t. } 0 < \alpha_i \geq \frac{1}{n\nu}, \ \forall i \in [n], \sum_{i\in[l]} \alpha_i = 1 \end{aligned} \tag{A5}$$

Using the final optimized Lagrange multiplier α_i, the decision function for the unknown data $\boldsymbol{x}$ are as follows.

$$f(\boldsymbol{x}) = \text{sgn}\left(\sum_{i \in [l]} \alpha_i k(\boldsymbol{x_i}, \boldsymbol{x}) - \rho \right) \tag{A6}$$

The outlier-likeness of unknown input patterns mapped to a higher-dimensional space increases as the distance from the origin is shortened. Therefore, input patterns that are placed closer to the origin than the classification plane do not belong to a class and are judged to be outliers.

Appendix B. ChangeFinder

ChangeFinder is an algorithm that computes the likelihood of change-points in time-series data. A change-point is a time when the behavior of time-series data changes significantly. In this study, we use the SDAR model, which has a forgetting function and can deal with non-stationary data to some extent. The SDAR model is effective in detecting essential change-points because it suppresses the increase in change-point scores caused by noise through two-stage learning [14]. The calculation procedure for each time-series data is as follows.

1. Reading time-series data x_t.
2. Learning SDAR model and computing probability density function p_t.
3. Calculation of outlier scores during one-step learning

$$\text{Score}_1(x_t) = -\log_{p_{t-1}}(x_t) \tag{A7}$$

4. Smoothing of scores and derivation of moving average y_t using smoothing width T_1.

$$y_t = \frac{1}{T_1} \sum_{i=t-T_1+1}^{t} \text{Score}_1(x_i) \tag{A8}$$

5. Apply SDAR to y_t and calculate the probability density function q_t for two-step learning.
6. Smoothing again yields the moving average at time t as the change-point score.

$$\text{Score}(t) = \frac{1}{T_2} \sum_{i=t-T_2+1}^{t} (-\log_{q_{i-1}}(y_i)) \tag{A9}$$

The higher the score, the greater the degree to which it is a change point. The following is a description of the SDAR algorithm.

1. Read the time-series data x_t.
2. The computed estimator μ is used to update the estimate of the statistic to be used. Where $r \in (0,1)$ is the forgetting coefficient. The higher the forgetting coefficient, the greater the influence of the past state.

$$\hat{\mu}_t = (1-r)\hat{\mu}_{t-1} + r\hat{\mu} \tag{A10}$$

$$C_{j_t} = (1-r)C_{j_{t-1}} + r(x_t - \hat{\mu}_t)(x_{t-j}\hat{\mu}_t)^T \tag{A11}$$

3. Solve the Yule–Walker equation for ω_i.

$$\begin{bmatrix} C_{0_t} & C_{1_t} & \cdots & C_{k-1_t} \\ C_{0_t} & C_{0_t} & \cdots & C_{k-2_t} \\ \vdots & \vdots & \ddots & \vdots \\ C_{k-1_t} & C_{k-2_t} & \cdots & C_{0_t} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_k \end{bmatrix} = \begin{bmatrix} C_{1_t} \\ C_{2_t} \\ \vdots \\ C_{k_t} \end{bmatrix} \tag{A12}$$

4. Compute the variables used in the score calculation.

$$\hat{\Sigma}_t = (1-r)\hat{\Sigma}_{t-1} + r(x_t - \hat{x}_t)(x_t - \hat{x}_t)^T \tag{A13}$$

$$\hat{x}_t = \sum_{i=1}^{k} \omega_i (x_{i-j} - \hat{\mu}_t) + \hat{\mu}_t \tag{A14}$$

References

1. Ehrfeld, W.; Hessel, V.; Löwe, H. *Microreactors: New Technology for Modern Chemistry*; Wiley/VCH: Weinheim, Germany, 2000.
2. Yao, X.; Zhang, Y.; Du, L.; Liu, J.; Yao, J. Review of the applications of microreactors. *Renew. Sustain. Energy Rev.* **2015**, *47*, 519–539. [CrossRef]
3. Burns, J.R.; Ramshaw, C. Development of a Microreactor for Chemical Production. *Chem. Eng. Res. Des.* **1999**, *77*, 206–211. [CrossRef]
4. Tamagawa, O.; Muto, A. Development of cesium ion extraction process using a slug flow microreactor. *Chem. Eng. J.* **2011**, *167*, 700–704. [CrossRef]
5. Zhao, D.; Tan, G. A review of thermoelectric cooling: Materials, modeling and applications. *Appl. Therm. Eng.* **2014**, *66*, 15–24. [CrossRef]
6. Najafi, H.; Woodbury, K.A. Optimization of a cooling system based on Peltier effect for photovoltaic cells. *Sol. Energy* **2013**, *91*, 152–160. [CrossRef]
7. Deng, M.; Inoue, A.; Goto, S. Operator based Thermal Control of an Aluminum Plate with a Peltier Device. *Int. J. Innov. Comput. Inf. Control* **2008**, *4*, 3219–3229.
8. Takahashi, K.; Wen, S.; Sanada, M.; Deng, M. Nonlinear cooling control for a Peltier actuated aluminum plate thermal system by considering radiation heat transfer. In Proceedings of the 2012 International Conference on Advanced Mechatronic Systems, Tokyo, Japan, 18–22 September 2012; pp. 18–23.
9. Hwang, I.; Kim, S.; Kim, Y.; Seah, C.E. A survey of fault detection, isolation, and reconfiguration methods. *IEEE Trans. Control Syst. Technol.* **2009**, *18*, 636–653. [CrossRef]
10. Isermann, R.; Balle, P. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Eng. Pract.* **1997**, *5*, 709–719. [CrossRef]
11. Gao, Z.; Cecati, C.; Ding, S.X. A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault Diagnosis with Model-Based and Signal-Based Approaches. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3757–3767. [CrossRef]
12. Zhang, W.; Xu, D.; Enjeti, P.N.; Li, H.; Hawke, J.T.; Krishnamoorthy, H.S. Survey on Fault-Tolerant Techniques for Power Electronic Converters. *IEEE Trans. Power Electron.* **2014**, *29*, 6319–6331. [CrossRef]
13. Venkatasubramanian, V.; Rengaswamy, R.; Kavuri, S.N.; Yin, K. A review of process fault detection and diagnosis: Part III: Process history based methods. *Comput. Chem. Eng.* **2003**, *27*, 327–346. [CrossRef]
14. Takeuchi, J.; Yamanishi, K. A Anifying Framework for Detecting Outliers and Change Points from Time Series. *IEEE Trans. Knowl. Data Eng.* **2006**, *18*, 482–492. [CrossRef]
15. Ogihara, Y.; Deng, M. Operator-based Nonlinear Fault Detection and Fault Tolerant Control for Microreactor using One-Class SVM. *Int. J. Adv. Mechatron. Syst.* **2020**, *8*, 109–115. [CrossRef]
16. Deng, M.; Inoue, A.; Ishikawa, K. Operator-based nonlinear feedback control design using robust right coprime factorization. *IEEE Trans. Autom. Control* **2006**, *51*, 645–648. [CrossRef]
17. Furukawa, Y.; Deng, M. SVM-based fault detection for double layered tank system by considering ChangeFinder's characteristics. *Int. J. Adv. Mechatron. Syst.* **2022**, *9*, 185–192. [CrossRef]
18. Li, D.; Wang, Y.; Wang, J.; Wang, C.; Duan, Y. Recent advances in sensor fault diagnosis: A review. *Sens. Actuators A Phys.* **2020**, *309*, 111990. [CrossRef]
19. Nishizawa, K.; Deng, M. Operator-based Nonlinear Modeling and Control of Microreactor Considering Symmetry. In Proceedings of the 2021 IEEE International Conference on Networking, Sensing and Control (ICNSC), Xiamen, China, 3–5 December 2021; pp. 1–6.
20. Tsutsui, S.; Yamamura, M.; Higuchi, T. Multi-parent recombination with simplex crossover in real coded genetic algorithms. In Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation, Orlando, FL, USA, 13–17 July 1999; pp. 657–664.
21. Eshelman, L.J.; Schaffer, J.D. Real-Coded Genetic Algorithms and Interval-Schemata. *Found. Genet. Algorithms* **1993**, *2*, 187–202.
22. Schölkopf, B.; Platt, J.C.; Shawe-Taylor, J.; Smola, A.J.; Williamson, R.C. Estimating the support of a high-dimensional distribution. *Neural Comput.* **2001**, *13*, 1443–1471. [CrossRef] [PubMed]
23. Yin, S.; Zhu, X.; Jing, C. Fault detection based on a robust one class support vector machine. *Neurocomputing* **2014**, *145*, 263–268. [CrossRef]

applied sciences

Article

Transformer-Based High-Speed Train Axle Temperature Monitoring and Alarm System for Enhanced Safety and Performance

Wanyi Li, Kun Xie *, Jinbai Zou, Kai Huang, Fan Mu and Liyu Chen

School of Railway Transportation, Shanghai Institute of Technology, Shanghai 201418, China; 17717086039@163.com (W.L.); zoujb@sit.edu.cn (J.Z.); hk15988893039@163.com (K.H.); m17793810475@163.com (F.M.); c7012919013@163.com (L.C.)
* Correspondence: mrxk2000@163.com

Abstract: As the fleet of high-speed rail vehicles expands, ensuring train safety is of the utmost importance, emphasizing the critical need to enhance the precision of axel temperature warning systems. Yet, the limited availability of data on the unique features of high thermal axis temperature conditions in railway systems hinders the optimal performance of intelligent algorithms in alarm detection models. To address these challenges, this study introduces a novel dynamic principal component analysis preprocessing technique for tolerance temperature data to effectively manage missing data and outliers. Furthermore, a customized generative adversarial network is devised to generate distinct data related to high thermal axis temperature, focusing on optimizing the network's objective functions and distinctions to bolster the efficiency and diversity of the generated data. Finally, an integrated model with an optimized transformer module is established to accurately classify alarm levels, provide a comprehensive solution to pressing train safety issues, and, in a timely manner, notify drivers and maintenance departments (DEPOs) of high-temperature warnings.

Keywords: data generation; algorithm model; axle temperature alarm; early warning; fault diagnosis

1. Introduction

Railway transportation plays a crucial role in contemporary social development [1]. In recent years, rail transit, represented by high-speed rail, has developed rapidly. As a high-quality "name card" of China, it has promoted the development and realization of the "belt and road" strategy and the "going global" strategy of China's high-speed railway [2]. In the context of railway systems, a train is typically classified as "high-speed" if it can travel at speeds of 250 km/h (155 mph) or more on newly built tracks, or 200 km/h (124 mph) on existing tracks. This classification is essentially based on international standards set by organizations such as the International Union of Railways (UIC) and is crucial for differentiating between conventional and high-speed rail systems. The axles of high-speed trains are specifically designed to withstand these higher operational speeds, which requires enhanced durability and safety standards. How to incorporate the rapid development of big data technology to enhance the safety of train operations has become the top priority of current high-speed train development. As a crucial part of the train, the train bearing and axle box must withstand all the train's weight. Since the train should commonly run at high speed during operation, the vibration shock due to the rough track and irregular road crossings could exhibit a remarkable impact on the bearing, so the most vulnerable parts of the train include the bearings and the axle box [3]. In general, high-speed train bearings are also the main pieces of equipment to ensure the safe operation of trains [4]. When the bearing fails and deteriorates quickly, it even endangers the safety of the train operation [5]. Therefore, effective monitoring and fault diagnosis of high-speed train bearings is an essential way to ensure the safety of train operations [6]. High-speed train axles are subject to significant mechanical and thermal stress during operation, making temperature monitoring crucial for ensuring safety and performance. While much attention has been paid to bearing temperatures, other key components, such as brake systems, also experience substantial thermal loads. In particular, brake friction pairs can be exposed to extreme temperatures, especially under conditions of brake blockage or malfunction. In such scenarios, excessive heat can be generated, which could potentially impact axle safety and performance. Therefore, comprehensive temperature monitoring, which includes the thermal behavior of both bearings and brake systems, is essential to effectively detect and mitigate risks. At present, the bearing fault diagnosis method mainly includes the diagnosis approach based on temperature data [7], the diagnostic method based on fault signal (such as acoustic signal [8,9] and vibration signal [10–12]), the detection and diagnosis method based on ferro-spectral analysis, the oil-based diagnosis method [13], and the diagnosis method based on oil film resistance monitoring. However, in practical application, the bearing monitoring method, which is extensively accepted in China, collects the axial temperature data through the bearing monitoring system's sensor [14].

Since train operation monitoring and fault diagnosis technology plays a significant role in railway transportation, companies and investigators at home and abroad significantly contribute to the research of train operation monitoring and fault diagnosis technology. In the 1980s, the United States developed a bypass acoustic detection system. Through the analysis of the sound signal of the train bearings, it was combined with the infrared thermal shaft detection system to achieve the effect of detecting bearing faults in advance. SKF in Sweden proposed to use the vibration signal and shaft temperature data of bearings to monitor the working state of the axle system. The on-board fault diagnosis system was investigated at the University of Southampton, installed on a passenger train, and then utilized Perpetuum wireless sensors to measure train vibrations. The system transmits raw data and calculates measurement data to the cloud, with train number, wheel position, recording time, speed, position, direction, ambient temperature, and bearing and wheel health indexes. The operators and maintenance personnel can access real-time data through the website, monitor the health status of bearings and wheels, and determine whether failures occur [15]. Until the end of the 20th century, the railways in China began to pay attention to the detection and fault diagnosis of train bearings by utilizing the infrared axle temperature detection system and on-mounted axle temperature detection device. China Railway also adopts monitoring technology at the infrastructure management level as a diagnostic solution to improve the safety and reliability of railway operations. By deploying a large number of sensors and intelligent monitoring systems along the railway, these devices can monitor the vibration, temperature, stress, and other parameters during the operation of the train, especially the health of the axles and the status of the welding points. Although these systems are useful for fault warning, usually when the alarm is issued, the bearing damage has typically advanced to a more serious stage, still posing significant safety risks [16].

Based on the above reasons, herein, the obtained data are first methodically preprocessed, which mainly includes missing values' treatment, outlier treatment, and normalization treatment. The axial temperature feature data with a strong thermal level is difficult to obtain, which is generated by the optimized adversary-generated network, and the objective function and discriminator in the network are optimized to enhance the effectiveness and diversity of the generated data. Finally, an optimal integration model based on the transformer module is designed to detect the alarm level. The novelty of this approach lies in the utilization of an optimized adversary-generated network to create realistic and diverse strong heat state data, overcoming the scarcity of such data. This approach ensures better generalization and robustness in the model's performance, enhancing its ability to accurately detect alarm levels. In this way, when the abnormal increase in bearing temperature is detected, the system will immediately issue a high-temperature warning, and convey in a timely manner the warning information to the driver and relevant service departments (such as maintenance workshop, DEPO), to ensure that potential problems are quickly responded to and dealt with, and ensure the safe operation of the train and the maintenance efficiency of the equipment to the greatest extent.

2. Data Processing

The train's advanced on-board axle temperature monitoring system ensures the accurate collection of bearing temperature data. Using the axle temperature host, the system effectively captures temperature data transmitted by sensors on the bearings and axle boxes. These critical data can then be efficiently transmitted through the network to ground storage or downloaded by personnel using the on-board temperature detection system [17]. The axle temperature data collected by the on-board axle temperature monitoring system on the train often exhibit some deficiencies. In particular, the following two issues are more common:

(1) Communication anomalies often result in partial periods of missing bearing temperature data when the train passes through tunnels or areas with weak network signals.

(2) Temperature sensor failure due to poor connections, electromagnetic interference, and other factors can lead to transient abnormalities in certain axle temperature variables that subsequently recover. These anomalies usually do not occur simultaneously in multiple sensors and therefore lead to outliers in the bearing temperature data.

The proposed methodology here is essentially based on a data-driven latent structure approach that requires normalized data during modeling. Introducing a small amount of missing axle temperature data or outliers during the modeling process could lead to noticeable modeling errors and thus affect the accuracy of fault diagnosis. As a result, appropriate data preprocessing is also performed on the acquired axle temperature data.

2.1. Missing Data Handling

In cases where there are low-density missing values in the data, common approaches include missing data removal, manual completion, regression estimation completion, or data interpolation for handling. Removing missing data can ensure data integrity but may lead to a significant waste of data resources and affect the objectivity and accuracy of the results. Manual completion is labor-intensive and not very accurate, while completing the regression estimate is complex for data with low densities of missing values. Therefore, we herein choose to use linear interpolation, a data interpolation method, as a suitable approach to deal with missing values. Common data interpolation methodologies include spline interpolation, mean imputation, multiple imputation, and maximum likelihood estimation imputation. In the present paper, linear interpolation is employed to interpolate the missing data. The main idea is that, after obtaining the historical bearing temperature data, the axial temperature data and the previous moment of the missing data are selected to estimate the missing data value. In practical applications, the reported temperature typically represents an average or maximum value, which provides a general indication of the thermal state of the axle without delving into the complexities of spatial variations. However, it is crucial to recognize the significance of temperature gradients along the axle. These gradients can cause differential expansion and contraction, leading to additional stresses and potential damage over time [18]. Studies have shown that temperature variations can indeed impact the mechanical properties and longevity of axles [19,20]. Furthermore, the temperature field along the axles is generally both time-varying and spatial-varying. The temperature field is able to vary significantly across the diameter and length of the axle, leading to gradients that may induce forces and stresses, potentially affecting the service life of the axle. These spatial and temporal variations are critical for understanding the thermal behavior of the axle but have not been accurately taken into account in the linear interpolation model used for handling missing data. Linear interpolation, while effective for simple data gaps, does not capture the complex variations in the temperature field. Future research could focus on developing more sophisticated models that account for these spatial temperature variations to enhance predictive maintenance strategies. Advanced interpolation techniques or models that consider both time-varying and spatial-varying characteristics could improve the accuracy and reliability of temperature data reconstruction, thereby providing a more comprehensive understanding of the axle's thermal behavior and contributing to better maintenance and safety strategies.

The experimental data for this study were obtained from an on-board axle temperature monitoring system. We collected axle temperature data using 36 sensors. Data from the first 10 days were utilized to train a neural network model, and 1 day was randomly selected from eight typical locations for missing data processing. Initially, the axle temperature data collected from the temperature sensors on the train bearings were transmitted to the axle temperature monitoring system and then submitted to the ground system through network transmission. Let us assume that the train operating temperature data are represented as a data matrix $\mathbf{X} \in \mathrm{R}^{\mathrm{N} \times \mathrm{m}}$, where N represents the number of samples, and m denotes the number of bearing temperature variables. In addition, the period of missing data is denoted by $t \in [t_1, t_2]$, where t_1 represents the start time of the missing data period, and t_2 denotes the end time of the missing data period. The sampled values of bearing temperature within the period of missing data are empty. Let $\mathbf{X}$ be denoted as follows:

$$\mathbf{X} = \begin{pmatrix} x_1(1) & x_2(1) & \cdots & x_m(1) \\ x_1(2) & x_2(2) & \cdots & x_m(2) \\ \vdots & \vdots & & \vdots \\ x_1(N) & x_2(N) & \cdots & x_m(N) \end{pmatrix} \tag{1}$$

where each row represents the axle temperature data collected at the same time for different bearing temperature variables, while each column denotes the axle temperature data collected at different times for the same bearing temperature variable. Assuming $x_i(k)$ to $x_i(k + \partial)$ for $0 < \partial < N - k$ is represented as a segment of missing data. Firstly, locate the nearest data segments before and after the missing data $x_i(k')$ and $x_i(k'')(k' < k, k + \partial < k'')$.

$$x_{i,L}(k) = x(k') + \frac{x_i(k'') - x(k')}{k'' - k'}(k - k') \tag{2}$$

where $x_{i,L}(k)$ denotes the estimated value of the missing data segment from $x_i(k)$ to $x_i(k + \partial)$, which is the result value that should be inserted into the time period with missing data. The data matrix after the interpolation of the missing data segments is denoted by $\mathbf{X}_\mathrm{s}$.

The experimental data in this article are derived from the on-board axle temperature monitoring system. During transmission, data defects may occur at points with missing data due to weak network signals when the train passes through such areas. Figure 1 shows the shaft temperature data of the vehicle after startup and before storage. A data preprocessing method based on linear interpolation is adopted to properly fill in the missing data points in the bearing temperature data. The missing values in the bearing temperature data exhibit characteristics of sparse points. Interpolation results of missing data are shown in Figure 1.

After interpolation is completed, it is also necessary to verify whether the interpolated data are significantly different from the original data. At this time, the Kolmogorov–Smirnov test (K-S test) is a commonly utilized non-parametric statistical method used to compare whether the distribution of the two sets of data is significantly different. The detailed step of the K-S one-sample test method is to exploit the actual cumulative distribution of sample sampling with the comparison of the hypothetical theoretical distribution, calculate the cumulative distribution function of the two functions, and then evaluate the maximum value D of the distance between the distribution function, and check the D-value distribution table to determine the confidence interval of the D value. If the D value is placed in the corresponding confidence interval, that is, when the maximum difference value D is in the specified numerical range, you can determine the data sample by approximately obeying or obeying the distribution of the hypothesis. However, when the difference exceeds the specified range when testing that the two data samples have a remarkable difference, the detected data samples do not meet the requirements.

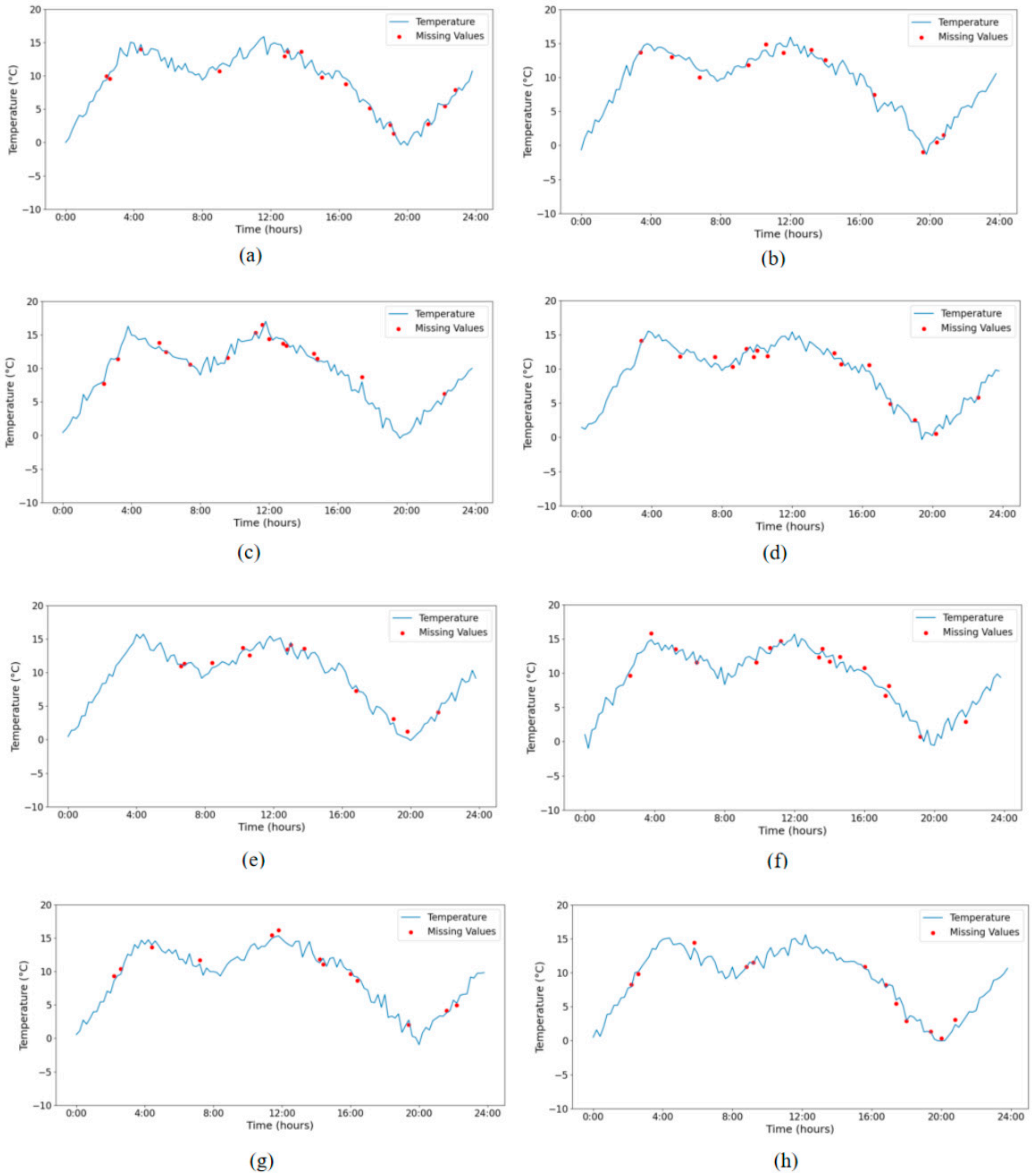


Figure 1. The time-varying temperature of various axles in 2 data samples: (**a**) axle 1; (**b**) axle 2; (**c**) axle 3; (**d**) axle 4; (**e**) axle 5; (**f**) axle 6; (**g**) axle 7; (**h**) axle 8.

In the K-S test, the primary task is to calculate the cumulative empirical distribution function of the two sets of observed data. Assuming that the samples of the datasets are expressed as (x_1, x_2, x_3,. . ., x_n), the cumulative distribution function is expressed as:

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{i} I_{[-\infty,x]}(x_i) \tag{3}$$

where I represents the indicator function, expressed by:

$$I_{[-\infty,x]}(x_i) = \begin{cases} 1, x_i \leq x; \\ 0, x_i > x; \end{cases} \tag{4}$$

The K-S test generates a p-value for judging the distribution of the two sets of data. If the *p*-value is large, it means that the interpolation possesses less influence on the data distribution, and the interpolated data can be taken as consistent with the original data.

2.2. Outlier Treatment

To ensure the accuracy of the modeling and subsequent monitoring and fault diagnosis, outlier treatment is performed on the transient anomalies in certain axle temperature variable data caused by factors such as weak connections and electromagnetic interference in the collected bearing temperature data. Based on industry standards and our findings, the typical operational limit for axle-bearing temperatures in passenger trains usually ranges from 80 °C to 120 °C under normal conditions. However, temperatures could reach up to 140 °C under extreme situations without immediate risk when cooling systems and monitoring mechanisms are in place. Outliers are usually points that are significantly different from the surrounding data, so that a gradually varying temperature gradient should not be considered as outliers. This can happen by considering the continuity of the data, namely that this phenomenon should not be regarded as abnormal if the temperature value gradually rises or decreases. Smooth methods, such as moving window or sliding average, are used to identify progressive trends and avoid misidentification of temperature gradient changes as anomalies. For the anomaly detected automatically by the program, the anomaly is confirmed by manual secondary verification or a more detailed physical measurement. This can effectively prevent the true temperature change from being misjudged.

The outliers present in the data can also be considered as a type of fault. In 2014, Li et al. [21] proposed a reconstruction-based contribution method based on dynamic principal component analysis (DPCA), namely the multi-directional reconstruction contribution method. Using a continuous stirred-tank reactor (CSTR) as an example, the effectiveness of their proposed method was validated. By combining the dynamic characteristics of the axle temperature data used in this study and the long-term scale axle temperature data, the outliers belong to sparse points. Therefore, the present study also attempts to propose an effective approach for outlier reconstruction by combining DPCA and principal component search, mainly dynamic principal component search (DPCS) [22]. To this end, we commence with integrating the DPCA-based approach with variable delay expansion to extract the dynamic relationships between variables, and the matrix $\mathbf{X}_\mathrm{d}$ of delayed data is obtained from $\mathbf{X}_\mathrm{s}$ based on the delay time d in the following form:

$$\mathbf{X}_d = \begin{pmatrix} \mathbf{X}_s^T(1 : N - d + 1, 1 : m) \\ \mathbf{X}_s^T(2 : N - d + 2, 1 : m) \\ \mathbf{X}_s^T(d : N, 1 : m) \end{pmatrix} \tag{5}$$

Based on the delayed data matrix, $\mathbf{X}_\mathrm{d}$ is decomposed into $\mathbf{X}_\mathrm{d}$ = $\boldsymbol{L}$+$\boldsymbol{S}$+$\boldsymbol{N}$ as per the formula, where $\boldsymbol{L}$,$\boldsymbol{S}$, and$\boldsymbol{N}$represent the low-rank data matrix, the sparse outlier data, and noise, respectively. By combining DPCA with DPCS via L_1-norm and convex optimization [23], the method could reconstruct the outlier data to obtain a matrix $\mathbf{X}_\mathrm{r}$ of normal axle temperature data. The results of the outlier treatment of the axial temperature data used in this experiment are shown in Figures 2 and 3.

In order to reduce the experimental error and improve the performance and generalization ability of the machine learning model, this paper selected the k-fold cross-validation method, and the ratio of the training set, validation set, and test set was 7:2:1, and the number of iterations was 50. The original dataset was randomly split into 3 equal-sized sub-cross validation sets, with 2 sub-cross validation sets used for model training and the remaining sub-cross validation set for model testing. The above process was repeated three times, and finally three independent model performance assessments were obtained. Using k-fold cross-validation ensured that the program had consistent performance on different subsets of the data, avoiding overfitting or underfitting.

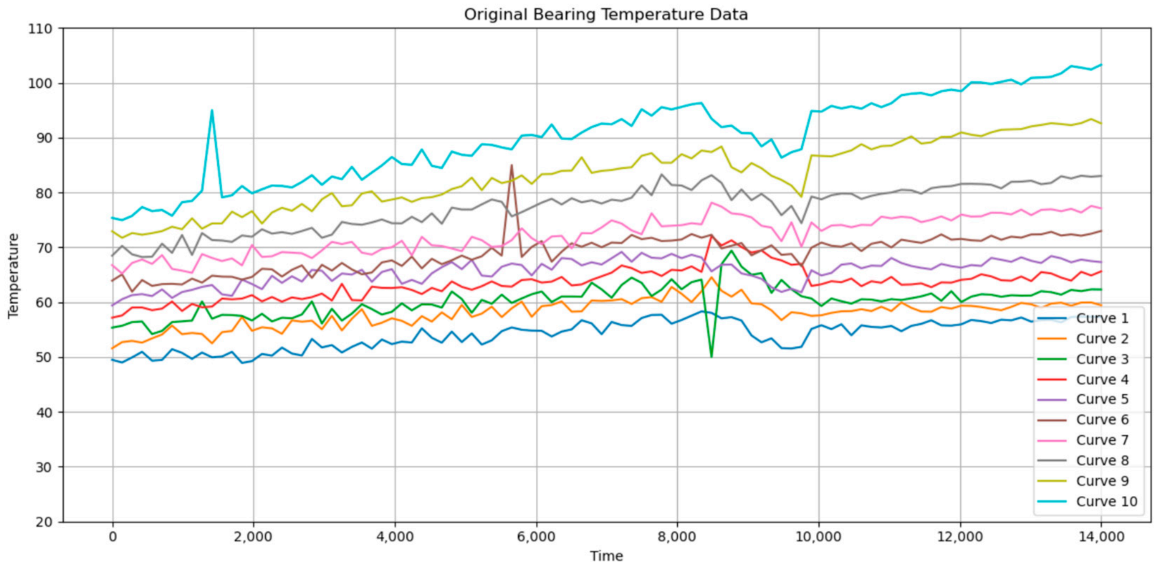


Figure 2. Before data processing of the outlier axle temperature.

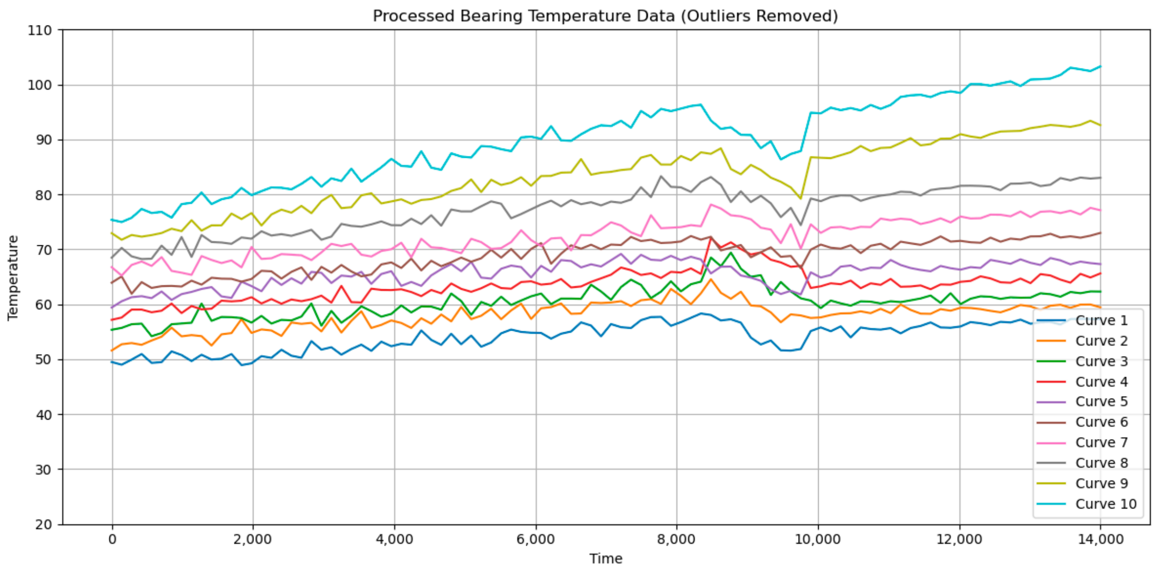


Figure 3. After data processing of the outlier axle temperature.

2.3. Normalization Processing

Due to the various dimensions and magnitudes among various types of data (such as train speed, train axle temperature, and train load), the direct analysis of diverse data would affect the analysis results. Similarly, if there is a significant difference in magnitude between the data used, it would greatly affect the analysis results. To guarantee precise data analysis and reliable data modeling results, it is therefore crucial to first preprocess the raw axle temperature data by addressing missing values and outliers and then applying normalization processing. Normalization processing is able to standardize data of various types on the same scale. In the present investigation, the normalization method applied to the axle temperature data matrix after missing value handling is standard deviation normalization, which makes the data conform to a standard normal distribution with a mean of 0 and a standard deviation of 1 [24]. For the data matrix that has not been normalized, normalization processing is conducted to obtain the normalized data matrix $\boldsymbol{X}_p$:

$$X_p = \frac{X_s - \mu}{\sigma} \tag{6}$$

3. Screening and Regeneration of Strong Thermal Grade Characteristic Data

Due to the infrequent occurrence of strong heat states in trains, the characteristic axle temperature data for such states are relatively scarce. Using a significant amount of normal and mild heat state data alongside a limited quantity of strong heat state data could lead to poor generalization abilities of the trained model. To acquire more data regarding high-intensity heat levels in axle temperature characteristics, this chapter focuses on data generation for robust heat features and presents a methodology for generating simulated data. Based on the processed data, the results are divided into datasets that facilitate the generation of simulated data, ensuring a more balanced and comprehensive representation of high-intensity heat levels.

3.1. Selection of Axle Temperature Features

According to Chinese domestic axle temperature designs, an axle temperature raised to a level in the range of 40–60 °C is taken as mild heat, the so-called alert level; exceeding 60 °C is regarded as intense heat, classified as the alarm level [25]. However, this discrimination approach faces issues. Firstly, it involves numerous parameters such as train weight and speed, which interact and intersect with each other. Secondly, the mixture of old and new bearing types broadens the normal operating temperature range. Therefore, this study selects the temperature rise, column temperature rise difference, and vehicle temperature rise difference as the main features of the axle temperature status. By employing the same train and same car approach, the present study is capable of eliminating the influence of some crucial factors such as vehicle type, speed, and load on the axle temperature. These three indicators also serve as the main basis for axle temperature discrimination by the current HDBS-III type infrared detection equipment, validated through years of on-site experiments for their scientific validity.

The present study utilizes the Pearson correlation coefficient method for feature selection. The collected features related to axle temperature are evaluated via the Pearson correlation coefficient method to determine the correlation coefficient γ between the features and the axle temperature [26]. The range of γ is set as $[-1, 1]$, where a higher correlation coefficient γ indicates a stronger correlation between the feature and the axle temperature. By employing this approach, irrelevant or weakly related features with the target feature are removed from the sample set. The formula for calculating the Pearson correlation coefficient can be provided as:

$$\gamma = \frac{N\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{N\sum x_i^2 - (\sum x_i)^2}\sqrt{N\sum y_i^2 - (\sum y_i)^2}} \tag{7}$$

where N represents the number of feature samples, x_i denotes the axle temperature at the i-th instance, and y_i signifies the feature value at the i-th instance.

In this study, there exists a total of 500 data points for normal and mild heat-level axle temperature states and 200 data points for intense heat-level axle temperature states. Initially, it is necessary to expand the 200 data points for intense heat-level axle temperature states to 500 to serve as the final dataset for intense heat axle temperature states.

3.2. Generation of Intense Heat-Level Axle Temperature Feature Data

Different to the traditional generative network, the adversarial generative network consists of a generative model that captures the training set distribution and a discriminant model that tests the truth and falsehood of the data. The main task of the generative model is to receive a random noise vector as input and translate it into features similar to the real data. The initial output of the generative model may be very random, but as the number of model training epochs increases, it gradually generates more realistic samples. The training goal of the generative model is to deceive the discriminant model, making it impossible to accurately distinguish the generated samples from the real samples. The discriminant model is commonly employed to evaluate the authenticity of the input sample, and it is essentially a dichotomous model. It receives data from the generative model and real data and attempts to correctly classify them as "true" or "false" samples. The training goal of the discriminative model is to distinguish between the generated data and the real data as accurately as possible, forcing the output of the generated model to be more realistic.

This paper utilizes generative adversarial networks (GANs) for data generation. Wang et al. [27] established a fault sample generation approach with heterogeneous imbalanced monitoring data proposed by a modified GAN (the so-called mixed dual discriminator GAN, MD2GAN). As in this article, the first discriminator D is utilized to discriminate whether the generated sample is real, whereas the second discriminator F is employed to discriminate whether the generated sample is a faulty one. During the training process, the generative and discriminator models engage in adversarial learning, until the generated examples are realistic enough that the discriminator cannot effectively distinguish between real and fake examples, reaching a Nash equilibrium [28]. Finally, during sample generation, only the model generator is utilized for the generation process. The flowchart of the overall procedure is presented in Figure 4.

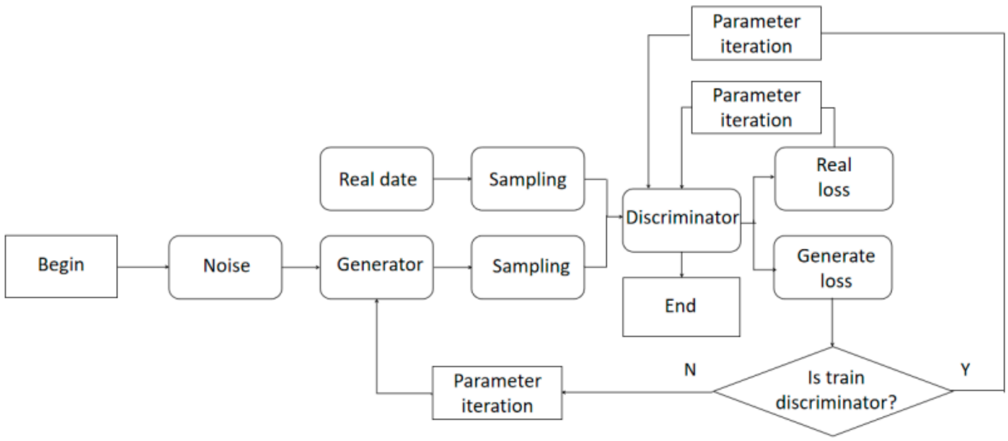


Figure 4. Flowchart of training the adversarial generative networks.

3.3. Discriminative Model Optimization for the Adversarial Generative Model

The reason for the pattern collapse of the GAN network is that the real data often possess a multimodal distribution, but the scoring output of the discrimination model can only be utilized to distinguish whether the input is real data or generated data, and the difference between the characteristics of the data cannot be known. This will cause the generated feature data to be concentrated at the peak of the distribution, while the data at the peak of the other distribution are almost absent. To solve this problem, in this study, an optimization layer is added to its discriminant model, so that different samples can be connected to each other, thus avoiding the pattern collapse. The schematic representation of the added optimization layer is presented in Figure 5.

For the input $\boldsymbol{X}\in R^{N\times A}$ of the optimization layer, it is obtained after the input of the whole discriminant model goes through a fully connected layer. In the optimization layer, it first passes through a trainable 3D matrix $\mathrm{T}\in R^{A\times B\times C}$ to obtain $\mathrm{M}\in R^{N\times B\times C}$, where N represents the total number of samples of the input discriminator. Each sample has B features, and the length of each feature is C. Taking the B feature as an example, the sum of the difference of the B feature of the current sample and the B feature of all samples should be appropriately calculated. The formula is presented in Equation (6). At present, the difference between the two samples is evaluated via the paradigm distance formula L1:

$$\begin{gathered} c_b(x_i, x_j) = exp(-\left\|M_{i,b} - M_{j,b}\right\|_{L1}) \\ o(x_i)_b = \textstyle\sum_{j=1}^{N} c_b(x_i, x_j) \end{gathered} \tag{8}$$

Each feature of each sample is calculated according to the above expression, and the sum of distance differences between the *i*-th sample and the corresponding features of the other sample is taken as the output $o(x_i)$ of the sample after the optimization layer, whose expression is illustrated in Equation (7):

$$o(x_i) = [o(x_i)_1, o(x_i)_2, \ldots, o(x_i)_B\ ] \tag{9}$$

At this time, the output of the optimization layer for all samples is described by $Y \in R^{N\times B}$, whose expression is given by Equation (8):

$$Y = [o(x_1), o(x_2), \ldots, o(x_N)] \tag{10}$$

Finally, the optimization layer of output and input after combining columns as the input is included, and the optimization layer of the output and the basic output of the generated network are added. Thereafter, each sample will receive a score to judge the input data after iteration according to the target function of the model parameters.

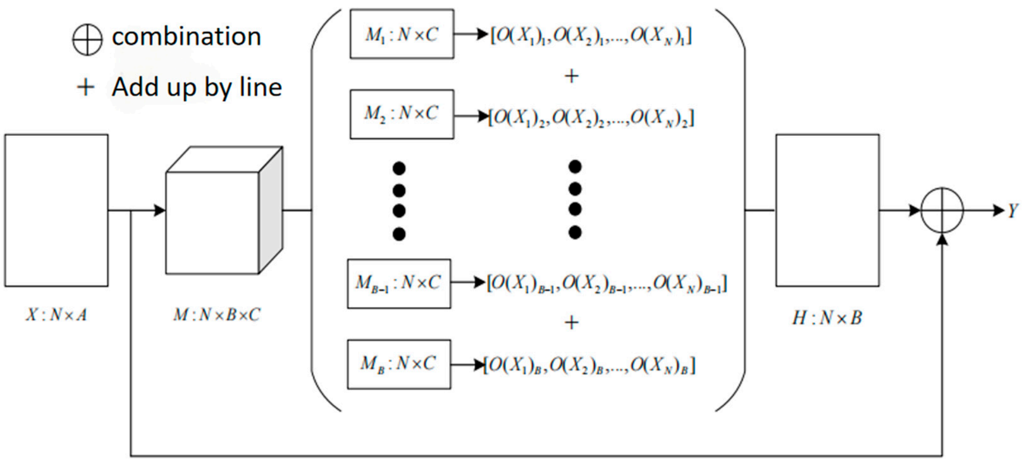


Figure 5. Schematic representation of the optimization layer for the discriminant model.

3.4. Generation of the Strong Thermal Signature Data

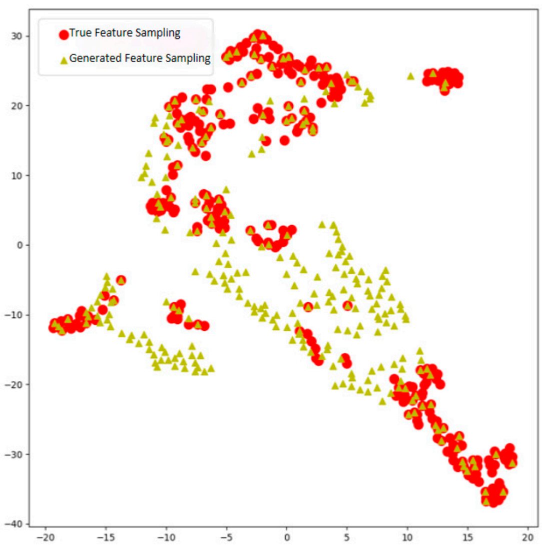
One of the aims of this study is to generate 500 intense heat state feature data based on the existing 200 intense heat state feature data. During the model training process, the shapes of the input noise for the generator are (Batch and Feature), where batch represents the batch of input noise to the generator model, and feature denotes the dimension of the noise shape. At this stage, the Batch value is 40 and the Feature value is 16. During parameter iteration, the detector parameters are fixed when the generator model parameters are updated, and, similarly, the generator parameters are frozen when the detector model parameters are updated [29]. Both models utilize the Adam optimizer with beta parameters (0.5, 0.999). The training phase includes a total of 2000 iterations, where the model storage is performed every 200 iterations. Upon the completion of model training, generating intense heat axle temperature state features only requires the use of the generator model. At this stage, 10,000 noise features of length 16 are input for generation, and 500 generated feature data are randomly selected according to the criteria of concern.

For the three features of temperature rise, column temperature rise difference, and vehicle temperature rise difference, the temperature rise value must be greater than both the temperature rise differences of the column and the vehicle. This limitation is very important. To ensure that the generated intense heat-level feature data demonstrate diversity, the Euclidean distance between the samples generated by the generator G must exceed 2, which ensures great dissimilarity between the samples. In addition, a random selection of generated and authentic intense heat samples can be performed, followed by dimensionality reduction to two dimensions using the T-SNE method, to facilitate subsequent training of the alarm system model [30]. Subsequently, the reduced dimensional data for real and generated data are extracted separately and visualized, as depicted in Figure 5.

The left shows the feature data generated using the adversarial generative network before optimization, and the left demonstrates the optimized feature data on the right. In fact, the principle of T-SNE is a probabilistic-based approach to measure the similarity between high dimensional data points and try to preserve these similarities in low dimensional space. Therefore, the results of the feature data of the real strong thermal axis temperature state in the two graphs after employing the T-SNE algorithm are not identical, but this does not affect the distinction between the distribution of the strong thermal feature data generated before and after optimization and the real feature distribution. After comparison, the two graphs show that the data distribution of the optimized adversarial generated network is closer to the distribution of real data compared with the unoptimized generated data.

In the optimized adversarial generated network according to the definition of adversarial generative network, in the process of network training, the parameters of the discriminative model are iterated by comparing the real feature data and the feature data generated by the generative model through the difference output after the discriminative model. With the increase in the iterations of the adversarial generation network, the difference between the two will gradually decrease, making it impossible for the discriminant model to judge whether the data are generated or real. Based on this idea, the quality of the generated data is judged by the comparison of the difference between the output of the generated feature data and the real data input. The experimental results show that the absolute difference of the output before optimization is 0.39 and the optimized output is 0.25, indicating that the quality of the strong thermal axis temperature state feature data generated after optimization is higher.

Figure 6 indicates on the right side that the strong thermal-level axial temperature data can generate the characteristic data, and the real characteristic data distribution is roughly the same. After dimension reduction in many real feature data distributions, denser peaks exhibit more generated feature data. In addition, these data indicate that it is easier to identify the strong thermal-level of axial temperature data, generated in the peak connection and the rest of the axial temperature state of the strong thermal characteristic data. The addition of these axial temperature data could assist the alarm model to better distinguish between the two states of strong heat and micro heat.



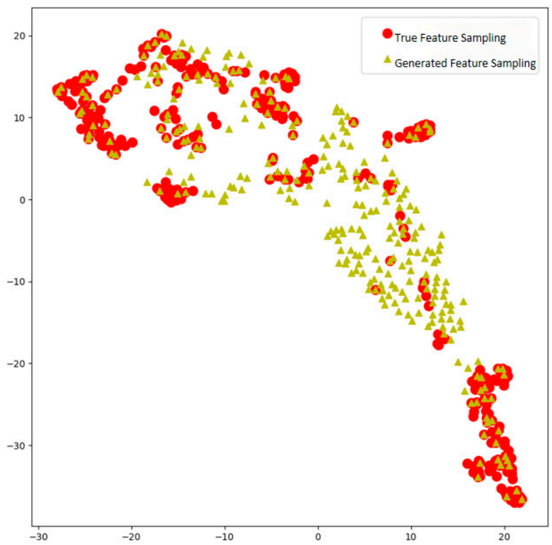


Figure 6. Comparison of data visualization of strong thermal features generated before and after optimization.

4. Intelligent Algorithm Model for Determination of the Axle Temperature Alarm Level

4.1. Training the Axle Temperature Alarm-Level Discrimination Model

This paper adopts a discriminative approach that determines the relationship weights between sequences through the collective action of multiple attention mechanism layers and finally integrates the results of each attention mechanism layer to obtain the final information [31]. In terms of training the alarm level discrimination network, the whole discrimination model consists of a transformer module and an output layer. To construct the transformer, the most crucial aspect of the whole model is the selection of the number of layers of the attention mechanism and the number of neurons in the hidden layer. Since the number of input features is small, only one transformer module is utilized to extract information from the input sequence samples. The components of a transformer are presented in Table 1.

Table 1. Test set accuracy and loss values.

The Fold Times k	Loss Value/%	Precision/%
1	4.63	98.62
2	9.52	97.88
3	7.36	98.03
Average	7.17	98.17

As shown in Table 2, is a component of a Transformer. For the basic axle temperature alarm-level determination model, since the total number of parameters in the model is very small, with only fifty thousand learnable parameters, the excessive number of neurons in the hidden layer could lead to overfitting of the discrimination model [31]. Therefore, the number of neurons in the hidden layer is herein set as 64. Additionally, to ensure the effectiveness of the multi-head attention mechanism layer, the dimensions of the Q, K, and V vectors for each attention mechanism cannot be too low. Simultaneously, the number of neurons in the hidden layer must be divisible by the number of attention layers in the multi-head attention mechanism layer. Therefore, the number of attention layers in the multi-head attention mechanism layer is set to 2, which means that the dimensions of the Q, K, and V vectors for each attention mechanism are 32.

Table 2. Structure of the transformer model.

Layer Type	Output Type	Parameter Value
Embedding	(Batch, 3, 3)	256
Multi-head attention	(Batch, 3, 64)	16,640
Add	(Batch, 3, 64)	0
Layer norm	(Batch, 3, 64)	0
FFN	(Batch, 3, 64)	33,088
Add	(Batch, 3, 64)	0
Layer norm	(Batch, 3, 64)	0
Output	(Batch, 3, 4)	260

The alarm-level discrimination model benefits from two sets of inputs: the sequence features of the input sequence and the position vector of the detection station. The sequence features of the input sequence, denoted by **X** in the model training phase, have the form (Batch, Seq, Feature), where the first, second, and third dimensions represent the data batch processed simultaneously, the number of detection stations considered, and the selected number of features, respectively. In the model training stage, Batch in **X** can be selected based on the computer performance, while Seq and Feature sizes are fixed at 3. For instance, consider the *i*-th sample $\mathbf{X}_\mathrm{i}$ within a sequence of samples **X** as follows: $\mathbf{X}_\mathrm{i}$ = (x1, x2, x3), where $\mathbf{X}_\mathrm{i}$ represents a vector of axle temperature features detected by the detection station, with corresponding values (f1, f2, f3). Specifically, the arrangement of axle temperature feature data obtained by the *i*-th detection station in order are temperature rise, column temperature rise difference, and vehicle temperature rise difference. For the sequence length in the alarm rule Seq less than 3 optimized timing sample data, the missing part of the corresponding vector is [−2, −2, −2]. For example, for a training sample, only the first detection station of axial temperature characteristic information, then for the second and three detection stations corresponding to the vector [−2, −2,−2]. For another discriminant model, the input position vector is denoted by P, which is a fixed vector [1, 2, 3], so the input formula of the multi-head attention mechanism layer is given by:

$$E = drop\left(W_{xe}X + W_{pe}P\right) \tag{11}$$

In Formula (9), W_{xe} is the timing of the input feature vector by embedding layer mapping to hidden layer neurons a number of the weight matrix, W_{pe} is used for position information mapping weight matrix, eventually both and through the dropout layer containing the output of position information and characteristic information of E, after the long attention mechanism layer and the residual connection for the first time.

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W_o \tag{12}$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{13}$$

$$Attention(Q, K, V) = softmax(QK^T / \sqrt{d_k}\Big)V \tag{14}$$

$$A = E + Norm(MultiHead(Q, K, V)) \tag{15}$$

The first formula above represents the output of the multi-head attention mechanism layer composed of the output of multiple attention mechanism layers. W_o is the output transformation matrix. h represents the number of attention mechanism layers contained in the multi-head attention mechanism layer. The output of each header can be expressed by the second formula, where ${W_i}^Q$, ${W_i}^K$, and ${W_i}^V$ are the query, key, and value transformation matrix of the *i*-th head. Q, K, and V represent the query vector, the key vector, and the value vector, respectively. In this study, since the self-attention mechanism is used, the values of *Q*, *K*, and *V* are all from *E*. The output of the attention mechanism layer is calculated as shown in the third formula, where $\mathrm{d_k}$ is the dimensional size of the vector Q. The final output is shown in the fourth formula, where the residual connection and norm layer normalization are used to improve the generalization ability of the model.

Then, there is a normalization layer, which is composed of two fully connected layers, and the final output result is obtained through the second residual connection.

4.2. Axle Temperature Alarm-Level Discrimination

For the prediction stage, the input is the same as the training stage, but, at this point, the Batch value of X is fixed to 1, Feature to 3, and Seq can be of any length. Figure 7 illustrates the input scenario during the model's decision-making process.

The entire section in Figure 7 represents the sequence of input features X to the model during the prediction process. Taking a time window as an example, the width of the time window corresponds to the sequence length Seq in the training process of the discrimination model, while the height represents the selected number of features, the so-called Feature. As the time window continuously slides, the single-time feature samples obtained by detection station 3 will integrate the axle temperature feature information obtained by the previous two detection stations. In the case of a train start-up, when the sequence length Seq is less than 3, only the feature data obtained by the current detection station are considered. In the final validation process of this study, the accuracy of detection is considered based on various sequence lengths of feature data input to the discrimination model. The values of the input sequence length range from 1 to 10, which ensures the efficiency of the model in predicting alarm levels.

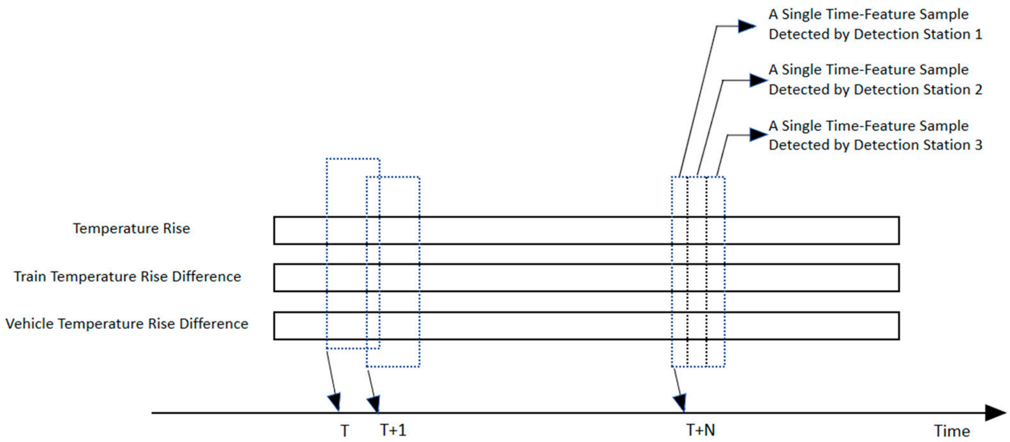


Figure 7. Input to the prediction stage model.

4.3. Optimization of the Axle Temperature Alarm-Level AdamW-Based Discriminator Model

In the former process of determination of the axle temperature alarm level, only the prediction and evaluation of axle temperature alarm levels are conducted. At this stage, the model acquires the probability values associated with various levels of alarm grades through three main steps. Firstly, useful information is extracted through the transformer model, and after passing through a fully connected layer, the output 0 is obtained. The shape of 0 is (Batch, 3, 4). After that, the maximum sequence corresponding to the predictions from the second dimension 0 is extracted (from right to left, where the rightmost is the first dimension), resulting in the prediction results in Pr with a shape of (Batch, 4). Finally, Pr is normalized along the first dimension to obtain the probability values of the model's judgments at various alarm levels of the input data. In this step, the indicators used for model parameter iteration include the model-predicted probabilities of axle temperature alarm levels and the corresponding cross-entropy loss function values for the real alarm levels.

To optimize the existing loss function, the value of the cross-entropy loss function is therefore included between the predicted probability of axle temperature states based on the detection station judgment and the true axle temperature state labels. Consequently, for the output of the transformer module in the model, two distinct fully connected layers are required to extract the probabilities of each axle temperature alarm level and the probabilities of the axle temperature states for the three detection stations, as shown in Figure 8.

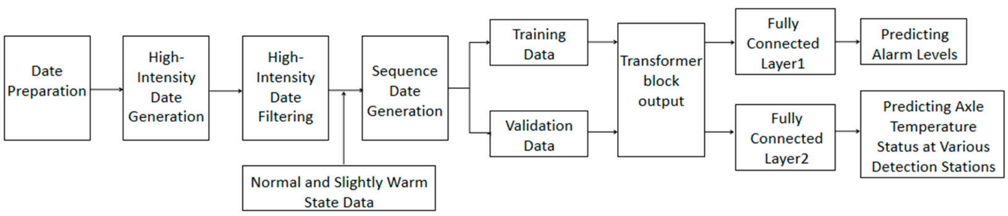


Figure 8. Input to the new prediction stage model.

Due to the sequential nature of detection at detection stations, the input feature vector of the current detection station to the model should only interact with the features obtained from previous detection stations through the attention mechanism (i.e., interaction with the features obtained from subsequent detection stations should not occur). It is hence crucial to optimize the multi-head attention mechanism in the alarm-level discrimination by incorporating the masked multi-head attention mechanism. The loss function at this point can be calculated through the following formula:

$$L = \sum_{n=1}^{N} -\log(y_i^{(n)}) + \omega \sum_{n=1}^{N} -\lambda_1 \log(p_1^{(n)}) - \lambda_2 \log(p_2^{(n)}) - \lambda_3 \log(p_3^{(n)}) \tag{16}$$

In Equation (6), the first term on the right-hand side signifies the loss caused by the prediction of axle temperature alarm levels for the current batch of feature data, whereas the remaining terms denote the loss caused by the axle temperature states of the three detection stations. In addition, ω represents the proportion of the axle temperature state prediction to the total loss. The factors λ_1, λ_2, and λ_3 denote the weights of the entire axle temperature state loss pertinent to the first, second, and third detection stations, respectively, while $p_1^{(n)}$, $p_2^{(n)}$, and $p_3^{(n)}$ represent the model output probabilities associated with the correct axle temperature states for the first, second, and third detections. It should be herein emphasized that, in the case of a train just starting up or restarting (when the sequence length is less than 3 and the missing parts are replaced with vectors all set to -2), the corresponding loss should be removed.

The iteration curves for the training and validation sets at λ_1, λ_2, and λ_3 (corresponding to 1, 3, and 9) are presented in Figure 9.

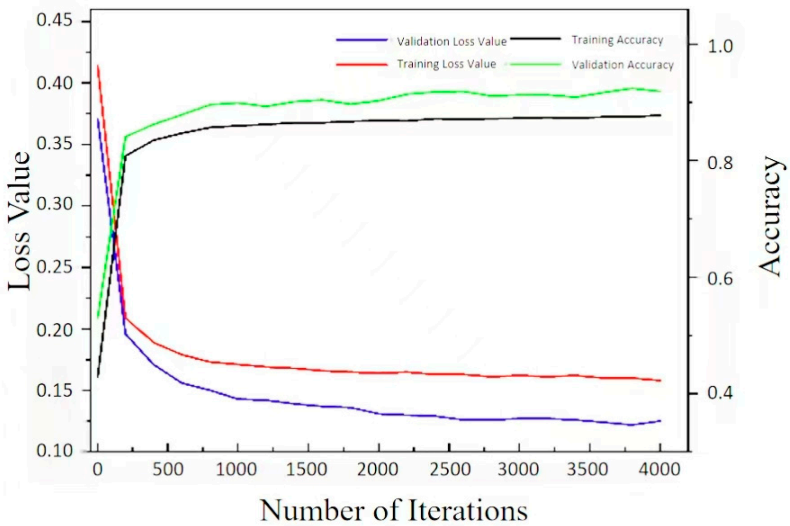


Figure 9. Variations of λ1, λ2, and λ3 during iteration process.

The plotted results in Figure 9 reveal that, when the number of iterations reaches 4000, the validation loss and training loss become almost stabilized. At this stage, the accuracy of the training set reaches 87.8% and the accuracy of the validation set touches 90.9%. Subsequently, using the remaining 10% of "normal"- and "mild"-level feature data, along with all the real "intense"-level feature data, a test set sample is constructed via the same methodology and input in the trained model. The obtained results are indicative of the accuracy of the test set being 86.9% compared with designing an axle temperature alarm system based solely on the axle temperature states detected at each monitoring station, which yields an alarm accuracy of 76.8%, exhibiting an accuracy increase of 13.2%. In addition to the AdamW optimizer, incorporating momentum-based optimizers like Nesterov accelerated gradient (NAG) or using optimization algorithms such as RMSprop or AdaGrad might yield faster convergence. It should be emphasized here that gradient clipping could prevent exploding gradients, enhancing stability during training. Furthermore, for the trained model at this stage, predicting the axle temperature alarm level at time T only requires inputting the current detected axle temperature feature data. The model automatically combines previous axle temperature feature data to provide an alarm level at time T.

5. Conclusions and Perspectives

This study conducted data preprocessing on the collected data to effectively utilize temperature rise features, along with column and vehicle temperature rise difference features for axle temperature alarms. It also integrated the existing method of axle temperature alarm-level discrimination based on these features. Addressing the challenges posed by the infrequent occurrence and difficulty in obtaining data related to intense heat axle temperature states in trains, which results in a scarcity of intense heat axle temperature feature data and hinders the training of a generalized axle temperature alarm-level discrimination model, this study developed an optimized generative adversarial network (GAN) for simulating the generation of limited intense heat feature data. Subsequently, the generated intense heat feature data were integrated with existing intense heat, mild heat, and normal feature data to generate sequence feature data for the warning-level discrimination model. Finally, using a transformer block, the alarm-level discrimination model enhanced the loss function by introducing cross-entropy loss based on the axle temperature states. The present study aims to provide comprehensive training, validation, prediction, and in-depth analysis on hyperparameter design, loss function optimization, and model integration.

Author Contributions: Conceptualization, W.L. and K.X.; methodology, W.L.; software, W.L.; validation, W.L. and K.H.; formal analysis, K.X. and J.Z.; investigation, W.L.; resources, F.M. and L.C.; data curation, K.H.; writing—original draft preparation, W.L.; writing—review and editing, K.X.; visualization, W.L.; supervision, K.X. and J.Z.; project administration, K.X. and J.Z.; funding acquisition, K.X. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Shanghai Science and Technology Commission "Belt and Road" China-Laos Railway Project International Joint Laboratory (No. 21210750300).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in this article; further inquiries can be directed to the corresponding author/s.

Acknowledgments: We would like to thank the editor and the anonymous referees for their valuable comments and suggestions that greatly improved the presentation of this work. This work was supported by various funding sources, as detailed in the funding section.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lukianenko, N. Epistemological research problems of rail transport as a social institution. *Transp. Res. Proceed.* **2022**, *63*, 1826–1833. [CrossRef]
2. Chen, X. Cross-cultural communication in the belt and road strategy. *Front. Soc. Sci. Technol.* **2021**, *3*, 48–56. [CrossRef]
3. Tang, W.C.; Wang, M.J.; Chen, G.D. Analysis on temperature distribution of failure axle box bearings of high speed train. *J. China Railw. Soc.* **2016**, *38*, 50–56. [CrossRef]
4. Randall, R.B. *Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2011; pp. 13–20.
5. Yi, C.; Lin, J.; Zhang, W.; Ding, J. Faults diagnostics of railway axle bearings based on IMF's confidence index algorithm for ensemble EMD. *Sensors* **2015**, *15*, 10991–11011. [CrossRef]
6. Henao, H.; Kia, S.H.; Capolino, G.-A. Torsional-vibration assessment and gear-fault diagnosis in railway traction system. *IEEE Trans. Ind. Electron.* **2011**, *58*, 1707–1717. [CrossRef]
7. Tchakoua, P.; Wamkeue, R.; Ouhrouche, M.; Slaoui-Hasnaoui, F.; Tameghe, T.A.; Ekemb, G. Wind turbine condition monitoring: State-of-the-art review, new trends, and future challenges. *Energies* **2014**, *7*, 2595–2630. [CrossRef]
8. Kilundu, B.; Chiementin, X.; Duez, J.; Mba, D. Cyclostationarity of Acoustic Emissions (AE) for monitoring bearing defects. *Mech. Syst. Signal Process.* **2011**, *25*, 2061–2072. [CrossRef]
9. Eftekharnejad, B.; Carrasco, M.R.; Charnley, B.; Mba, D. The application of spectral kurtosis on acoustic emission and vibrations from a defective bearing. *Mech. Syst. Signal Process.* **2011**, *25*, 266–284. [CrossRef]

10. Sun, H.; Zi, Y.; He, Z. Wind turbine fault detection using multiwavelet denoising with the data-driven block threshold. *Appl. Acoust.* **2014**, *77*, 122–129. [CrossRef]
11. Ming, A.B.; Zhang, W.; Qin, Z.Y.; Chu, F.L. Envelope calculation of the multi-component signal and its application to the deterministic component cancellation in bearing fault diagnosis. *Mech. Syst. Signal Process.* **2015**, *50*, 70–100. [CrossRef]
12. Zimroz, R.; Bartelmus, W.; Barszcz, T.; Urbanek, J. Diagnostics of bearings in presence of strong operating conditions non-stationarity—A procedure of load-dependent features processing with application to wind turbine bearings. *Mech. Syst. Signal Process.* **2014**, *46*, 16–27. [CrossRef]
13. Kharche, P.P.; Kshirsagar, S.V. Review of fault detection in rolling element bearing. *Int. J. Innov. Res. Adv. Eng.* **2014**, *1*, 169–174.
14. Liu, C.; Wang, F. A review of current condition monitoring and fault diagnosis methods for low-speed and heavy-load slewing bearings. In Proceedings of the 2017 9th International Conference on Modelling, Identification and Control (ICMIC), Kunming, China, 10–12 July 2017; pp. 104–109.
15. Corni, I.; Symonds, N.; Wood, R.J.K.; Wasenczuk, A.; Vincent, D. Real-time on-board condition monitoring of train axle bearings. In Proceedings of the Stephenson Conference, London, UK, 21–23 April 2015; p. 14.
16. Jayaswal, P.; Wadhwani, A.K.; Mulchandani, K.B. Machine fault signature analysis. *Int. J. Rotat. Mach.* **2008**, *2008*, 583982. [CrossRef]
17. Singh, K. Smart Components: Creating a Competitive Edge through Smart Connected Drive Train on Mining Machines. Master's Thesis, KTH, School of Industrial Engineering and Management (ITM), Stockholm, Sweden, 2021.
18. Xu, Q.; Sun, S.; Xu, Y.; Hu, C.; Chen, W.; Xu, L. Influence of temperature gradient of slab track on the dynamic responses of the train-CRTS III slab track on subgrade nonlinear coupled system. *Sci. Rep.* **2022**, *12*, 14638. [CrossRef] [PubMed]
19. Yang, L.; Xu, P.; Yang, C.; Guo, W.; Yao, S. High-temperature mechanical properties and microstructure of 2.5 DC/C–SiC composites applied for the brake disc of high-speed train. *J. Eur. Ceram. Soc.* **2024**, *44*, 116683. [CrossRef]
20. Kebede, Y.B.; Yang, M.-D.; Huang, C.-W. Real-time pavement temperature prediction through ensemble machine learning. *Eng. Appl. Artif. Intell.* **2024**, *135*, 108870. [CrossRef]
21. Li, G.; Qin, S.J.; Chai, T. Multi-directional reconstruction based contributions for root-cause diagnosis of dynamic processes. In Proceedings of the 2014 American Control Conference, Portland, OR, USA, 4–6 June 2014; pp. 3500–3505.
22. Song, Y.; Ma, Q.; Zhang, T.; Li, F.; Yu, Y. Research on fault diagnosis strategy of air-conditioning systems based on DPCA and machine learning. *Processes* **2023**, *11*, 1192. [CrossRef]
23. Candès, E.J.; Li, X.; Ma, Y.; Wright, J. Robust principal component analysis? *J. ACM* **2011**, *58*, 1–37. [CrossRef]
24. Li, Z.; He, Q. Prediction of railcar remaining useful life by multiple data source fusion. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2226–2235. [CrossRef]
25. Yan, G.; Yu, C.; Bai, Y. A new hybrid ensemble deep learning model for train axle temperature short term forecasting. *Machines* **2021**, *9*, 312. [CrossRef]
26. Pan, Z.; Xu, D.; Zhang, Y.; Wang, M.; Wang, Z.; Yu, J.; Zhang, G. New energy transmission line fault location method based on Pearson correlation coefficient. In Proceedings of the 2nd International Conference on Smart Energy, Fenghuang, China, 29–30 July 2024; p. 012007.
27. Wang, C.; Liu, J.; Zio, E. A modified generative adversarial network for fault diagnosis in high-speed train components with imbalanced and heterogeneous monitoring data. *J. Dyn. Monit. Diagn.* **2022**, *1*, 84–92. [CrossRef]
28. Jabbar, A.; Li, X.; Omar, B. A survey on generative adversarial networks: Variants, applications, and training. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–49. [CrossRef]
29. Yildirim, M.; Sun, X.A.; Gebraeel, N.Z. Sensor-driven condition-based generator maintenance scheduling—Part I: Maintenance problem. *IEEE Trans. Power Syst.* **2016**, *31*, 4253–4262. [CrossRef]
30. Matetić, I.; Štajduhar, I.; Wolf, I.; Ljubic, S. A review of data-driven approaches and techniques for fault detection and diagnosis in HVAC systems. *Sensors* **2022**, *23*, 1. [CrossRef]
31. Lv, H.; Chen, J.; Pan, T.; Zhang, T.; Feng, Y.; Liu, S. Attention mechanism in intelligent fault diagnosis of machinery: A review of technique and application. *Measurement* **2022**, *199*, 111594. [CrossRef]

applied sciences

MDPI

Article

Fault Feature Extraction Using L-Kurtosis and Minimum Entropy-Based Signal Demodulation

Surinder Kumar [1], Sumika Chauhan [2], Govind Vashishtha [2,3,*], Sunil Kumar [1] and Rajesh Kumar [1]

[1] Precision Metrology Laboratory, Department of Mechanical Engineering, Sant Longowal Institute of Engineering and Technology, Longowal 148106, India; surinderthakur38@gmail.com (S.K.); sunil_thappa@yahoo.com (S.K.); rajesh_krts@sliet.ac.in (R.K.)
[2] Faculty of Geoengineering, Mining and Geology, Wroclaw University of Science and Technology, Na Grobli 15, 50-421 Wroclaw, Poland; sumi.chauhan2@gmail.com
[3] Department of Mechanical Engineering, Graphic Era Deemed to be University, Dehradun 248002, India
* Correspondence: govindyudivashishtha@gmail.com

Abstract: The health of mechanical components can be assessed by analyzing the vibration and acoustic signals they produce. These signals contain valuable information about the component's condition, often encoded within specific frequency bands. However, extracting this information is challenging due to noise contamination from various sources. Narrow-band amplitude demodulation presents a robust technique for isolating fault-related information within the signal. This work proposes a novel approach based on cluster-based segmentation for demodulating the signal and extracting the frequency band of interest. The segmentation process leverages the criteria of maximum L-kurtosis and minimum entropy. L-kurtosis maximizes impulsiveness in the signal, while minimum entropy signifies a low degree of randomness and high cyclo-stationarity, and both characteristics are crucial for identifying the desired frequency band. Simulations and experimental tests using vibration signals from different gears demonstrate the effectiveness of this technique. The processed envelope of the signal exhibits distinct improvements, highlighting the ability to accurately extract the fault-related information embedded within the complex noise-ridden signals. This approach offers a promising solution for accurate and efficient fault diagnosis in mechanical systems, contributing to enhanced reliability and reduced downtime.

Keywords: frequency band; demodulation; cluster-based segmentation; vibration signal; gear

1. Introduction

The condition monitoring and fault diagnosis of rotary components remains a crucial area of research within the scientific community. The ability to accurately assess the health of these components is essential for ensuring operational reliability, preventing catastrophic failures, and minimizing downtime in industrial processes [1–3].

Vibration and acoustic signals emitted by mechanical components offer a rich source of information about their health status. These signals contain valuable clues about the internal condition of the component, including wear, damage, or impending failure. The presence of specific frequency components in these signals can be directly linked to specific faults within the component [4–6].

Vibration-based techniques have emerged as a powerful tool for fault diagnosis in rotating machinery [7–9]. These techniques leverage the relationship between the frequencies present in the vibration signal and the corresponding faults within the machine. Spectral analysis of the vibration signal is a fundamental approach for extracting this crucial information [10]. For instance, Sepulveda and Sinha [11] successfully demonstrated the viability of a two-step artificial neural network approach for condition monitoring and fault diagnosis in rotating machinery. This approach has the potential to significantly improve operational efficiency by enabling proactive maintenance and minimizing costly downtime caused by unexpected breakdowns. Almutairi et al. [12] presented a compelling approach to vibration-based fault detection in rotating machinery, demonstrating the effectiveness of integrating the poly-coherent composite spectrum (pCCS) with machine learning techniques. By effectively condensing multi-location vibration data into a single spectrum, pCCS significantly reduces the computational burden associated with large datasets, while preserving critical amplitude and phase information. Bendjama [13] presented a novel and effective feature extraction method for enhancing the accuracy of rolling bearing fault diagnosis under time-varying operating conditions. By combining the complementary ensemble empirical mode decomposition (CEEMD), Teager–Kaiser energy operator (TKEO), and self-organizing map (SOM), the proposed approach effectively removes unwanted noise components and extracts relevant bearing characteristics from the vibration signal. Tahmasbi et al. [14] demonstrated the successful diagnosis and root cause analysis of bearing failure in a 2300 kW motor, highlighting the importance of a comprehensive approach that considers both vibration and lubrication factors.

However, the conventional approach provides a robust framework for fault diagnosis in rotating machines but limitations arise from the presence of noise and the complex nature of real-world signals. Recent research focuses on developing advanced signal-processing techniques and machine learning algorithms to enhance the accuracy and efficiency of fault diagnosis in rotating machinery.

The accurate demodulation of vibration signals is crucial for extracting fault-related information. The selection of the appropriate frequency band for filtering plays a vital role in achieving this accuracy. Several techniques have been developed in recent years to effectively extract the informative frequency band. Antoni's Spectral Kurtosis (SK) method has proven to be highly effective for detecting transient events and pinpointing their location within the frequency spectrum. Notably, SK exhibits robustness against additive stationary noise. This technique provides a valuable tool for isolating the relevant frequency band within the vibration signal, paving the way for more precise fault diagnosis in rotating machinery [15–19]. Borghesani et al. [20] investigated the effectiveness of cepstrum pre-whitening as a technique for enhancing rolling element-bearing diagnostics, particularly in industrial settings where heterogeneous vibration sources introduce challenges for traditional signal processing methods. The study demonstrated that cepstrum pre-whitening, with its low computational requirements and simplicity, offers a promising alternative to traditional pre-whitening techniques like linear prediction filters and spectral kurtosis. Sacerdoti et al. [21] presented a comprehensive comparative study of signal analysis techniques for rolling element-bearing diagnostics, focusing on fault detection, diagnosis, and prognosis. The study revealed that statistical parameters like RMS, kurtosis, and detectivity are effective in detecting the onset of faults in bearings.

Antoni and Randall proposed the first systematic approach of band selection using kurtogram. The optimal central frequency f_c and bandwidth of the band pass filter can be determined by maximizing the kurtogram [22]. Berrouche et al. [23] proposed a Single Ensemble Empirical Mode Decomposition (SEEMD) method, which demonstrates a significant advancement in fault detection for rolling element bearings, particularly within the challenging context of mining industry signals. Wang et al. [24] proposed a novel and effective method for enhancing early weak fault features in rolling element bearings by combining resonance sparse decomposition with multi-objective information frequency band selection (MIFBS). This approach effectively addresses the challenge of interference components obscuring early fault signatures in vibration signals. Hebda-Sobkowicz et al. [25] provided a comprehensive comparative analysis of various signal-processing techniques designed to extract the signal of interest (SOI) from non-Gaussian noise in vibration signals acquired from machinery. Schmidt et al. [26] addressed the critical challenge of identifying informative frequency bands for extracting weak damage components in vibration signals acquired from rotating machines operating under time-varying conditions. A novel framework is proposed that leverages angle-frequency instantaneous power spectrum and order-frequency cyclic modulation spectrum to construct consistent feature planes.

Zhao et al. [27] proposed a novel and effective method for early fault detection in rolling element bearings, addressing the challenge of weak fault features being masked by environmental noise. The proposed method integrates the negative entropy of the square envelope spectrum approach with optimized stochastic resonance (SR) signal enhancement. Wu et al. [28] presented a novel approach to constructing envelope spectra for the surveillance and diagnostics of rotating machinery, addressing limitations of existing methods like narrowband demodulation and cyclo-stationary analysis. The proposed Combined Weighted Envelope Spectrum (CWES) framework is designed to extract specific characteristic frequencies, effectively mitigating challenges posed by low SNR conditions, non-Gaussian noise, and cyclo-stationary noise.

However, the current methods often struggle with non-stationary noise, common in real-world scenarios. This limits their accuracy and reliability, particularly for early fault detection. Many techniques rely on predefined or manually selected frequency bands, which may not be optimal for all fault types or operating conditions. Developing adaptive methods that automatically identify the most informative frequency bands is crucial. The existing techniques often assume stationary operating conditions, which are not representative of real-world machinery, especially in industries like wind turbines. Further research is needed to develop methods that can reliably function under time-varying conditions.

Thus, an attempt is made in this work to address these issues. In this current work, cluster-based segmentation is used to divide signals into different segments. The frequency band and their central frequencies are automatically estimated using fuzzy C-mean clustering. The informative frequency band among the different segments was selected based on a combination of maximum L-kurtosis and minimum entropy in the segments [29]. This informative frequency band is further utilized to design a band pass filter to filter the original signal. Hilbert's transform-based envelope spectrum is applied to demodulate the filtered signals for extraction of the fault frequency.

2. Background

2.1. Clustering-Based Segmentation

Clustering-based segmentation is a powerful technique for dividing data into distinct groups based on similarities within the data points. This method is particularly useful for analyzing complex signals, such as those generated by mechanical components, where identifying specific frequency bands containing relevant information is crucial for fault diagnosis. The process involves grouping data points that share similar characteristics, such as frequency, amplitude, or other relevant features. These clusters represent distinct segments within the signal, each potentially holding unique information about the system's health.

Clustering algorithms, like K-means, hierarchical clustering, or density-based clustering, are employed to group data points based on distance or similarity metrics. The choice of algorithm depends on the specific dataset and desired outcome. Once the clusters are formed, each segment can be analyzed independently, allowing for more targeted signal processing and the extraction of meaningful information related to faults or other system anomalies. Clustering-based segmentation, therefore, provides a valuable tool for isolating relevant frequency bands, enhancing signal quality, and ultimately improving the accuracy and efficiency of condition monitoring and fault diagnosis in various applications.

2.2. Fuzzy C-Mean

Fuzzy C-mean (FCM) is a popular clustering algorithm that allows data points to belong to multiple clusters with varying degrees of membership. Unlike traditional hard clustering methods where a data point belongs solely to one cluster, FCM assigns fuzzy membership values, ranging from 0 to 1, reflecting the degree of belonging to each cluster. This flexibility is particularly valuable when dealing with complex data where boundaries between clusters are not well-defined. FCM utilizes an iterative process to optimize the cluster centers and membership values, minimizing an objective function that balances the distances between data points and cluster centers while considering the degree of membership. This approach leads to more robust and informative cluster assignments, particularly in noisy or overlapping data situations, making it a valuable tool for data analysis and pattern recognition. The main target of the FCM algorithm is to minimize an objective function. The objective function is the weighted sum of squared errors within groups and is defined as follows:

$$J(U) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} d_{ij}^{2} \quad (1)$$

$$J(U,r) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} \left\| x_i - r_j \right\|^2 \quad (2)$$

where n is the number of data points, m is the number of clusters, m is any real number greater than or equal to 1, x_i is the ith data point in a multi-dimensional space, r_j is the jth cluster center, $d_{ij} = \|x_i - r_j\|$ is the distance or dissimilarity between data point x_i and cluster center r_j, and u_{ij} is the membership degree of data object x_i in cluster c_j $(0 \leq u_{ij} \leq 1)$.

3. Methodology

A signal-processing scheme was developed for identification of the fault frequencies, as shown in Figure 1. A raw signal was converted into frequency domain using envelope spectrum and decomposed into segments using fuzzy C-mean (FCM) segmentation. Each segment of the signal was converted into a time domain signal. Selection of the frequency band of interest was based on the two criterions: the first is the highest L-kurtosis (maximum impulsiveness in the signal) of the segment and the second is the lowest entropy (low degree of randomness or high cyclo-stationarity in the signal) of the segment among all the segments. Using this selection of frequency band of interest, a band pass filter was developed to filter the raw signal. Then, the envelope spectrum of the signal was extracted to identify the various frequencies of interest.

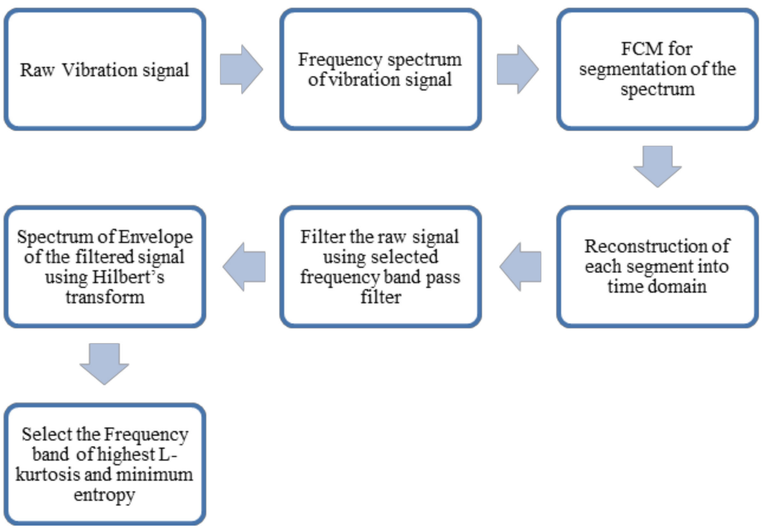


Figure 1. Proposed signal-processing scheme.

4. Results and Discussion

The proposed scheme was implemented in the simulated signal with a fault frequency of 16 Hz. A sampling rate of 20 kHz was considered and a white noise of signal-to-noise ratio (SNR) equal to -5 was added to the simulated signal as shown in Figure 2a. The envelope spectrum of the simulated signal is shown in Figure 2b, where the fault frequency of 16 Hz is not clearly visible.

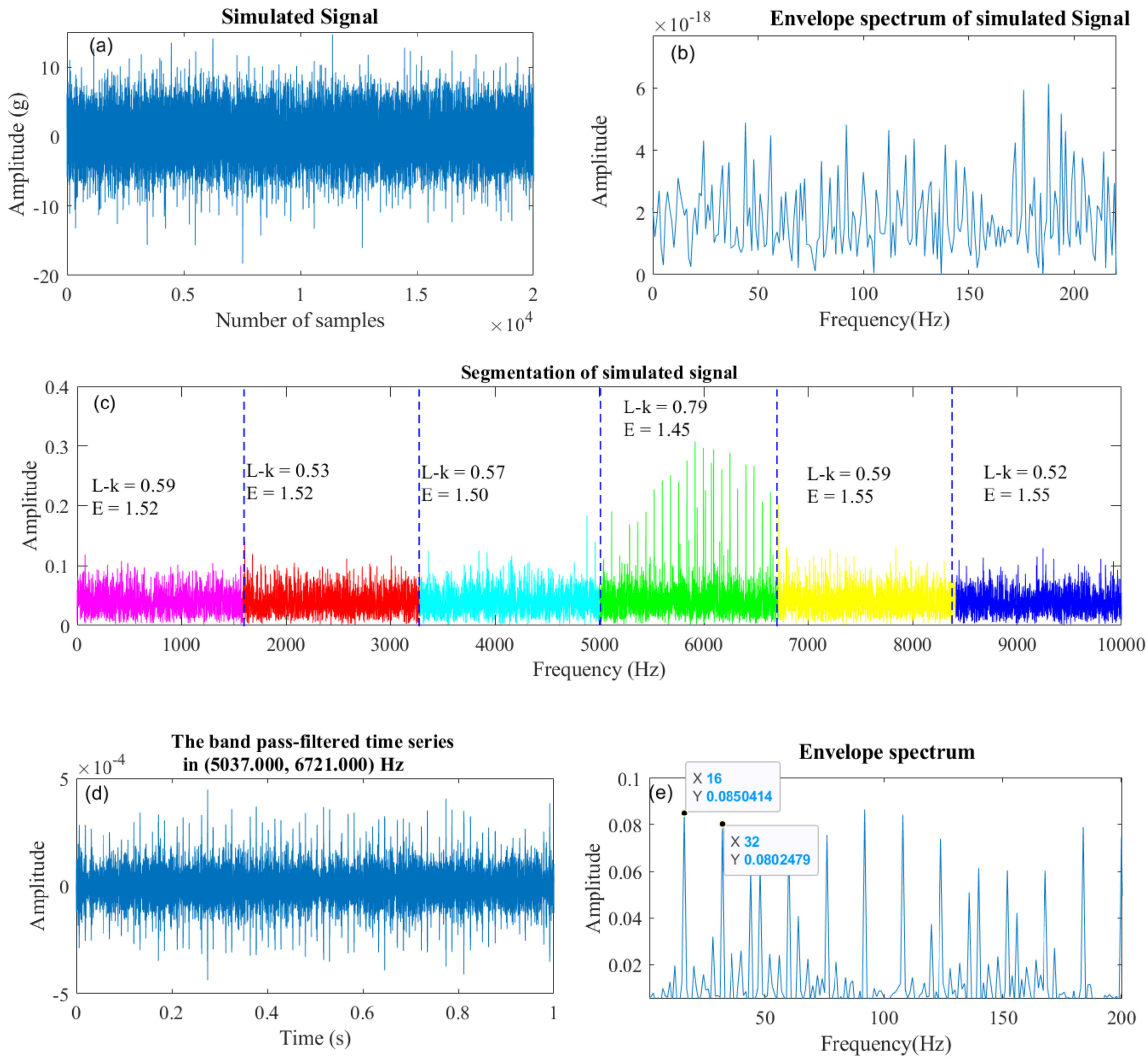


Figure 2. (**a**) Simulated signal (considered as raw signal). (**b**) Envelope spectrum of simulated signal. (**c**) Segmentation of the simulated signal. (**d**) Reconstructed signal using band-pass filter. (**e**) Envelope spectrum of the reconstructed signal.

The envelope spectrum of the signal was divided into six segments using FCM segmentation, as shown in Figure 2c. The frequency band and their central frequencies are automatically estimated using fuzzy C-mean clustering. Each segment was reconstructed as a time domain signal using inverse FFT and L-kurtosis, and the entropy of each reconstructed signal was extracted. The informative frequency band among the different segments was selected based on a combination of maximum L-kurtosis and minimum entropy of the segment. This informative frequency band was further utilized to design a bandpass filter. The original signal was reconstructed using this bandpass filter, as shown in Figure 2d. Hilbert's transform-based envelope spectrum is applied to demodulate the filtered signals for extraction of the fault frequency, as shown in Figure 2e, where a fault frequency of 16 Hz and its harmonics are clearly visible.

The effectiveness of the proposed scheme is further validated through its application to an experimental dataset obtained from a worm and wheel gearbox test rig (as depicted in Figure 3). The experimental setup utilizes a single-stage worm and wheel gearbox with a 20:1 gear ratio. The wheel, featuring 20 teeth and a pitch circle diameter of 65 mm, is driven by a 50 Hz, 1 hp DC motor (WAFRO) operating at 2880 rpm. Flexible coupling ensures the smooth transmission of power between the motor shaft and the gearbox. More details of the worm gearbox are tabulated in Table 1.

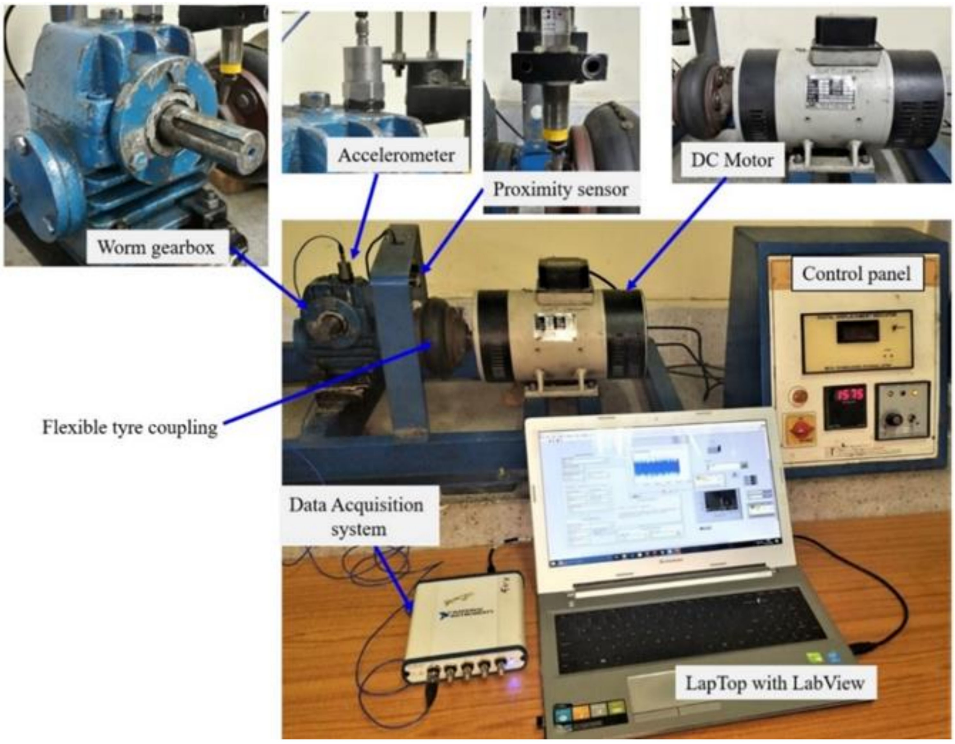


Figure 3. A typical photograph of the worm gearbox test rig.

Table 1. Details of worm gear test rig.

S. No.	Components	Specifications
1.	Motor	DCmotor; power: 1 hp.; maximum speed: 2880 rpm.
2.	Worm gearbox	Single stage; worm and wheel; gear ratio 20:1.

To capture vibration signals, a piezoelectric uniaxial PCB accelerometer with a sensitivity of 100 mV/g is strategically placed on the bearing seat of the wheel shaft. This accelerometer operates based on internal excitation, ensuring accurate measurement. A seeded pitting tooth fault, as illustrated in Figure 4, is deliberately introduced on a tooth of the wheel, simulating a common failure scenario.

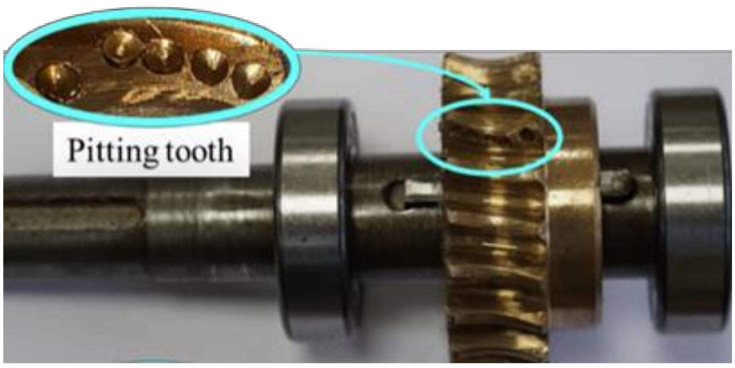


Figure 4. Gear wheel with pitting tooth seeded defect.

The experiment focuses on acquiring data under three distinct speeds: 1600 rpm, 1800 rpm, and 2000 rpm, all sampled at a rate of 20 kHz. The proposed scheme is then employed to process this acquired data. Figure 5a displays the raw signal obtained, while Figure 5b presents its corresponding envelope spectrum.

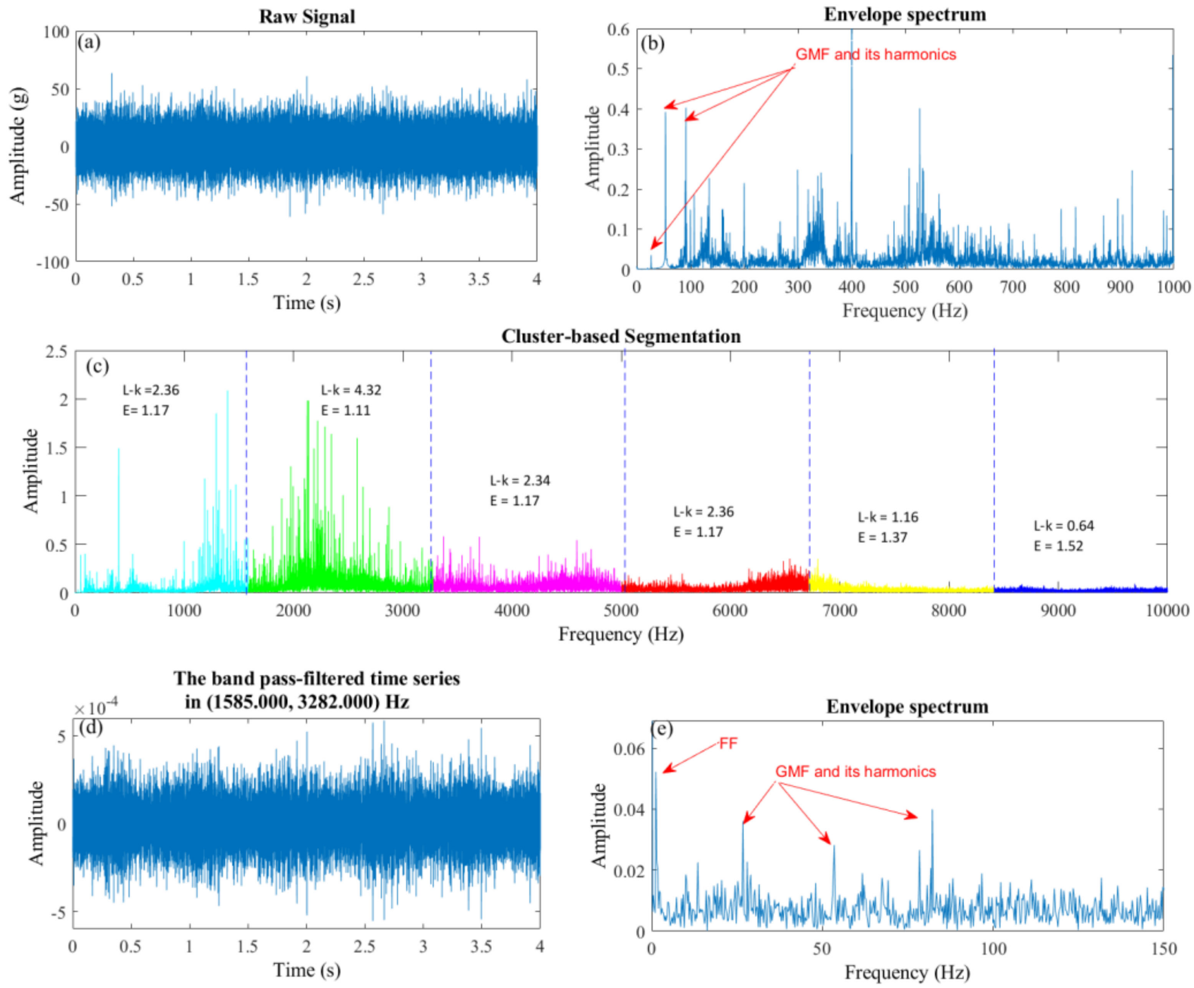


Figure 5. Processing of the signal of worm gearbox. (**a**) Raw signal. (**b**) Envelope spectrum of raw signal. (**c**) Segmentation of the signal. (**d**) Reconstructed signal using band pass filter. (**e**) Envelope spectrum of the reconstructed signal.

An analysis of the envelope spectrum reveals prominent frequency components associated with the Gear Meshing Frequency (GMF) at 26 Hz and its harmonics. This observation highlights the presence of regular tooth engagement within the gearbox. However, a crucial finding emerges: no identifiable fault signature related to the seeded pitting tooth fault is present in the spectrum. This absence of a discernible fault signature underscores the inherent challenges of detecting such faults in the presence of significant noise and complex gear meshing dynamics.

While the proposed scheme effectively captures the overall gear meshing behavior, its ability to accurately detect and characterize the specific pitting fault remains limited in this experimental scenario. The absence of a clear fault signature suggests the need for further refinement of the scheme or exploration of alternative techniques for effective fault identification in such complex systems. The proposed method is then employed to analyze the measured signal. From Figure 5c, the maximum L-kurtosis and minimum entropy of the frequency domain are 4.32 and 1.11, respectively, for the second segment among all the segments. As a result, this band is used as the passband of the filter with a frequency ranging from 1585 Hz to 3282 Hz, as shown in Figure 5d.

Applying the Hilbert transform to the filtered signal yields the frequency spectrum of the envelope signal plotted in Figure 5e, where the fault frequency (FF) of 1.25 Hz is clearly observable.

The robustness of the proposed scheme is further demonstrated through its application to a real-world scenario involving a three-stage helical gearbox. This gearbox features reduction ratios of 7, 3.2, and 3.1, resulting in a complex transmission system with multiple gear meshing frequencies. The schematic diagram of this gearbox is presented in Figure 6, whereas the detailing of the helical gearbox is tabulated in Table 2.

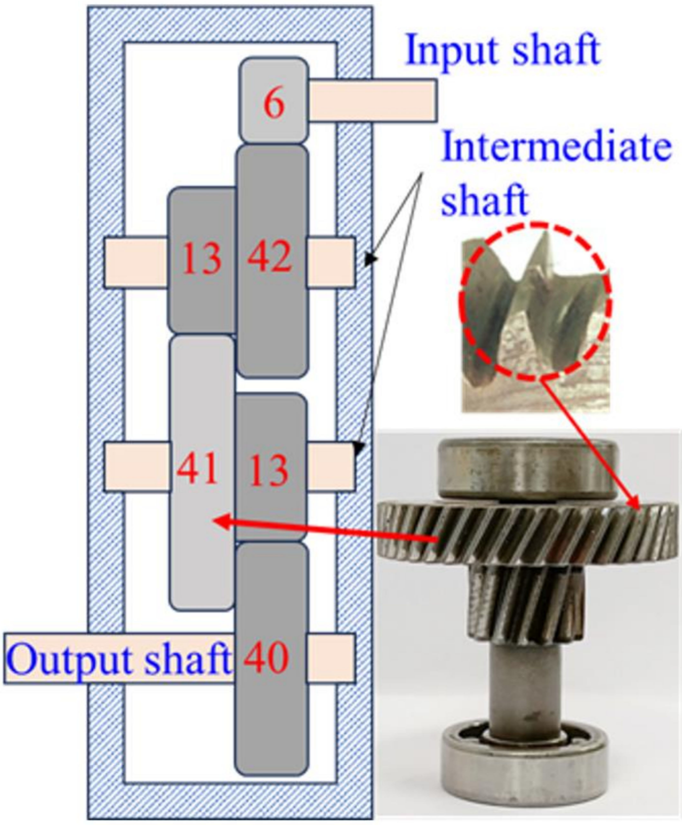


Figure 6. A schematic of helical gearbox and a photograph of seeded wear tooth defect.

Table 2. Detail of helical gear test rig.

S. No.	Components	Specifications
1.	Motor	DCmotor; power: 1 hp.; maximum speed: 2880 rpm.
2.	Helical gearbox	Three stages; helical gears; reduction ratio 7:3.2:3.1.

To introduce a controlled fault condition, a wear tooth defect is deliberately seeded on the gear of the intermediate shaft, which possesses 41 teeth (as shown in Figure 6). Vibration data are acquired at a motor speed of 2100 rpm (35 Hz), providing a representative operating condition for analysis.

The three stages of the gearbox lead to distinct gear meshing frequencies (GMFs) corresponding to the rotational frequencies of each shaft. These frequencies are calculated as 210 Hz (input shaft), 65 Hz (intermediate shaft), and 20 Hz (output shaft), based on the motor speed of 35 Hz and the gearbox reduction ratios.

The acquired vibration signal from the helical gearbox is displayed in Figure 7a, while its envelope spectrum is presented in Figure 7b. Initial analysis of the envelope spectrum reveals the presence of the characteristic frequency of the motor shaft (35 Hz, the rotational frequency of the input shaft) and its harmonics. However, the expected gear meshing frequencies (210 Hz, 65 Hz, and 20 Hz) and the fault frequency related to the wear tooth defect are conspicuously absent from the spectrum. This initial observation highlights the challenge of detecting subtle fault signatures amidst the complex vibration signals generated by the gearbox.

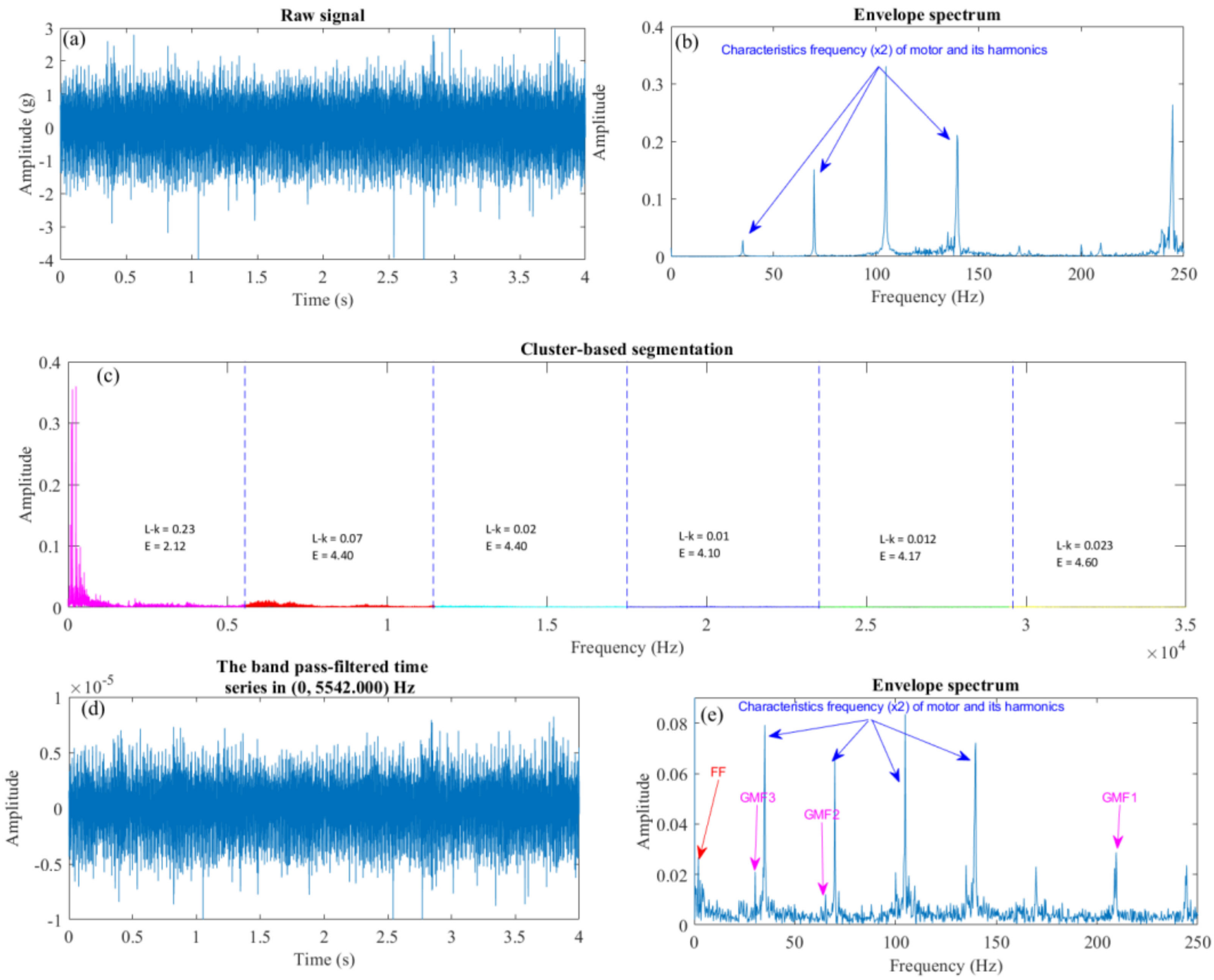


Figure 7. Processing of signal of helical gearbox. (**a**) Raw signal. (**b**) Envelope spectrum of raw signal. (**c**) Segmentation of the signal. (**d**) Reconstructed signal using band pass filter. (**e**) Envelope spectrum of the reconstructed signal.

To overcome this challenge, a segmentation strategy is implemented in the acquired signal. Segment one of the signal exhibits the highest L-kurtosis value and the lowest entropy, indicating potential significance in terms of fault information (as illustrated in Figure 7c). This segment, spanning a frequency range of 0 Hz to 5542 Hz, is selected as a passband filter for filtering the raw vibration signal (shown in Figure 7d).

The filtered signal is then subjected to the Hilbert transform, and the envelope spectrum of the resulting signal reveals key frequencies (Figure 7e). Crucially, the spectrum now exhibits not only the previously observed characteristic frequency of the motor shaft and its harmonics but also the three gear meshing frequencies (GMF1, GMF2, and GMF3) corresponding to the three stages of the gearbox. Most importantly, the fault frequency of 1.6 Hz associated with the seeded wear tooth defect is now clearly visible in the spectrum.

This successful identification of the fault frequency through the proposed scheme demonstrates its efficacy in extracting valuable fault information from complex gearbox vibration data. The combination of segmentation based on statistical features (L-kurtosis and entropy), passband filtering, and the Hilbert transform envelope analysis proves to be highly effective in isolating and highlighting subtle fault signatures that would otherwise be masked by noise and complex gear meshing dynamics.

5. Conclusions

This paper presents a novel signal-processing scheme for extracting fault-related information from vibration signals, leveraging cluster-based segmentation as a key component. This method utilizes the inherent capability of clustering algorithms to automatically divide the signal into a predetermined number of distinct segments.

The identification of the segment containing the most valuable fault-related information is based on two crucial criteria: maximum L-kurtosis and minimum entropy. L-kurtosis, a measure of impulsiveness, highlights segments characterized by sharp, transient events, often indicative of incipient faults. Conversely, minimum entropy identifies segments exhibiting a high degree of cyclo-stationarity, suggesting a strong periodic pattern related to the machine's operating conditions. By focusing on the segment exhibiting both maximum L-kurtosis and minimum entropy, the proposed scheme pinpoints the frequency range containing the most relevant fault-related information.

To further refine the signal and extract this crucial information, a passband filter is specifically designed for the identified frequency range. This filtering process effectively isolates the desired information, enriching the signal with impulsive and cyclo-stationary components directly related to the fault.

The proposed scheme has been successfully validated through fault detection experiments on various gearbox types. Its ability to automatically segment signals, identify the most informative frequency range, and enhance the signal with fault-related components demonstrates its robustness and versatility as a tool for condition monitoring and fault diagnosis in mechanical systems.

Author Contributions: S.K. (Surinder Kumar): Data curation, Writing—original draft; S.C.: Data curation, Writing—original draft; G.V.: Data curation, Software, Writing—original draft, Methodology; S.K. (Sunil Kumar): Data curation, Software, Writing—original draft; R.K.: Writing—review and editing, Supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data can be made available upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Peng, C.; Gao, H.; Liu, X.; Liu, B. A visual vibration characterization method for intelligent fault diagnosis of rotating machinery. *Mech. Syst. Signal Process.* **2023**, *192*, 110229. [CrossRef]
2. He, G.; Li, J.; Ding, K.; Zhang, Z. Feature extraction of gear and bearing compound faults based on vibration signal sparse decomposition. *Appl. Acoust.* **2022**, *189*, 108604. [CrossRef]
3. Hou, Y.; Zhou, C.; Tian, C.; Wang, D.; He, W.; Huang, W.; Wu, P.; Wu, D. Acoustic feature enhancement in rolling bearing fault diagnosis using sparsity-oriented multipoint optimal minimum entropy deconvolution adjusted method. *Appl. Acoust.* **2022**, *201*, 109105. [CrossRef]
4. Zhang, T.; Xu, F.; Jia, M. A centrifugal fan blade damage identification method based on the multi-level fusion of vibro-acoustic signals and CNN. *Measurement* **2022**, *199*, 111475. [CrossRef]
5. Li, Z.; Xiao, J.; Ding, X.; Wang, L.; Yang, Y.; Zhang, W.; Du, M.; Shao, Y. A new raw signal fusion method using reweighted VMD for early crack fault diagnosis at spline tooth of clutch friction disc. *Measurement* **2023**, *220*, 113414. [CrossRef]
6. Kumar, R.; Kumar, P.; Vashishtha, G.; Chauhan, S.; Zimroz, R.; Kumar, S.; Kumar, R.; Gupta, M.K.; Ross, N.S. Fault Identification of Direct-Shift Gearbox Using Variational Mode Decomposition and Convolutional Neural Network. *Machines* **2024**, *12*, 428. [CrossRef]
7. Cioch, W.; Knapik, O.; Leśkow, J. Finding a frequency signature for a cyclostationary signal with applications to wheel bearing diagnostics. *Mech. Syst. Signal Process.* **2013**, *38*, 55–64. [CrossRef]
8. Obuchowski, J.; Wyłomańska, A.; Zimroz, R. Selection of informative frequency band in local damage detection in rotating machinery. *Mech. Syst. Signal Process.* **2014**, *48*, 138–152. [CrossRef]
9. Smith, W.A.; Borghesani, P.; Ni, Q.; Wang, K.; Peng, Z. Optimal demodulation-band selection for envelope-based diagnostics: A comparative study of traditional and novel tools. *Mech. Syst. Signal Process.* **2019**, *134*, 106303. [CrossRef]

10. Lei, Y.; Lin, J.; He, Z.; Zuo, M.J. A review on empirical mode decomposition in fault diagnosis of rotating machinery. *Mech. Syst. Signal Process.* **2013**, *35*, 108–126. [CrossRef]
11. Espinoza-Sepulveda, N.; Sinha, J. Two-step vibration-based machine learning model for the fault detection and diagnosis in rotating machine and its blind application. *Struct. Health Monit.* **2024**, 14759217241249055. [CrossRef]
12. Almutairi, K.; Sinha, J.K.; Wen, H. Fault Detection of Rotating Machines Using poly-Coherent Composite Spectrum of Measured Vibration Responses with Machine Learning. *Machines* **2024**, *12*, 573. [CrossRef]
13. Bendjama, H. Feature extraction based on vibration signal decomposition for fault diagnosis of rolling bearings. *Int. J. Adv. Manuf. Technol.* **2024**, *130*, 821–836. [CrossRef]
14. Tahmasbi, D.; Shirali, H.; Sajad Mousavi Nejad Souq, S.; Eslampanah, M. Diagnosis and root cause analysis of bearing failure using vibration analysis techniques. *Eng. Fail. Anal.* **2024**, *158*, 107954. [CrossRef]
15. Antoni, J. The infogram: Entropic evidence of the signature of repetitive transients. *Mech. Syst. Signal Process.* **2016**, *74*, 73–94. [CrossRef]
16. Wang, Y.; Xiang, J.; Markert, R.; Liang, M. Spectral kurtosis for fault detection, diagnosis and prognostics of rotating machines: A review with applications. *Mech. Syst. Signal Process.* **2016**, *66–67*, 679–698. [CrossRef]
17. Combet, F.; Gelman, L. Optimal filtering of gear signals for early damage detection based on the spectral kurtosis. *Mech. Syst. Signal Process.* **2009**, *23*, 652–668. [CrossRef]
18. Wang, Y.; Liang, M. Identification of multiple transient faults based on the adaptive spectral kurtosis method. *J. Sound Vib.* **2012**, *331*, 470–486. [CrossRef]
19. Wang, T.; Chu, F.; Han, Q.; Kong, Y. Compound faults detection in gearbox via meshing resonance and spectral kurtosis methods. *J. Sound Vib.* **2017**, *392*, 367–381. [CrossRef]
20. Borghesani, P.; Pennacchi, P.; Randall, R.B.; Sawalhi, N.; Ricci, R. Application of cepstrum pre-whitening for the diagnosis of bearing faults under variable speed conditions. *Mech. Syst. Signal Process.* **2013**, *36*, 370–384. [CrossRef]
21. Sacerdoti, D.; Strozzi, M.; Secchi, C. A Comparison of Signal Analysis Techniques for the Diagnostics of the IMS Rolling Element Bearing Dataset. *Appl. Sci.* **2023**, *13*, 5977. [CrossRef]
22. Antoni, J.; Randall, R.B. The spectral kurtosis: Application to the vibratory surveillance and diagnostics of rotating machines. *Mech. Syst. Signal Process.* **2006**, *20*, 308–331. [CrossRef]
23. Berrouche, Y.; Vashishtha, G.; Chauhan, S.; Zimroz, R. Local damage detection in rolling element bearings based on a single ensemble empirical mode decomposition. *Knowl.-Based Syst.* **2024**, *301*, 112265. [CrossRef]
24. Wang, H.; Du, W. Early weak fault diagnosis of rolling element bearing based on resonance sparse decomposition and multi-objective information frequency band selection method. *J. Vib. Control* **2022**, *28*, 2762–2776. [CrossRef]
25. Hebda-Sobkowicz, J.; Zimroz, R.; Wyłomańska, A. Selection of the Informative Frequency Band in a Bearing Fault Diagnosis in the Presence of Non-Gaussian Noise—Comparison of Recently Developed Methods. *Appl. Sci.* **2020**, *10*, 2657. [CrossRef]
26. Schmidt, S.; Heyns, P.S.; Gryllias, K.C. An informative frequency band identification framework for gearbox fault diagnosis under time-varying operating conditions. *Mech. Syst. Signal Process.* **2021**, *158*, 107771. [CrossRef]
27. Zhao, H.; Jiang, X.; Wang, B.; Cheng, X. Bearing fault feature extraction method: Stochastic resonance-based negative entropy of square envelope spectrum. *Meas. Sci. Technol.* **2024**, *35*, 045102. [CrossRef]
28. Wu, K.; Tong, W.; Huang, B.; Wu, D. Combined Weighted Envelope Spectrum: An enhanced demodulation framework for extracting characteristic frequency of rotating machinery. *Mech. Syst. Signal Process.* **2024**, *209*, 111083. [CrossRef]
29. Kumar, S.; Kumar, R. L-Moments Ratio-Based Condition Indicators for Diagnosis of Fault in a Worm Gearbox. *J. Vib. Eng. Technol.* **2023**, *11*, 4131–4149. [CrossRef]

applied sciences

MDPI

Article

Machine Learning Use Cases in the Frequency Symbolic Method of Linear Periodically Time-Variable Circuits Analysis

Yuriy Shapovalov [1], Spartak Mankovskyy [1], Dariya Bachyk [1], Anna Piwowar [2], Łukasz Chruszczyk [3] and Damian Grzechca [3,*]

1 Institute of Telecommunications, Radioelectronics and Electronic Engineering, Lviv Polytechnic National University, 79013 Lviv, Ukraine; yurii.i.shapovalov@lpnu.ua (Y.S.); spartak.v.mankovskyi@lpnu.ua (S.M.); dariia.r.bachyk@lpnu.ua (D.B.)
2 Faculty of Electrical Engineering, Silesian University of Technology, 44-100 Gliwice, Poland; anna.piwowar@polsl.pl
3 Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, 44-100 Gliwice, Poland; lukasz.chruszczyk@polsl.pl
* Correspondence: dgrzechca@polsl.pl or damian.grzechca@polsl.pl

Abstract: This manuscript presents an analysis of machine learning (ML) usage in the Frequency Symbolic Method (FSM) to enhance the diagnosis of faults in parametric circuit analysis and optimization, with a particular focus on Linear Periodically Time-Variable (LPTV) systems. We put forth a few ML-based approaches for fault diagnosis (including anomaly detection), invisible feature detection, and the prediction of FSM output. These methodologies concentrate on identifying and diagnosing faults by evaluating particular ML techniques, extracting pertinent features, and determining the desired diagnostic outputs. The use cases of ML application considered in this paper demonstrate that machine learning can enhance fault detection and diagnosis, reduce human errors and identify previously unnoticed anomalies within the FSM framework. ML has never been used in FSM before, so the key aim of this paper is to consider possible use cases of AI application in FSM. Additionally, feature extraction, required as an input stage for the ML model, is proposed based on FSM peculiarities. This work can be considered a study of ML application in FSM.

Keywords: machine learning; frequency symbolic method; linear periodically time-variable circuits

1. Introduction

A linear time-varying (LTV) system is a non-stationary deterministic system defined by parameters that vary over time. These systems, also known as parametric systems or LTV systems, feature time-dependent coefficients, referred to as parametric functions.

A multitude of theoretical studies [1,2] and practical applications have concentrated on these systems, as they are regarded as suitable models for real-world scenarios. Systems that have been modeled by time-varying circuits are extensively utilized in modern electrical circuits [3], communication systems [4], and automatic control [5]. In particular, parametric systems are applied to the analysis of analog and digital signal processing systems [2], especially as models of sampling systems and modulation [6]. Moreover, parametric systems are used in current compensation in power grids [7], wideband low-noise amplifiers [8], filter systems and noise reduction [9,10] generators [11], and medical devices [12]. Furthermore, the modeling of circuits directly in the time domain serves as a foundation for functional testing and fault diagnosis, where time-domain excitation and/or circuit response constitute key data [13]. The implementation of LTV systems has a number of benefits, the most important of which are the improvement of system dynamics and the reduction of transients. Consequently, the modeling of parametric systems remains a highly relevant task.

It is a well-known fact in circuit theory that there are two basic ways to analyze linear and time-invariant (LTI) systems. The first method is to analyze systems in the time domain, either by solving differential equations describing the systems or by using convolution techniques. The second group of methods, called frequency analysis, is based on the use of Laplace or Fourier transforms.

LTV systems represent a direct generalization of classical linear and time-invariant (LTI) systems. Accordingly, an analysis of their dynamic and frequency properties can be performed in both the time and frequency domains. Furthermore, the concepts of impulse response, frequency response, and transmission, familiar from the theory of stationary systems, can be generalized to non-stationary systems. In the time domain, parametric systems can be described using two different methods. The first method involves representing systems in the time domain using a parametric linear differential Equation (1) [7,9,14]:

$$a_n(t)y^{(n)}(t) + a_{n-1}(t)y^{(n-1)}(\mathrm{t}) + \ldots + a_1(t)y' + a_0(t)y(t) = x(t), \tag{1}$$

where $x(t)$, $y(t)$, are the input and output signals, t is the independent variable (time), and $a_i(t)$ are the time-varying coefficients of the differential equation, which are called parametric functions. Except for a few special cases, there are no well-established analytical methods for solving this equation. The authors' current research focuses on a class of parametric systems with periodically varying coefficients known as LPTV, or Linear Time Periodic Systems [15].

The alternative method of describing an LTV system in the time domain, as opposed to the parametric differential equation approach, is a parametric convolution using the impulse response $h(t,\tau)$ as the kernel:

$$y(t) = \int_0^t h(t,\tau)x(\tau)d\tau. \tag{2}$$

It is important to note that the impulse response of LTV systems differs from that of time-invariant systems in that it depends not only on the time t, but also on the instant τ at which an excitation is applied to the input of the circuit.

In the frequency domain, two approaches may be used to describe a parametric circuit. The first is identical to that of LTI systems and is defined as the ratio of the complex output signal $y(t)$ to the complex input signal, which is the monoharmonic function $\mathrm{e}^{j\omega t}$. The second method requires knowledge of the system's impulse response function and is expressed by the following equations:

$$W(j\omega,t) = \mathcal{F}\{h(t,\tau)\} = \int_0^t h(t,\tau)\mathrm{e}^{-j\omega\tau}\mathrm{d}\tau \tag{3}$$

$$W(s,t) = \mathcal{L}\{h(t,\tau)\} = \int_0^t h(t,\tau)\mathrm{e}^{-s\tau}\mathrm{d}\tau \tag{4}$$

From a mathematical perspective, Formulas (3) and (4) define the classical Fourier and Laplace transforms of the impulse response with respect to the variable τ.

The main difference between the time-frequency analysis of LTV systems and that of stationary systems is the inability to proceed directly from differential equations to transfer functions. In this conventional approach, the determination of the impulse response function or the system response to an arbitrary input is of paramount importance in determining the frequency characteristics of LTV systems. The main challenge is to find fundamental solutions to the homogeneous parametric differential equations satisfying Equation (1).

Analytical solutions to differential equations are available only in specific cases, which heavily depend on the equation's order and the form of the parametric functions.

The fundamental solutions of these equations, even for first or second order, are highly intricate and can be represented by special functions from mathematical physics, such as Bessel functions of the first and second kind with non-integral orders, as well as confluent hypergeometric functions [16]. Assuming periodic variation of the parameters of LPTV systems (with the constraint that they are at most second-order systems), well-known equations from mathematical physics, such as the Mathieu, Meissner, or Hill equations, can be used. However, the key issue remains the stability of these solutions. Depending on the choice of the coefficient variation over time, the systems can show instabilities. Furthermore, in most cases, the solutions to these equations are not available in closed form (analytically), requiring support from numerical computations in the analysis. While the exact analytical determination of this function is feasible in a limited number of cases, even then, the function is represented by special functions, rendering further analysis exceedingly challenging. The authors emphasize that while the theory of differential equations provides fundamental solutions (in some cases) through integral equations, the determination of closed-form solutions—such as the response of the system to a non-zero external forcing or an impulse response function—is often analytically intractable. Consequently, these solutions typically require the use of analytical numerical methods, and one must often rely on approximate solutions to effectively address the problem.

Given these complexities, the Frequency Symbolic Method (FSM), introduced by the authors in their earlier works [15] and briefly summarized in Section 2, becomes a valuable tool for describing and analyzing LPTV systems in the frequency domain, as it enables the frequency description of a specific class of parametric systems with periodically varying parameters in conjunction with the L.A. Zadeh equation [1]. The FSM allows direct formulation and analysis in the frequency domain, eliminating the need to solve time-domain differential equations. This approach is particularly advantageous as it avoids the intermediate step of time-domain analysis, which can be both complex and computationally expensive for systems with periodically varying parameters (LPTV). The FSM is applied using the L.A. Zadeh matrix equation, which provides a more organized and symbolic representation of LPTV systems. This matrix-based approach decomposes the problem into several independent matrix equations, making analysis easier and allowing a more methodical approach to solving the system.

Although the authors are well-versed in the frequency analysis methodology employed using this approach, there is a need for appropriate methods and computational algorithms to minimize the time required to determine the frequency response coefficients of the system. One can observe that LPTV circuits often describe circuits functioning at a variable operating point with properties that depend strictly on time-varying coefficients. Therefore, minimizing computational time is essential for the optimal selection of the variation waveform of the parametric functions. This necessity gives rise to the concept of implementing machine learning estimation, a branch of artificial intelligence (AI), in the context of the frequency-domain method of symbolic parametric circuit analysis and optimization. When implemented using specialized techniques such as d-trees, FSM offers significant improvements in both computational time and complexity management compared to standard methods such as MATLAB. This makes FSM a more practical choice for tackling large-scale problems where traditional methods may become impractical due to computational limitations.

2. About the Frequency Symbolic Method

The frequency symbolic method (FSM) is based on the following fundamental concepts [15]:

1. Consider the LPTV circuit where:
   - There are one or more parametric elements, whose parameters change periodically over time with the same period T (in the simplest version),

- There is one input with an input signal $x(t)$ and one output with a response $y(t)$.

2. Such an LPTV circuit in the time domain can be described by the symbolic system of linear differential equations (SSLDE).
3. By eliminating internal variables in the SSLDR using one of the known methods, one can obtain a differential Equation (1) describing the circuit.

$$a_n(t)y^{(n)} + a_{n-1}(t)y^{(n-1)} + \ldots + a_0(t)y = b_m(t)x^{(m)} + b_{m-1}(t)x^{(m-1)} + \ldots + b_0(t)x, \tag{5}$$

where $y(t)$ is the output (sought) and $x(t)$ is the input (set) variable, t is the independent variable (time), and $a_i(t)$, $b_j(t)$ are parametric functions of time t.

4. For an LPTV circuit described by the differential Equation (5), using the equation of L.A. Zadeh [1], we determine the differential equation describing this circuit in the frequency domain:

$$\frac{1}{n!}\frac{d^nA(s,t)}{ds^n}\frac{d^nW(s,t)}{dt^n} + \ldots + \frac{dA(s,t)}{ds}\frac{dW(s,t)}{dt} + A(s,t)W(s,t) = B(s,t), \tag{6}$$

where:

$$W(s,t) = \frac{Y(s,t)}{X(s,t)}, \tag{7}$$

is the transfer function of the LPTV circuit, and:

$$A(s,t) = a_n(t)s^n + \ldots + a_1(t)s + a_0(t), \tag{8}$$

$$B(s,t) = b_m(t)s^m + \ldots + b_1(t)s + b_0(t), \tag{9}$$

are the corresponding time-dependent periodic functions of time t with period T coefficients of Equation (1), whereas $Y(s,t)$ and $X(s,t)$ are the images of the output and input variables in the frequency domain, respectively, and s is a complex variable.

5. In general, Equation (6) does not have an exact analytical solution, so one must find an approximate solution that accounts for both the known properties of the desired transfer function $W(s,t)$ and the specifics of the algorithm used for its determination:
   (a) The approximation $\hat{W}(s,t)$ of the transfer function $W(s,t)$ must achieve a predefined accuracy level of choice,
   (b) Given that $\hat{W}(s,t)$ needs to undergo n-times differentiation when inserted into (6), it should be represented by a function that allows straightforward differentiation (i.e., intricate fractions, etc.). Consequently, for solving (6), one can suggest an approximation of the transfer function by a trigonometric Fourier series:

$$W(s,t) = W_0(s) + \sum\nolimits_{i=1}^{k}\left[W_{ci}(s)\cos(\Omega t) + W_{si}(s)\sin(\Omega t)\right],\ \Omega = \frac{2\pi}{T}, \tag{10}$$

or in a complex form:

$$\hat{W}(s,t) = W_0(s) + \sum\nolimits_{i=1}^{k}\left[W_{-i}(s)\exp(i\Omega t) + W_{+i}(s)\exp(i\Omega t)\right]. \tag{11}$$

The frequency symbolic method of solution to (6) finding is as follows:

Step 1. One of the expressions (11) or (13), for instance, Equation (11) is differentiated n times with respect to the variable t. The resulting derivatives, along with the original expression, are then substituted into (6).

Step 2. By transferring the right-hand side of expression (6) to the left side, one can obtain the following algebraic expression:

$$\delta(W_0, W_{c1}, W_{s1}, W_{c2}, W_{s2}, \ldots, W_{ck}, W_{sk}) = 0. \tag{12}$$

The functional $\delta(\cdot)$ described by (13) is periodic with period T and contains $(2k + 1)$ unknowns W_0, W_{c1}, W_{s1}, W_{c2}, W_{s2}, . . ., W_{ck}, W_{sk}, which need to be determined.

Step 3. As the functional $\delta(\cdot)$ is periodic, we decompose it into a Fourier series with period T. According to (13), we equate k harmonics and the constant component of this series to zero. This results in a system of $(2k + 1)$ linear algebraic equations, which form a symbolic system of linear algebraic equations (SSLAE) of the $(2k + 1)$ order with $(2k + 1)$ unknowns.

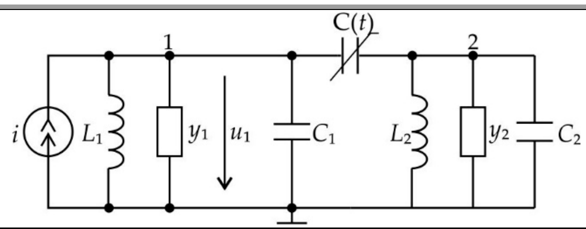
Step 4. The solution of the resulting SSLAE determines the desired unknowns of expression (8) and approximates (11) or (12).

The features of the frequency symbolic method of analyzing LPTV circuits described above are as follows:

1. Regarding the choice of the number of k harmonics (step 3), it should be noted that, as a rule, the first k harmonics are typically chosen. However, there are algorithms that allow solutions to be obtained even when the number of equations is greater than the number of unknowns. In our opinion, the choice of harmonics and the number of equations in each case should be left to the specialist who designs and studies the given LPTV circuit.
2. The solution obtained in step 4 of the SSLAE (in the form of $\mathbf{M} \times \mathbf{W} = \mathbf{P}$ matrix) has its own peculiarities, as some or all elements of the matrix **M** and vector **P** are symbolic. Therefore, symbolic methods should be used to solve the SSLAE.
3. The approximation selection of Equations (11) or (12) is not fundamentally important. However, empirical experience of using the approximation described by Equation (12), has shown that the matrix **M** is sparser than with approximation (11) of the same order. This is significant in the symbolic solution of the SLAP. Therefore, approximation (12) is more beneficial, in our opinion. The values of $W_0(s)$, $W_{ci}(s)$, $W_{si}(s)$ or $W_0(s)$, $W_{+i}(s)$, $W_{-i}(s)$ in this case are fractional rational expressions, where the denominators are identical and correspond to the determinant of the **M** matrix. The numerators are the determinants of the modified matrices **M**, where the corresponding column is replaced by the vector **P**.

   The peculiarity of steps 1–4 is that several (or even all) of the parameters of the analyzed circuit, including s, Ω, and other parametric element coefficients, are specified symbolically. Therefore, the differentiation of the approximating function, substitution of it and its derivatives into the equation (6), determination of k harmonics with $(2k+1)$-fold integration of the expression (13) products into the corresponding orthogonal functions, and solving the SSLAE in symbolically is very cumbersome. In this context, the sophisticated symbolic computation capabilities of contemporary CAD software (MATLAB 2014a) proved invaluable, enabling the calculations described in this manuscript to be carried out with a high degree of feasibility [15].
4. The symbolic solution of the SLAE is performed in the system user-defined functions MAOPCs [15] using standard functions of the MATLAB environment.

Figure 1 shows the most commonly used steps of the FSM procedure. There are other possible steps depending on the input data for FSM, but they are not detailed in this manuscript. Figure 1 also presents the division of feature sets that could be extracted, which are described in Section 3.



Electrical circuit (schematic)

$$\begin{bmatrix} C''(t)+\frac{1}{L_1} & y_1+2C'(t) & C_1+C(t) & -C''(t) & -2C'(t) & -C(t) \\ -C''(t) & -2C'(t) & -C(t) & C''(t)+\frac{1}{L_2} & y_2+2C'(t) & C_2+C(t) \end{bmatrix} \begin{bmatrix} u_1 \\ u_1' \\ u_1'' \\ u_2 \\ u_2' \\ u_2'' \end{bmatrix} =$$

$$= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} i \\ i' \end{bmatrix}$$

System of differential equations ⇨ Features set I1

$$a_n(t)y^{(n)} + a_{n-1}(t)y^{(n-1)} + \cdots + a_1(t)y =$$
$$= b_m(t)x^{(m)} + b_{m-1}(t)x^{(m-1)} + \cdots + b_1(t)x$$

Transfer equation ⇨ Features set I2

$$\begin{bmatrix} F_{-k,-k} & \cdots & F_{-k,-1} & F_{-k,0} & F_{-k,1} & \cdots & F_{-k,k} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ F_{-1,-k} & \cdots & F_{-1,-1} & F_{-1,0} & F_{-1,1} & \cdots & F_{-1,k} \\ F_{0,-k} & \cdots & F_{0,-1} & F_{0,0} & F_{0,1} & \cdots & F_{0,k} \\ F_{1,-k} & \cdots & F_{1,-1} & F_{1,0} & F_{1,0} & \cdots & F_{1,k} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ F_{k,-k} & \cdots & F_{k,-1} & F_{k,0} & F_{k,0} & \cdots & F_{k,k} \end{bmatrix}$$

System of linear equations obtained using FSM ⇨ Features set M1

$$\widehat{W}(s,t) = W_0(s) + \sum_{i=1}^{k}[W_{-i}(s)\exp(i\Omega t) + W_{+i}(s)\exp(i\Omega t)].$$

Symbolic transfer function ⇨ Features Set O1

Figure 1. Typical FSM steps and extracted feature sets.

3. Results

3.1. Consideration of Feature Extraction Required for Machine Learning Applications

This section considers the feature extraction required for machine learning (ML) applications based on the frequency symbolic method. The subsequent sections detail the specific use cases, including the inputs, outputs, and ML approaches to be employed for each case.

The basic steps in any artificial intelligence (AI) system development are as follows:

1. Data Collection.
2. Feature Extraction (Data Conversion for applicable inputs of the ML system).
3. Target Definition (what is the target of the ML system?).
4. Model Selection.
5. Model Training.
6. Model Evaluation.
7. Parameter Tuning.

The key purpose of this work is to estimate the feature extraction and target definition steps from the list above. The data collection step could be accomplished by analyzing a set of parametric circuits with different parameters and configurations or by collecting statistics on configurations applied during FSM usage by researchers. The next steps, such as selection, training, and evaluation of the model are out of scope of this work and will be considered in further phases of the research. The feature extraction step in ML system development is devoted to converting input data into a vector of features, which later become the direct inputs to the ML model. The feature data type should typically be one of two types: numerical or categorical. Numerical values can take any continuous or discrete value within a given range. Categorical values can take one of the predefined values, such as [“Yes”, “No”] or [“Low”, “Medium”, “High”]. Figure 1 shows the feature sets that can be extracted at different steps of FSM usage.

The aforementioned feature sets are delineated and designated with unique identifiers:

- I1 (Input type 1)—features extracted from the system of differential equations.
- I2 (Input type 2)—features extracted from the transfer equation.
- M1 (Method features 1)—features extracted from the System of Linear Algebraic Equations (SLAE) obtained using FSM.
- O1 (Output features 1)—features extracted from the symbolic transfer function obtained using FSM.

In consideration of the FSM steps illustrated in Figure 1, we present the selected list of features in Tables 1–4. As the ML application to the FSM progresses, we will provide more precise details regarding these features.

Table 1. Proposed list of features set I1 extracted in machine learning application to FSM.

Feature ID and Short Name	Description	Data Type	Valid Range
I1.1 Number of LC components	Number of LC components (except parametric) in the system of differential equations describing the circuit.	Numerical	Positive integer number
I1.2 Percentage of non-zero elements	Relation of non-zero elements to all elements.	Numerical	Positive integer [0, 100]
I1.3 Count of symbolic parameters	Total number of parameters in symbolic form.	Numerical	Positive integer number
I1.4 Percentage of symbolic cells to the numerical cells	Relation of cells number in the system of differential equations containing at least one symbolic variable to the total number of cells.	Numerical	Positive integer [0, 100]
I1.5 Time-varying function type	Time-varying function type as one of single harmonic, bi-harmonic, multi-harmonic, etc.	Categorical	[Harm, biharm, multiharm]

Table 2. Proposed list of feature set I2 extracted in machine learning application to FSM.

Feature ID and Short Name	Description	Data Type	Valid Range
I2.1 Order of the left part	Order of the left part of the transfer equation.	Numerical	Positive integer number
I2.2 Order of the right part	Order of the right part of the transfer equation.	Numerical	Positive integer number

Table 2. *Cont.*

Feature ID and Short Name	Description	Data Type	Valid Range
I2.3 Count of symbolic parameters	Total number of parameters in symbolic form.	Numerical	Positive integer number
I2.4 Complexity of expression	Numerical value representing the complexity of the symbolic expression. For example, a function depending on the number of multiplications, exponents, logarithmic operation, and so on.	Numerical	Positive integer number [1]

[1] Justification: It is assumed that the assigned weights in a binary expression tree are positive integer numbers, and that the resulting complexity will also be a positive integer.

Table 3. Proposed list of feature set M1 extracted in machine learning application to FSM.

Feature ID and Short Name	Description	Data Type	Valid Range
M1.1 Order of the SLAE ($2k + 1$)	Order of the SLAE obtained using FSM.	Numerical	Positive integer number
M1.2 Complexity of SLAE	Numerical value representing the complexity of SLAE, i.e., some functions depend on the average complexity of cells in SLAE. The complexity of each SLAE cell could be calculated as binary expression tree complexity.	Numerical	Positive integer number [1]

[1] Justification: It is assumed that the assigned weights in a binary expression tree are positive integer numbers, and that the resulting complexity will also be a positive integer.

Table 4. Proposed list of feature set O1 extracted in machine learning application to FSM.

Feature ID and Short Name	Description	Data Type	Valid Range
O1.1 Number of terms in the transfer function	Number of terms in the transfer function.	Numerical	Positive integer number
O1.2 Complexity of transfer function	Numerical value representing the complexity of the transfer function. For example, some functions depend on the average complexity of each term.	Numerical	Positive integer number [1]
O1.3 Maximal complexity of one term	Numerical value representing the complexity of the most complex term in the transfer function. For example, some functions depend on the number of multiplications or exponentiations of symbolic variables.	Numerical	Positive integer number [1]

[1] Justification: It is assumed that the assigned weights in a binary expression tree are positive integer numbers, and that the resulting complexity will also be a positive integer.

An example of the calculation of the complexity of a symbolic expression is shown in a separate chapter at the end of this article.

Target Definition. This step is appointed to determine data and its type that is desired as output of the ML system. The output data can also be of numerical or categorical type. An example of target values is shown in Table 5.

Table 5. Example target definitions of the ML system applied to FSM.

Target ID and Short Name	Description	Data Type	Valid Range
T1.1 Integral deviation of obtained transfer function from desired	Integral deviation of the obtained transfer function from desired. It is obtained after substitution of the numerical values (may be also optimization applied) into the symbolic transfer function and integral comparison with the target transfer function.	Numerical	Floating-point number
T1.2 Local deviation of obtained transfer function from desired	Like above but evaluated in a specified local range of time or frequency.	Numerical	Floating-point number
T1.3 Amplification possibility	Is it possible that module of transfer function causes signal amplification?	Categorical	["Yes", "No"].
T1.4 Possible non-stability	Is the non-stability of the system/circuit possible?	Categorical	["Yes", "No"]

3.2. Use Case of Parametric Circuits Synthesis

This use case is mainly based on a reinforcement learning type (Figure 2). In such a case, the ML system tries to prepare a symbolic transfer function and receives a corresponding "reward" according to the level of how this transfer function fits the desired output.

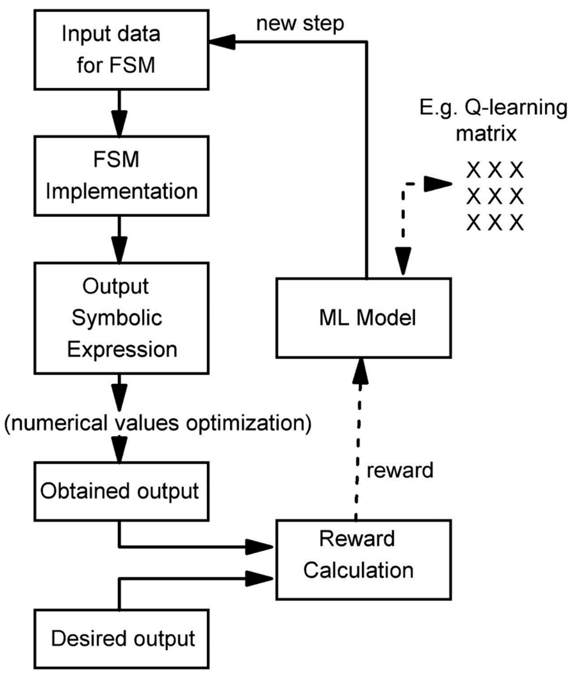


Figure 2. Use case of circuit synthesis using FSM and AI.

The "new step" in Figure 2 could involve adding or removing an existing electric component (e.g., R, L, C) to specific nodes of the electrical circuit. The steps are as follows:

- AI tries to put electronic components in a semi-random place in the schematic considering its previous experience.
- Calculate the symbolic expression of the transfer function.
- Try to optimize component values with some constraints.
- Check the best result with the target frequency characteristic and provide the corresponding "reward".

At each iteration, AI adjusts its coefficients and remembers the change in the schematic and corresponding impact. The Q-learning ML method could be used for remembering previous experiences.

The reward is the criterion for how well the obtained FSM output matches the desired output. For example, the targets defined in Table 5 could be used.

3.3. Use Case of Invisible Features Detection by Clusterization

This approach was appointed to find unseen by human features of FSM usage through clusterization of results based on predefined criteria. The idea, as shown in Figure 3, is to collect a large dataset of simulation results using FSM and use Unsupervised Machine Learning approach. AI will perform clusterization of the obtained results and allow for the identification of new features based on new clusters.

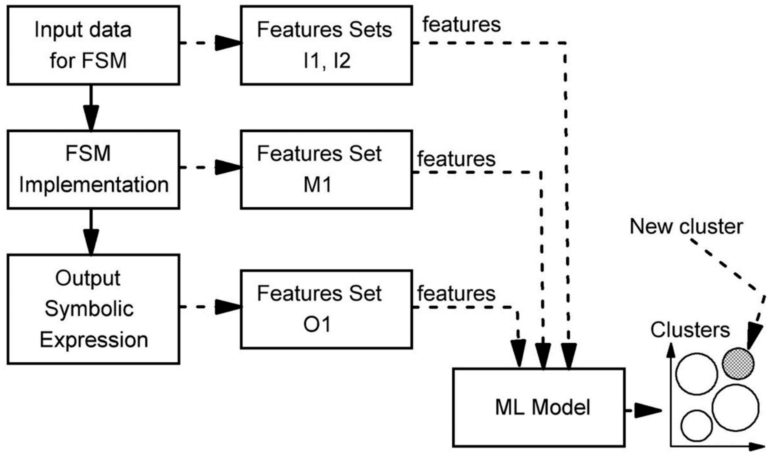


Figure 3. Use case of new FSM feature detection.

3.4. Use Case of Prediction of FSM Output

This idea is like the previous one but uses Supervised Machine Learning. Figure 4 shows the use case when ML is already trained. For training, the dataset consists of feature sets of type I1 and I2, corresponding to feature sets of type O1.

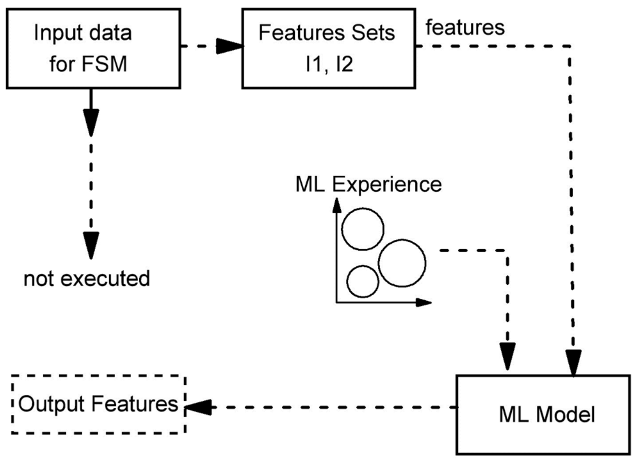


Figure 4. Use case of FSM output feature prediction (ML model already trained).

For example, the complexity of the transfer function is not provided as an input. Here, AI tries to predict the symbolic expression's complexity based on previous experience.

3.5. Use Case of Assistance in Frequency Symbolic Method Usage Based on Anomaly Detection

The idea, as shown in Figure 5, is to use a trained ML system as an "Assistant" for users of FSM implementation. For training, Supervised Machine Learning should be used. First, AI is trained by collecting a dataset of symbolic expression (obtained using FSM) features to corresponding inputs. Once trained, it could be used as an "Assistant" (supervisor) for the following FSM usage. For example, the ML system can detect anomalies and inform the user. An anomaly could be a human mistake, for example, when the conductivity matrix is not as usual or a parametric element is incorrectly connected.

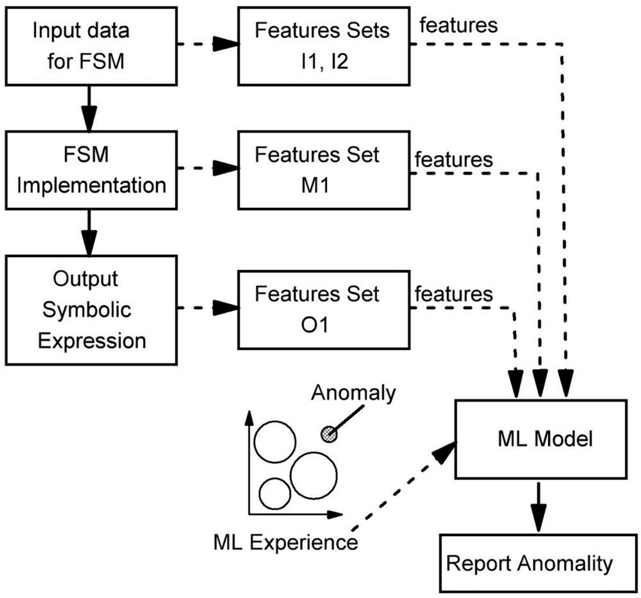


Figure 5. Use case of AI-based assistance for FSM usage.

3.6. Use Case of Fault Diagnosis and Testing

The merging of the proposed methods with AI techniques opens new avenues for enhancing the analysis and optimization of LPTV circuits. The incorporation of AI, particularly machine learning (ML), offers significant advantages in handling the complex and computationally intensive tasks associated with the frequency symbolic method. A special case is fault diagnosis which should be differentiated from typical functional testing. It is possible for the circuit to be in a faulty state but still pass functional test, which is equivalent to behavioral test. In such a case, one or more elements can have parameters, e.g., beyond the tolerance range. This can mark the beginning of a component's degradation process (due to aging or environmental influence) and can affect the circuit (system) lifetime (or Mean-Time Between Failure parameter, MTBF). Successful fault diagnosis can be an important part of predictive maintenance, especially in cases of a component's soft fault (parameter beyond the tolerance range while still operating), as it can be a step before hard fault (malfunction), often leading to circuit (system) malfunction. Exemplary areas are as follows:

1. Fault Diagnosis in Electrical/Electronic Circuits:
   - Predictive Maintenance: AI can be used to predict potential failures in circuits by analyzing patterns and anomalies in frequency responses. By training ML models on historical data of circuit performance, it is possible to foresee faults before they occur, allowing for proactive maintenance [13,17].
   - Anomaly Detection: AI systems can be employed to detect anomalies in circuit behavior that may indicate faults. Machine learning models can learn the normal operating conditions of circuits and flag deviations from these norms, thus identifying potential issues in real-time.

2. Testing and Validation:
    - Automated Testing: AI-driven automated testing procedures can significantly reduce the time and effort required for validating circuit designs. By using reinforcement learning techniques, AI systems can optimize test scenarios to cover a wide range of operating conditions, ensuring robust performance.
    - Simulation-Based Testing (SBT): AI can enhance simulation-based testing by rapidly generating and evaluating numerous test cases, thus providing comprehensive insights into circuit behavior under various conditions. This approach can help identify edge cases and potential weaknesses in the design.
    - Parameter Tuning: AI techniques can be used to optimize the parameters of LPTV circuits. Genetic algorithms and other optimization techniques can help find the best parameter values that improve circuit performance while meeting design constraints.

By integrating AI techniques with the frequency symbolic method, the processes of circuit analysis, fault diagnosis, testing, and optimization can be improved. This integration not only enhances efficiency but also provides a deeper understanding of the complex behaviors of LPTV circuits, paving the way for more reliable electrical and electronic systems.

3.7. Example of Symbolic Transfer Function Complexity Calculation

This chapter contains an example of symbolic expression complexity calculation based on a transfer function obtained in symbolic form using the FSM. Specifically, it shows the extraction of feature O1.2, "Complexity of transfer function". The complexity value obtained is an integer number, which could be used in a similar way to extract the features with IDs I2.4, M1.2, O1.2, and O1.3 described above. For this example, the one-circuit parametric amplifier shown in Figure 6 has been selected.

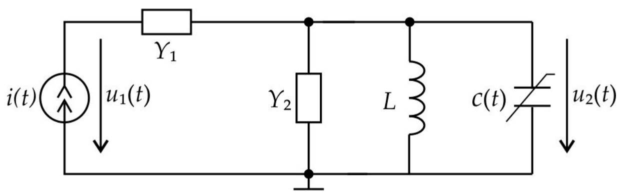


Figure 6. Model of a single parametric-variable circuit.

A parametric circuit with a single varying coefficient, shown in Figure 6, has been analyzed. The time-dependent parameter $c(t)$ of the system is the capacitance, which is represented by a function:

$$c(t) = C_0\{1 + m_c\cos(\Omega t)\},\ \ C_0 = 10\text{ pF}. \tag{13}$$

The parameters of the LPTV circuit shown in Figure 6 are as follows: a constant angular frequency Ω for the parametric function $c(t)$: $\Omega = 2\pi f$, where f = 200 MHz, and a modulation coefficient m_c = 0.01. The inductance is L = 253.3 nH, and the conductance values are Y_1 = 0.25 S and Y_2 = 0.4 mS. A monoharmonic input signal is a current $i(t)$ with a time waveform:

$$i(t) = I_m\cos\left(\omega t + \varphi_0\right),\ I_m = 0.1\text{ mA},\ \varphi_0 = 45^{\circ},\ \omega = 2\pi f_{in},\ f_{in} = 100\text{ MHz}. \tag{14}$$

Three parameters remain in symbolic form: time t, constant angular frequency of the parametric function Ω, and m_c—modulation coefficient of the time-varying capacitance c. Using FSM, the transfer function was obtained for three cases: 1, 2, and 3 harmonics used to approximate the transfer function. It can be noted that the FSM makes it possible to obtain the approximation of the transfer function using Fourier series with a predetermined number of harmonics. Figure 7 (magnitude) and Figure 8 (phase) present plots of the transfer function (with 1 harmonic approximation) for presented circuit. It can be observed that there is an influence of variable capacitance at frequency f = 200 MHz.

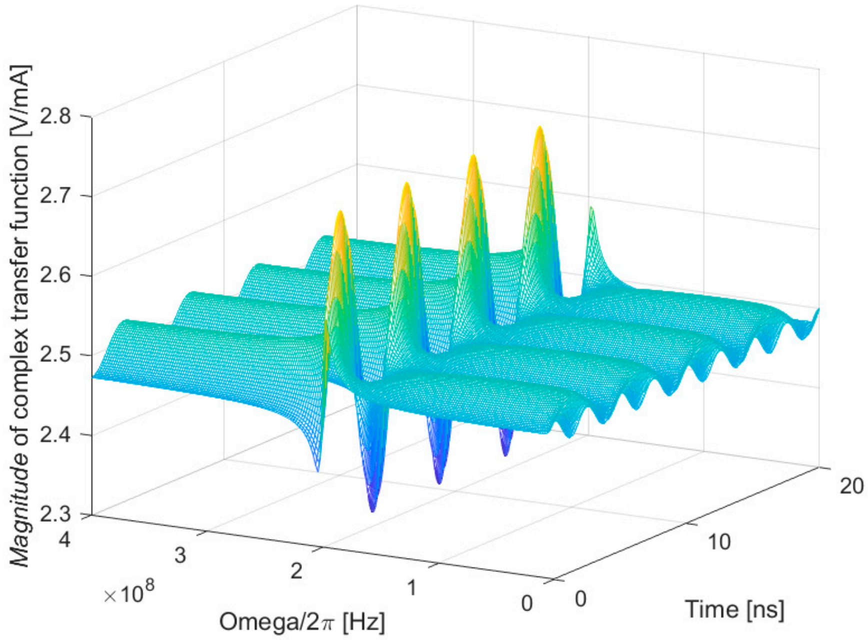


Figure 7. Magnitude of complex transfer function (for 1 harmonic).

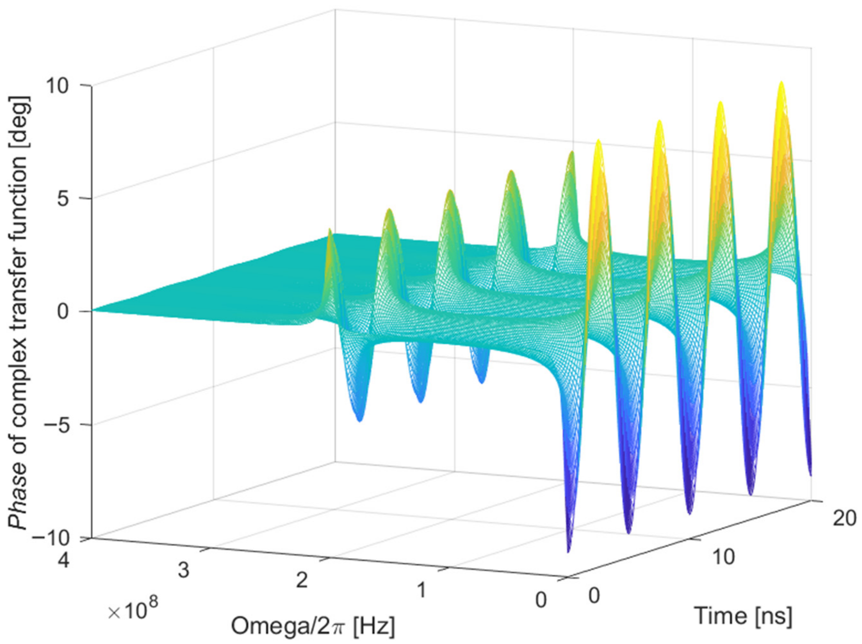


Figure 8. Phase plot of complex transfer function (for 1 harmonic).

The following formula is proposed to calculate the complexity C of an expression (transfer function):

$$C = \sum_{k=1}^{K} w_i \cdot N_i, \tag{15}$$

where:

N_i is number of particular operations (for i-th mathematical operand),
w_i is complexity weight of the i-th operand,
K is total number of distinguished operands (depending on the case).

There following operands are distinguished (with abbreviations): addition (add), subtraction (sub), multiplication (mul), division (div), and exponentiation (exp). Negative numbers are not counted as subtraction (they have assumed particular hardware representation).

Deciding on the weights of each operation is a debatable topic, as it depends very much on the tools, algorithms, hardware implementation, etc. Therefore, the following assumptions have been made for this example, which may not apply to other environments:

1. Every number in the expression in the generic case is a complex number and it serves as our reference.
2. The addition or subtraction of two complex numbers has the same complexity as two additions/subtractions of real numbers and is considered as a unit weight with value $w_{add} = w_{sub} = 1$.
3. Multiplication of two real numbers typically uses hardware acceleration, resulting in a weight of 1.
4. Complex multiplication involves two multiplications and one addition, and based on the assumptions above, its weight is $w_{mul} = 3$.
5. The division of two complex numbers uses multiplication by the conjugate, and its weight is $w_{div} = 7$.
6. Binary exponentiation is calculated with complexity $O(2 \cdot \log_2 n)$. The real and imaginary parts of the complex number are stored as 32-bit values, so $\log_2 n = 32$, therefore the weight of the exponentiation is $w_{exp} = 64$.

Summary of weights for complexity calculation is presented in Table 6.

Table 6. Assumed complexity weights for particular mathematical operations.

Operation	Weight
add	1
sub	1
mul	3
div	7
exp	64

Table 7 shows the number of operations and the complexity calculated using expression (11) and the assumptions considered above for the transfer functions obtained using FSM with 1 harmonic approximation. Tables 8 and 9 show the calculation complexities for transfer function calculation using 2 and 3 harmonics, respectively.

It can be observed that the required complexity increases rapidly with the number of harmonics used to find a particular transfer function.

Table 7. Result of transfer function complexity calculation using FSM for 1 harmonic.

Operation	Number of Operations	Cost
add	48	48
sub	33	33
mul	67	201
div	3	21
exp	44	2816
		Total 3119

Table 8. Result of transfer function complexity calculation using FSM for 2 harmonics.

Operation	Number of Operations	Cost
add	240	240
sub	238	238
mul	433	1299
div	5	35
exp	373	23,872
		Total 25,684

Table 9. Result of transfer function complexity calculation using FSM for 3 harmonics.

Operation	Number of Operations	Cost
add	722	722
sub	626	626
mul	1213	3639
div	7	49
exp	1081	69,184
		Total 74,220

4. Conclusions

This paper examines five instances of machine learning (ML) applications to the FSM method. For each use case, the requisite feature sets, learning type, benefits, and potential issues are analyzed. A summary is presented in Table 10.

Table 10. Use cases summary.

Use Case	Benefit	Learning Type	Difficulties/Problems
Parametric circuits synthesis	Circuit could be synthesized automatically.	Reinforcement learning	Could be time consuming. Big database is needed for ML experience storage.
New features detection	Identify not visible features of the FSM method or its implementation by specific clusters occurrences.	Unsupervised learning	At the moment, it is not clear which clusters could be obtained.
Prediction	Could predict the result based on previous experience without modeling and simulation.	Supervised learning	For some not typical inputs, the result could be predicted incorrectly.
Anomaly detection	Human mistakes could be detected. This use case acts as an assistant.	Supervised learning	Could accidentally report an anomaly when the correct approach is used.

The initial use case is parametric circuit synthesis, in which a circuit can be synthesized automatically through reinforcement learning. The primary challenges are the time-consuming process and the necessity for a substantial database for ML experience storage. The second use case pertains to the detection of new features, which involves identifying features of the FSM method or its implementation that are not visible or cannot be discerned from specific cluster occurrences. This use case employs unsupervised learning, but it is currently unclear which clusters could be obtained. The third use case is prediction, where results can be predicted based on previous experience without the need for modeling and simulation. This use case employs supervised learning, but the result may be incorrectly predicted for atypical inputs. The fourth use case involves an assistant that could detect human mistakes through unsupervised learning.

However, there was a potential problem of inadvertently reporting an anomaly when the correct approach was used. The fifth use case was fault detection, which allowed the identification of faults not detected by standard functional tests. This use case used supervised learning but required a sufficient circuit model (simulation). There are other issues that must always be properly addressed, such as data preparation, model selection, training, and evaluation, with the aim of improving the overall efficiency of the diagnostic system, but these are all circuit-dependent and must be chosen by a test designer, engineer, or AI expert. Therefore, there is no single or best solution for a particular test case, so we have proposed a set of real-world applications that should reduce the time taken to develop a single test for a custom circuit.

Author Contributions: Conceptualization, Y.S. and D.G.; Methodology, Y.S. and D.B.; Software, S.M.; Validation, S.M. and Ł.C.; Formal analysis, Y.S. and D.B.; Investigation, D.B., S.M. and A.P.; Resources, D.B., A.P. and S.M.; Data curation, D.B., S.M. and Y.S.; Project administration, D.G. and A.P.; Visualization, S.M., D.B. and A.P.; Writing—original draft, Y.S., D.B., S.M. and A.P.; Writing—review and editing, Ł.C., S.M., A.P. and D.G.; Supervision, D.G; Funding acquisition, D.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: This paper is supported by the Polish Ministry of Science and Higher Education and Ministry of Education and Science of Ukraine funding for statutory activities.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zadeh, A. Frequency Analysis of Variable Networks. *Proc. IRE* **1950**, *38*, 291–299. [CrossRef]
2. Papandreou-Suppappola, A. *Applications in Time-Frequency Signal Processing*; CRC Press Inc: Boca Raton, FL, USA, 2003.
3. Ortigueira, M.D.; Martynyuk, V.; Kosenkov, V.; Batista, A.G. A New Look at the Capacitor Theory. *Fractal Fract.* **2023**, *7*, 86. [CrossRef]
4. Ding, Y.; Deng, H.; Xie, Y.; Wang, H.; Sun, S. Time-Varying Channel Estimation Based on Distributed Compressed Sensing for OFDM Systems. *Sensors* **2024**, *24*, 3581. [CrossRef] [PubMed]
5. Dhahri, S.; Naifar, O. Fault-Tolerant Tracking Control for Linear Parameter-Varying Systems under Actuator and Sensor Faults. *Mathematics* **2023**, *11*, 4738. [CrossRef]
6. Jia, X.; Jiang, Y. Non-Magnetic Circulator Based on a Time-Varying Phase Modulator. *Appl. Sci.* **2023**, *13*, 512. [CrossRef]
7. Grabowski, D.; Maciążek, M.; Pasko, M.; Piwowar, A. Time-invariant and time-varying filters versus neural approach applied to DC component estimation in control algorithms of active power filters. *Appl. Math. Comput.* **2017**, *319*, 203–217. [CrossRef]
8. Eom, B.H.; Day, P.K.; LeDuc, H.G.; Zmuidzinas, J. A wideband, low-noise superconducting amplifier with high dynamic range. *Nat. Phys.* **2012**, *8*, 623–627. [CrossRef]
9. Wiechetek, K.; Piskorowski, J. A Concept of the Non-Stationary Filtering Network with Reduced Transient Response. *Appl. Sci.* **2019**, *9*, 4570. [CrossRef]
10. Zhang, H.; Guoan, B.; Zhao, L.; Razul, S.G.; See, C.-M.S. Time varying filtering and separation of nonstationary FM signals in strong noise environments. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, Italy, 4–9 May 2014; pp. 4171–4175. [CrossRef]
11. Ou, B.; Liu, D. Chaotic attractor generation via a simple linear time-varying system. *Discret. Dyn. Nat. Soc.* **2010**, *2010*, 840346. [CrossRef]
12. Koconń, S.; Piskorowski, J. Time-Varying IIR Notch Filter with Reduced Transient Response Based on the Bézier Curve Pole Radius Variability. *Appl. Sci.* **2019**, *9*, 1309. [CrossRef]
13. Golonek, T.; Chruszczyk, Ł. Analog Circuits Specification Driven Testing by Means of Digital Stream and Non-Linear Estimation Model Optimized Evolutionarily. *Bull. Pol. Acad. Sci. Tech. Sci.* **2020**, *68*, 1283–1299. [CrossRef]
14. Yin, Z.; Jiang, X.; Zhang, N.; Zhang, W. Stability Analysis for Linear Systems with a Differentiable Time-Varying Delay via Auxiliary Equation-Based Method. *Electronics* **2022**, *11*, 3492. [CrossRef]
15. Shapovalov, Y.; Bachyk, D.; Detsyk, K. Multivariate Modelling of the LPTV Circuits in the MAOPCs Software Environment. *Prz. Elektrotech. (Electr. Rev.)* **2022**, *98*, 158–163. [CrossRef]

16. Polyanin, A.D.; Zaitsev, V.F. *Handbook of Exact Solutions for Ordinary Differential Equations*, 2nd ed.; Chapman & Hall/CRC: Boca Raton, FL, USA, 2003.
17. Pawełczyk, R.; Grzechca, D. Improvement of functional safety of the level crossing barrier machine by a noninvasive angle detection method. *IEEE Des. Test* **2022**, *39*, 43–53. [CrossRef]

applied sciences

MDPI

Article

CNN-Based Damage Identification of Submerged Structure-Foundation System Using Vibration Data

Ngoc-Lan Pham, Quoc-Bao Ta and Jeong-Tae Kim *

Department of Ocean Engineering, Pukyong National University, 45 Yongso-ro, Nam-gu, Busan 48513, Republic of Korea; pnlan@pukyong.ac.kr (N.-L.P.); tabao838@pukyong.ac.kr (Q.-B.T.)
* Correspondence: idis@pknu.ac.kr; Tel.: +82-51-629-6585

Abstract: This study presents a convolutional neural network (CNN) deep learning approach for identifying damage in submerged structure-foundation systems using vibration data. Firstly, foundation damage in a lab-scale caisson-foundation system is simulated to measure time-history responses. Singular value decomposition (SVD) responses are derived from the time-history responses. Secondly, the 1-D CNN deep learning model is trained using both the time-history responses and SVD responses. Finally, the trained CNN models are implemented to evaluate the foundation damage under conditions of noise contamination and partially untrained data. The experimental results demonstrate the effectiveness of CNN models for damage identification and highlight the comparative strengths of time-history and SVD data. The CNN model trained using SVD data outperforms the other model when under noise contamination conditions, while the CNN model trained using time-history data maintains better accuracy in partially untrained data conditions. Integrating both types of data enhances the accuracy of damage classification.

Keywords: submerged structure-foundation system; vibration monitoring; time-history response; singular value decomposition response; 1-D CNN; deep learning; damage identification

1. Introduction

Submerged structure-foundation systems, such as caisson breakwater systems, face significant challenges from extreme conditions like earthquakes and storm surges over their operational lifespan [1]. Caisson systems consist of an above-water concrete cap, the submerged caisson, and the foundation. Over time, several issues can develop in the foundation, such as settlement, overturning, and sliding [2,3]. Additionally, the caisson can suffer from large storm surges, causing severe cracking and material leakage [4]. Due to the inaccessibility of underwater components, innovative evaluation methods are required to assess their structural integrity.

Among several structural health monitoring techniques, vibration-based monitoring has emerged as an effective method for assessing the damage in submerged structures [5–7]. Most studies have focused on manual damage identification using vibration features [8–10]. Vibration features are complicated and are affected by foundation conditions and environmental factors [11,12]. As a result, assessing submerged structure-foundation systems has remained challenging due to the uncertainties associated with field parameters. Further research is needed to address these complexities and improve the reliability of structural integrity assessments.

Recent studies have demonstrated the effectiveness of 1-D convolutional neural networks (CNNs) in real-time applications and the ability to learn complex tasks. Such 1-D CNN models can be trained using vibration features such as singular value decomposition (SVD) and time-history responses. SVD data are obtained from multiple preprocessing steps for each damage level [13]. As an alternative approach, time-history responses from accelerometers can be directly utilized for the CNN model, bypassing several preprocessing steps, thereby enhancing speed and reducing computational costs [14–16].

There has been a lack of research comparing CNN models trained using different vibration features, and investigating how integrating results could improve prediction performance. This study uses both time-history responses and SVD responses for the CNN model. The advantages and disadvantages of each type of training data are identified by evaluating the CNN model under noise contamination and partially untrained data conditions. The main contributions of this study are as follows:

- A CNN deep learning approach is developed to identify damage in submerged structure-foundation systems using both time-history and SVD data.
- The performance of the CNN model is evaluated with partially untrained cases, where certain damage levels were excluded from the training and validation datasets.
- A comparative study between CNN trained using time-history and SVD data is conducted. The results from the two models are integrated to strengthen the damage classification performance.

To achieve these objectives, a CNN deep learning approach for identifying damage in submerged structure-foundation systems using both time-history and SVD data is developed. Firstly, foundation damage in a lab-scale caisson-foundation system is simulated to measure time-history responses. SVD responses are derived from the time-history responses. Secondly, the 1-D CNN deep learning model is trained using both the time-history responses and SVD responses. Finally, the trained CNN models are implemented to evaluate the foundation damage under conditions of noise contamination and partially untrained data.

2. Literature Reviews

2.1. Vibration-Based Techniques for Caisson-Foundation Systems

Many researchers have utilized vibration-based techniques to handle the transient interactions between waves, structures, and foundations [3,8]. For caisson-foundation structures, vital structural information can be obtained from a few sensors strategically placed in the small above-water section [5–7]. Ming et al. [17] studied the dynamic response of a caisson breakwater using a rigid body on an elastic foundation model under wave action. Lee et al. [8] employed vibration-based methods to detect damage at the interface between the caisson and the foundation. Huynh et al. [9,10] combined simplified analytical approaches with in situ vibration measurements to assess caisson-foundation system integrity. Lee et al. [11,12] developed a practical scheme to differentiate the impact of foundation conditions and environmental factors on the vibration characteristics of the breakwater system. Pham et al. [13] proposed the pseudo-wave method superposing the vibration responses from several man-made excitations to simulate the wave impact on the caisson system.

2.2. 1-D CNN Deep Learning and Data Acquisition

The compact 1-D CNN excels in extracting features from vibration data with minimal training required [18,19]. The vibration data (i.e., time-history and SVD responses) are one-dimensional arrays with inherent order. The 1-D CNN model is suitable for processing 1-D and sequential input data, as it can effectively learn the ordered patterns by capturing local dependencies. Teng et al. [14] applied 1-D CNNs trained using vibration signals from a numerical model to detect damage in a steel bridge. Huang et al. [15] employed a CNN model to quantify the dynamic stiffness of railway tracks using vibration data. Li et al. [16] implemented an adaptive CNN model for bearing-fault diagnosis in noisy environments.

The vibration data for the CNN model can be obtained by two approaches. In the first approach, vibration data are collected directly from damaged caissons after extreme events such as storms or tsunamis. These data are labeled according to the severity of the damage and used for training. In the second approach, vibration data are collected indirectly through foundation damage simulation. Pham et al. [13] proposed a method called pseudo-damage simulation, where concrete weights were applied on top of the caisson to mimic the effect of foundation damage. In their study, experiments were conducted under laboratory conditions. The location, shape, and severity of foundation damage levels were predetermined.

3. Methodology

3.1. Research Framework

Detecting damage using conventional methods is challenging due to the complexity of the responses. The 1-D CNN approach can be practical to overcome this challenge. The 1-D CNN method trained using vibration data (i.e., time-history and SVD responses) is applied to automate the feature extraction process and to facilitate real-time monitoring.

Figure 1 illustrates the CNN deep learning model scheme for caisson-foundation system using vibration data. This scheme comprises three phases: (1) data acquisition via the vibration technique, (2) construction of the 1-D CNN model for the caisson-foundation system, and (3) evaluation of the 1-D CNN model. In Phase 1, the databanks corresponding to foundation damage on the caisson system are generated for deep learning models. The foundation damage simulation involves removing a certain amount of sand and gravel from the existing foundation. The vibration data (i.e., time-history and SVD responses) are acquired by accelerometers under man-made excitation. The time-history responses obtained from several accelerometers are concatenated to form one continuous response. The SVD responses are derived from the time-history responses using the frequency domain decomposition (FDD) method. The vibration data are processed with data augmentation to form a databank and used for the CNN model in the subsequent phases.

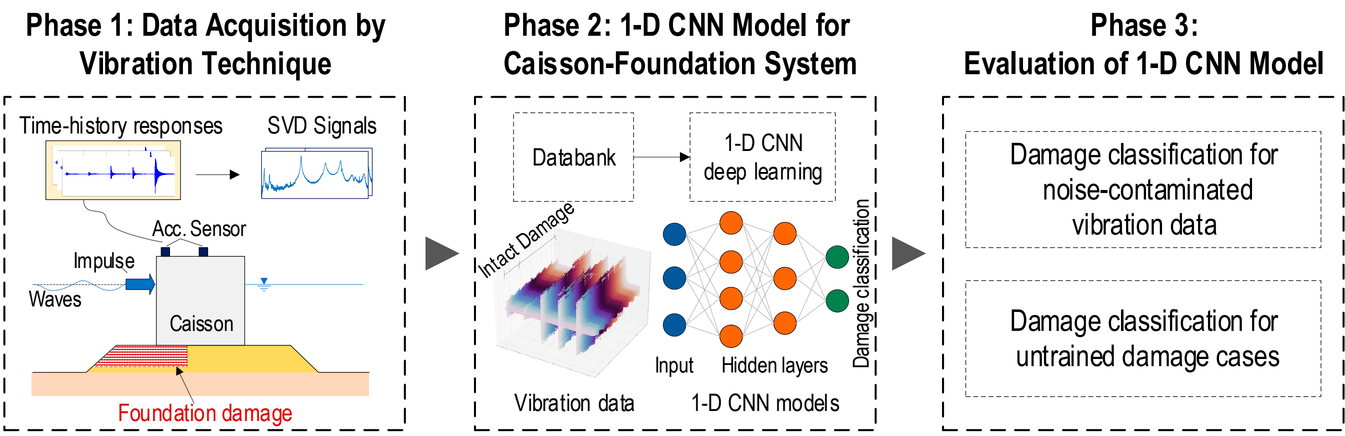


Figure 1. Scheme of CNN deep learning model for a caisson-foundation system using vibration data.

In Phase 2, a 1-D CNN model is developed to detect damage in the caisson-foundation system. The databank is augmented by introducing noise to the vibration. The 1-D CNN model processes the vibration data through several hidden layers to classify the damage levels in the foundation. In Phase 3, the effectiveness of the 1-D CNN model is evaluated for noise-contaminated and untrained damage cases.

3.2. Vibration-Data Acquisition Technique

3.2.1. Vibration Monitoring Method

Figure 2 illustrates a vibration monitoring setup on a submerged caisson-foundation system. This system has three main components: an above-water concrete cap, a submerged caisson, and a foundation. The foundation is constructed with a sand mound and gravel armor. The caisson is oriented perpendicular to the direction of wave propagation and is subjected to impulsive breaking wave forces. The vibration test for in situ submerged caisson-foundation systems uses wave forces as the excitation source. Alternatively, a tugboat with adjustable weight and speed can be used to generate controlled artificial excitation for the vibration tests.

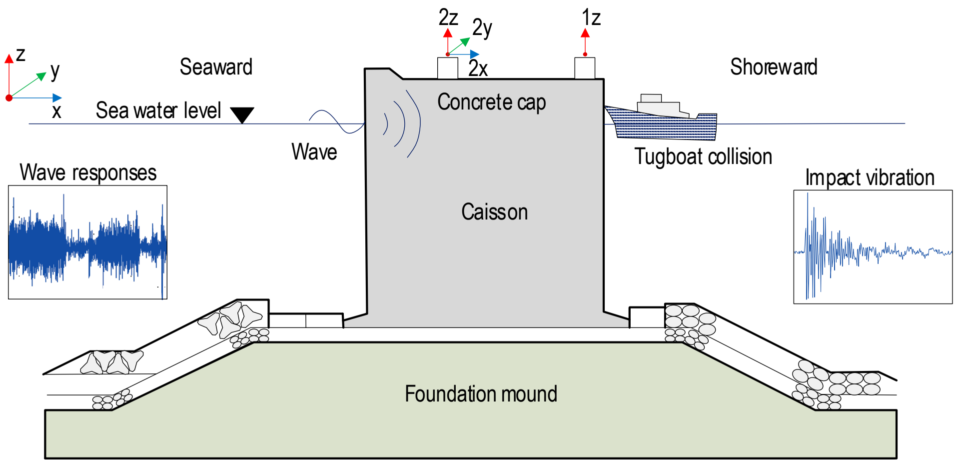


Figure 2. Vibration monitoring setup on a submerged caisson-foundation system.

Figure 3 illustrates vibration monitoring for a caisson-foundation system. Several accelerometers are strategically installed on the above-water concrete cap of the caisson to obtain vibration data. These accelerometers are oriented in three directions (x, y, and z) to capture all caisson motions, including heave, sway, and roll. In this study, the time-history responses are acquired by five accelerometers (i.e., ACC 1z, ACC 2z, ACC 2x, ACC 3x, and ACC 2y) for 15 s at a sampling rate of 1 kHz. Only 1000 data points that contain the most important information (i.e., 100 data points before the impact load and 900 data points after) are chosen. These time-history responses from the five sensors are then concatenated to form one continuous response, as shown in Figure 3b. The concatenating technique transforms the data into a 1-D array which is well-suited for the 1-D CNN model. The concatenated signal contains the information of all five accelerations with inherent order. The 1-D CNN model can simultaneously learn the characteristics of time-history responses from all sensors. In addition, concatenating allows the 1-D CNN model to capture the relation movements of five sensors.

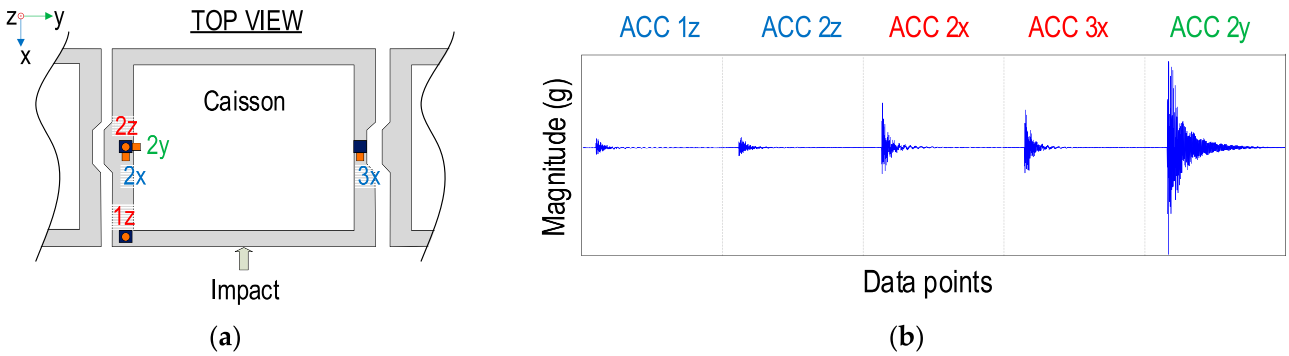


Figure 3. Vibration measurements for a caisson-foundation system: (**a**) accelerometer setup and impact load; (**b**) Concatenated time-history response.

3.2.2. Vibration Feature Extraction

Since frequency domain decomposition (FDD) theory was first described by Brincker et al. [20], the method has become popular due to its robustness and efficiency. The FDD method decomposes the vibration response into a series of independent single-degree-of-freedom systems. This is achieved by applying the SVD to the cross-spectral density (CSD) matrix [20,21], which outputs the natural frequencies and mode shapes of the system.

There are four main steps in the FDD method. In the first step, the CSD matrix is computed from the acceleration responses collected from n sensors on the structure. The general form of the CSD matrix $S_{yy}(\omega)$ is as follows:

$$S_{yy}(\omega) = \begin{bmatrix} S_{11}(\omega) & S_{12}(\omega) & \dots & S_{1n}(\omega) \\ S_{21}(\omega) & S_{22}(\omega) & \dots & S_{2n}(\omega) \\ \vdots & \vdots & \ddots & \vdots \\ S_{n1}(\omega) & S_{n2}(\omega) & \dots & S_{nn}(\omega) \end{bmatrix} \tag{1}$$

In the second step, the CSD matrix is decomposed into unitary matrices (i.e., $U(\omega)$, $V(\omega)$), and diagonal matrix $\sum(\omega)$) using the SVD. Due to the symmetricity of $S_{yy}(\omega)$, the unitary matrices $U(\omega)$ and $U(\omega)$ are identical. The diagonal matrix $\sum(\omega) = diag\{\sigma_n(\omega)\}$ contains singular values $\sigma_n(\omega)$ in descending order, with the first value, $\sigma_1(\omega)$, representing the highest energy of the dynamic system. The new form of the CSD matrix $S_{yy}(\omega)$ is as follows:

$$S_{yy}(\omega) = U(\omega)^T \ \sum(\omega) \ V(\omega) \tag{2}$$

In the third step, natural frequencies are identified by selecting peak frequencies ω_p on the singular value chart. The final step involves extracting mode shapes from the first column vector of $U(\omega_p)$ at these peak frequencies.

Considered as a linear structural system with light damping, the caisson system exhibits sharp resonance peaks in the SVD chart [22]. These peaks contain critical structural information, making them more significant compared to other regions of the frequency spectrum. To fully exploit the informational content within the SVD responses, a min–max normalization in a logarithmic scale technique is employed as follows [13]:

$$x' = \frac{\log(x) - \log(10^{-6})}{\log(10^{-1}) - \log(10^{-6})} \tag{3}$$

where the range from 10^{-6} to 10^{-1} for normalization is empirically chosen to suit the specific characteristics of the data used for deep learning.

3.3. Architecture of a 1-D CNN Model

Figure 4a illustrates the architecture of a 1-D CNN classification model designed to detect foundation damage. The 1-D CNN model trained using time-history data receives 1×5000 input data, while the 1-D CNN model trained using SVD data receives 1×2048 input data. The 1-D CNN model extracts essential features of vibration data (i.e., concatenated time-history and SVD responses) through multiple CNN layers before providing classification results for the level of foundation damage. The configuration of the 1-D CNN layers is empirically chosen for the specific data characteristics, as detailed in Table 1. The CNN model includes three convolution layers, three max-pool layers, three ReLU layers, one flatten layer, and three dense layers. Convolutional layers are responsible for feature extraction from the input signals. Max-pooling layers then down-sample these features to reduce computational load and mitigate overfitting risks. ReLU layers introduce non-linearity to the network by converting all negative values to zero. The flatten layer reshapes the extracted features into a format suitable for the dense layers. Each node in a dense layer is connected to every node in the previous layer, with each connection associated with a learnable weight parameter. The CNN models are trained with 40 epochs, a learning rate of 0.001, and a batch size of 1. During training, the model minimizes the difference between the predicted and actual labels using the categorical cross-entropy loss function. The accuracy metric is utilized to evaluate the classification performance of the trained model.

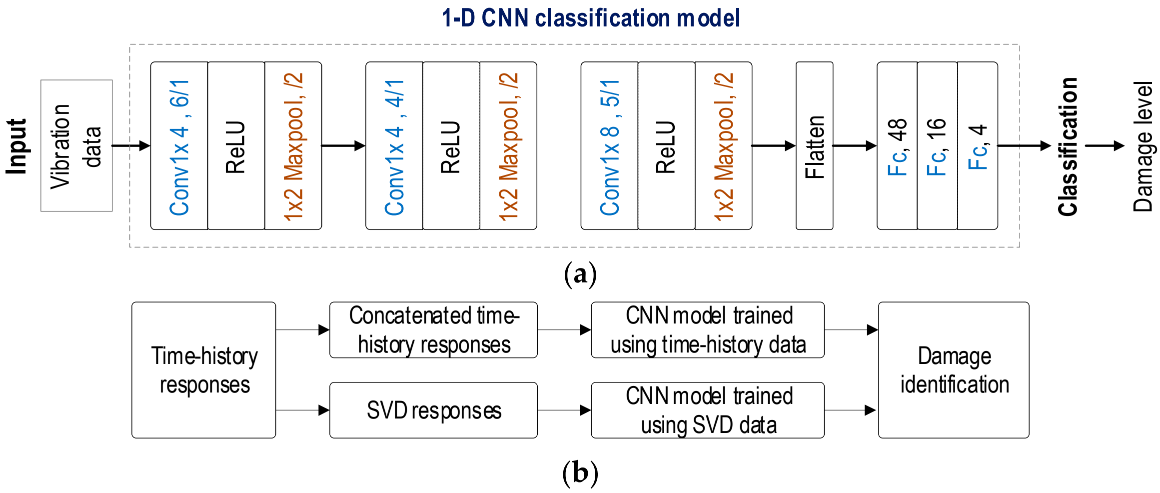


Figure 4. Schematic of a 1-D CNN classification model: (**a**) 1-D CNN architecture; (**b**) data preparation and training procedures.

Table 1. Specification of 1-D CNN layers.

No.	Type	Depth	Filter	Stride	No.	Type	Depth	Filter	Stride
1	Conv1	6	1 × 4	1	8	ReLU	-	-	-
2	ReLU	-	-	-	9	Maxpool3	-	1 × 2	2
3	Maxpool1	-	1 × 2	2	10	Flatten			
4	Conv2	4	1 × 4	1	11	Fc1	48	-	-
5	ReLU	-	-	-	12	Fc2	16	-	-
6	Maxpool2	-	1 × 2	2	13	Fc3	4	-	-
7	Conv3	5	1 × 8	1	14	Classification	-	-	-

Figure 4b shows the data preparation and training procedure for a caisson-foundation system. The time-history responses obtained under foundation damage cases are concatenated and are input into the CNN model. The time-history responses are used to extract the SVD responses. The extracted responses are used for the CNN model trained using SVD data. The prediction results of two CNN models are integrated for damage identification of the caisson-foundation system.

Robustness of the model can be achieved by ensuring the consistency and sensitivity of the training data. To ensure data consistency, experimental variations such as impact load magnitude, temperature fluctuations, and ambient noise are carefully controlled in lab-scale conditions. To achieve the high sensitivity of the training data, the training data are focused on regions susceptible to damage variations (i.e., here, 1000 data points surrounding the impact event were selected for the time-history analysis, and 2^{11} = 2048 data points for the SVD analysis).

To enhance the model's performance and prevent overfitting issue, data shuffling and early-stopping techniques are implemented [23]. Data shuffling helps to prevent the model from learning the order of the data, and the early-stopping halts training when the model's performance on the validation set begins to degrade.

3.4. Foundation-Damage Classification Approach

3.4.1. Deep Learning of Noise-Contaminated Databank

Several factors can affect the vibration test, such as noise (including environmental noise and electrical noise from the monitoring devices) and inconsistencies in the excitation source. The data are enriched by obtaining a large number of ensembles to capture as much variation in the excitation source as possible. Despite efforts to control the noise and the magnitude of the vibration source, variations in vibration data are inevitable. Conducting comprehensive experiments that account for all of these variables is challenging and costly. Data augmentation by injecting Gaussian noise into the measured signals is a practical solution for simulating realistic measurement conditions. Introducing noise into the data is also an effective method for enriching the dataset. Several noise levels injected ensure that the CNN model is exposed to a wider range of data during training. It leads to a more robust and general model that can perform on unseen and real-world vibration data.

Figure 5 illustrates the process for configuring the noise-contaminated databank. The vibration data were obtained in 15 ensembles for each foundation damage level. Eighty percent (80%) of the total vibration data was used for training and validation datasets. Data generation used Gaussian noise with standard deviations of 0%, 1%, 2%, 3%, 4%, and 5% of the signal amplitude. The remaining 20% of the total vibration data had noise levels injected ranging from 1% to 16% (in 1% increments) to form the evaluation dataset. This evaluation dataset was used to examine the effect of noise on the accuracy of the 1-D CNN model.

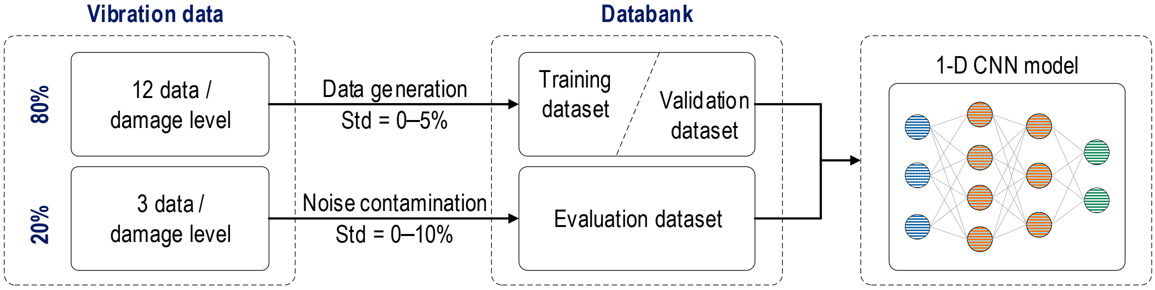


Figure 5. Configuration of the noise-contaminated databank.

3.4.2. Deep Learning of Partially Untrained Databanks

The foundation of the caisson system sustains various types of damage after extreme events such as storms or tsunamis. For training a CNN model, vibration data can be collected directly from these damaged caissons and labeled according to the severity of the damage. However, a concern is the availability of such data. With variations in shape, location, and severity of foundation damage, each collected sample differs and may not correctly represent feasible damage scenarios. Therefore, it is necessary to train the CNN model by utilizing limited data as well as to recognize unseen damage cases by classifying them into the trained databank of the CNN model. To assess the capability of the 1-D CNN model, certain damage levels were excluded to create partially untrained databanks.

As shown in Figure 6, noise levels ranging from 1% to 5% were added to the 12 ensembles to create the training and validation datasets. Three untrained cases were designed to examine the effect of missing data on the performance of the 1-D CNN. In Case 1, the training and validation set excluded three damage levels: D2, D4, and D6 (from eight damage levels D0–D7), while in Case 2 and Case 3, the training and validation set excluded four damage levels (Case 2: D1, D3, D5, and D7; Case 3: D1, D2, D5, and D6). For the evaluation dataset, the remaining three ensembles were injected with noise levels from 1% to 5% with intervals of 1%.

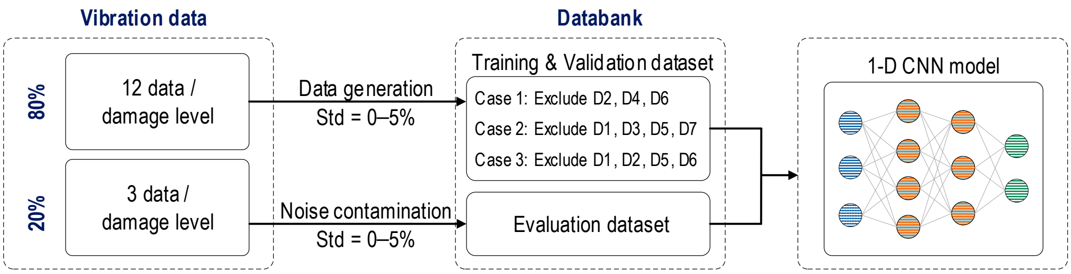
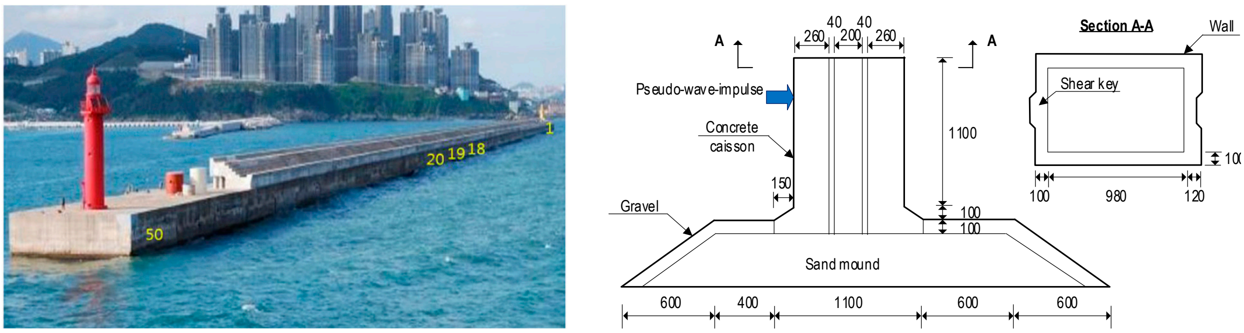


Figure 6. Configuration for a partially untrained databank.

4. Experiment on a Lab-Scale Caisson-Foundation System

4.1. Vibration Test on a Lab-Scale Caisson System

Figure 7a shows an Oryuk-do caisson-foundation system investigated by Lee et al. [12]. The submerged breakwater system consists of 50 caisson units and has a total length of 1004 m. Each caisson unit is 20 m in width, 20 m in length, and 20.78 m in height. As detailed in Figure 7b, the concrete caisson in the lab-scale experiment was scaled down to 1/20 of the real size, with measurements of 1200 mm in width, 1100 mm in length, and 1300 mm in height. The foundation mound of the lab-scale system is made of a sand layer of 480 mm and an armor gravel layer of 100 mm.



(**a**) (**b**)

Figure 7. Parameters of the caisson breakwater system: (**a**) Oryuk-do caisson breakwater system [19]; (**b**) lab-scale caisson-foundation model [13].

Figure 8 shows the experimental setup for the vibration test on the lab-scale caisson-foundation system. The foundation consisted of a sand mound and gravel armor. The sand mound was precisely dimensioned and compacted to guarantee the stability of the caisson system. The caisson was placed on top of the mound, featuring shear keys to ensure precise alignment and secure interlocking. To further reinforce the structural integrity, a 100 mm layer of gravel was applied on top of the sand mound.

Accelerometers (model PCB 393B04) (i.e., ACC x, ACC y, ACC z) were installed on top of the caisson via steel cubes to measure vibrations along the x, y, and z directions. In this setup, two sensors were placed in the x-direction (labeled as 2x and 3x), one sensor in the y-direction (labeled as 2y), and two sensors in the z-direction (labeled as 1z and 2z).

The data acquisition system comprised a signal conditioner (model 481A, PCB Piezotronics, New York, United States), a terminal block (BNC 2090A), a DAQ card (model 6036E), and a laptop. A steel block suspended from a 1300 mm rope was swung freely from a distance of 400 mm to simulate the impact load on the caisson. The time-history responses were collected in 15 ensembles. The obtained time-history responses were then concatenated according to the previously described methodology.

4.2. Foundation-Damage Scenarios

The foundation damage was simulated to mimic a variety of damage scenarios caused by scouring during extreme storm surge events. Figure 9 shows eight levels of foundation damage simulated by sequentially removing various amounts of gravel armor and sand from the foundation mound.

As shown in Figure 9a, the cut-off width was set to 60 cm (i.e., half of the caisson width). Damage 1 to Damage 4 were introduced by removing the depth of the mound up to 10 cm, 20 cm, 30 cm, and 35 cm, respectively. Damage 1 removed 0.37 kN of gravel, which was approximately 2.3% of the armor gravel. Damage 2 removed a total of 0.90 kN material from foundation. It involved 0.46 kN of gravel (about 2.8% of the armor gravel) and 0.44 kN of sand (about 0.7% of the sand mound). Damage 3 resulted in the removal of 0.56 kN gravel (about 3.4% of the armor gravel) and 0.91 kN sand (about 1.5% of the sand mound). Damage 4 removed another 50 mm layer, bringing the total to 0.58 kN of gravel (about 3.6% of the armor gravel) and 1.14 kN of sand (about 1.9% of the sand mound).

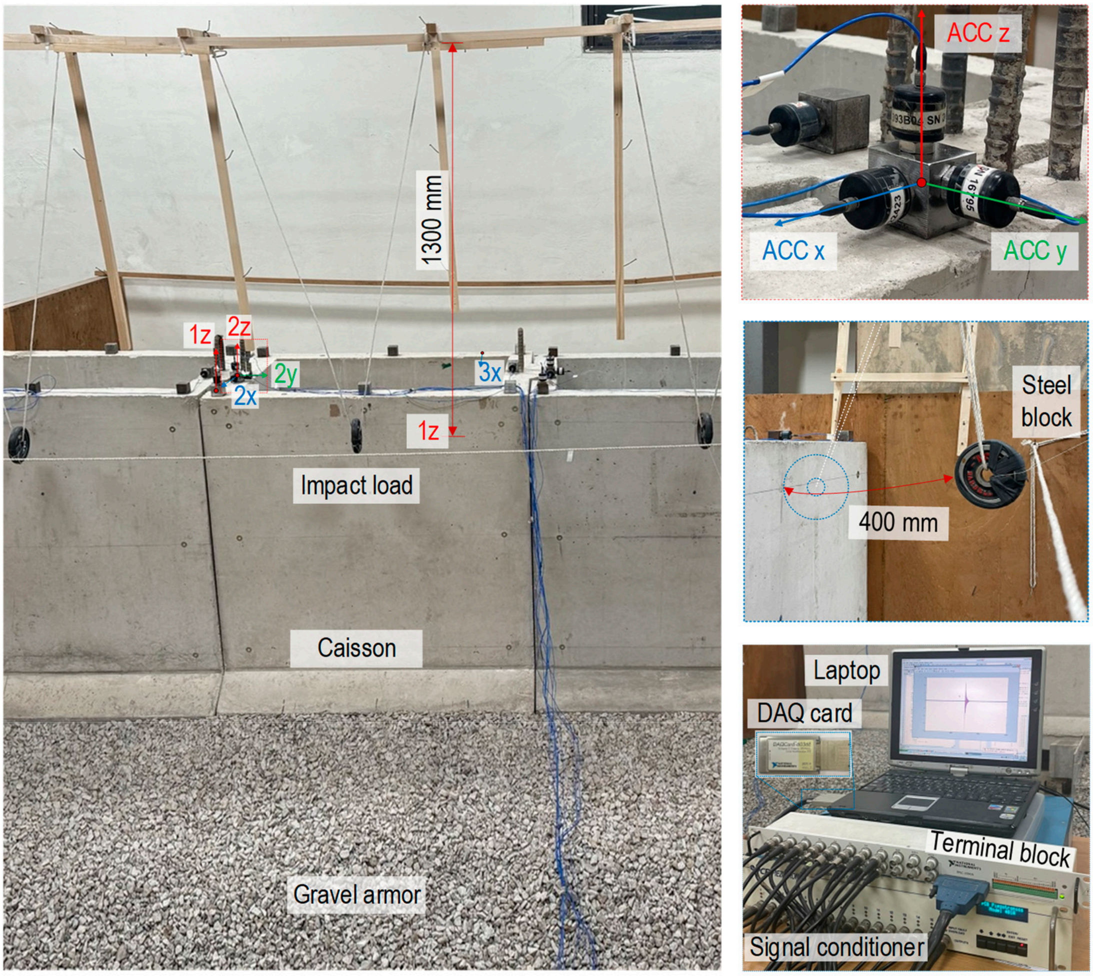


Figure 8. Experimental setup of the lab-scale caisson-foundation system.

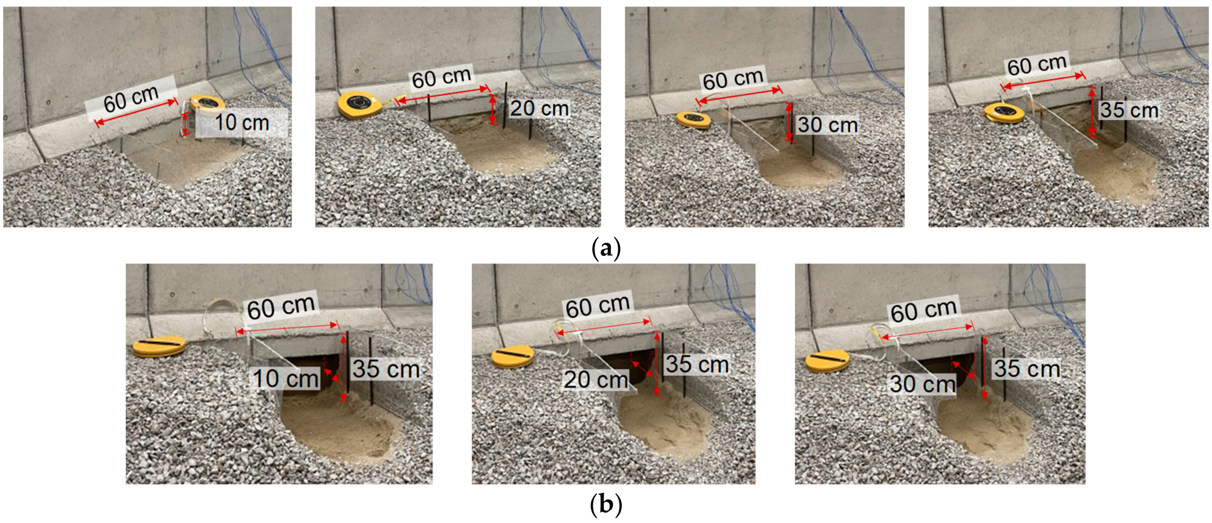


Figure 9. Foundation damage scenarios: (**a**) Damage 1–Damage 4; (**b**) Damage 5–Damage 7.

As shown in Figure 9b, Damage 5 to Damage 7 expanded the damage to foundation–caisson interface with dug hole depths ranging from 100 mm to 300 mm. This resulted in total extracted sand weights of 1.35 kN, 1.56 kN, and 1.77 kN (about 2.2%, 2.5%, and 2.9% of the sand mound), while the removed gravel remained constant at 0.58 kN. The configuration of the foundation damage scenarios is summarized in Table 2.

Table 2. Configuration of foundation damage scenarios.

Case	Gravel Removed (% of Gravel Armor)	Sand Removed (% of Sand Mound)	Descriptions
D0	-	-	Undamaged intact state
D1	0.37 kN (2.3%)	-	Damage in front slope
D2	0.46 kN (2.8%)	0.44 kN (0.7%)	
D3	0.56 kN (3.4%)	0.91 kN (1.5%)	
D4	0.58 kN (3.6%)	1.14 kN (1.9%)	
D5	0.58 kN (3.6%)	1.35 kN (2.2%)	Damage expanded to foundation–caisson interface
D6	0.58 kN (3.6%)	1.56 kN (2.5%)	
D7	0.58 kN (3.6%)	1.77 kN (2.9%)	

4.3. Vibration Data Acquired from Accelerometers

The time-history responses were acquired at a sampling rate of 1.0 kHz. For each impact load, the time-history responses of five sensors (i.e., ACC1z, ACC 2z, ACC 2x, ACC 3x, and ACC 2y) were recorded simultaneously. The observation period was set to 1 s, resulting in 1000 data points per response (90 data points before and 910 data points after the impact load). Comparing the five accelerometers on each caisson, the acceleration magnitude in the z-direction was smaller than in the x- and y-directions. The time-history responses from the five sensors were then concatenated into a single response of 5000 data points. The representative time-history responses of five accelerometer sensors in eight foundation damage levels (D0–D7) are plotted in Figure 10.

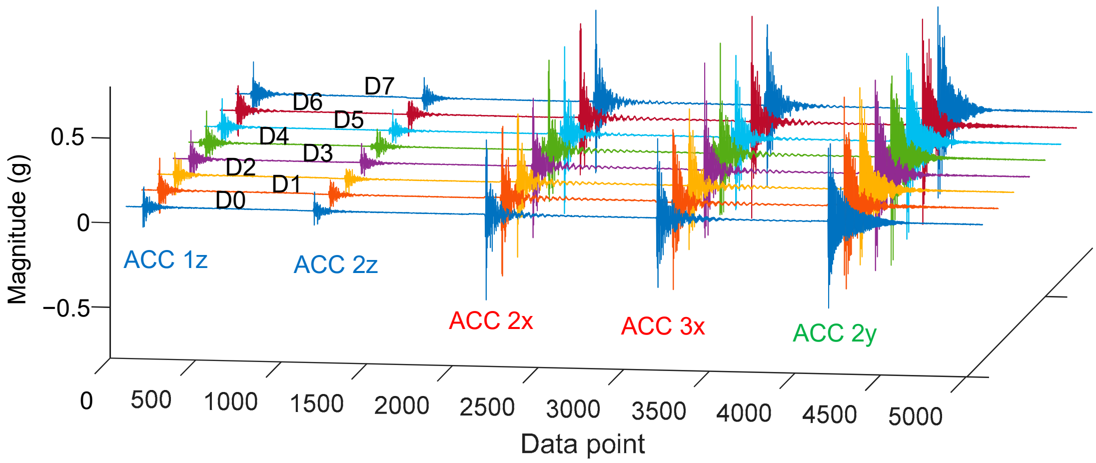


Figure 10. Time-history responses of five accelerometer sensors.

The FFD method was used to extract the SVD from the time-history responses. Figure 11 plots the SVD responses calculated simultaneously from all five accelerometers (i.e., 1z, 2z, 2x, 3x, and 2y). The SVD responses were relatively uniform under all foundation damage levels (D0–D7). Three peaks were identified to represent the caisson's behavior: the first peak frequencies ranged from 25.9 to 27.3 Hz, the second from 183.6 to 186.8 Hz, and the third from 279.8 to 281.2 Hz.

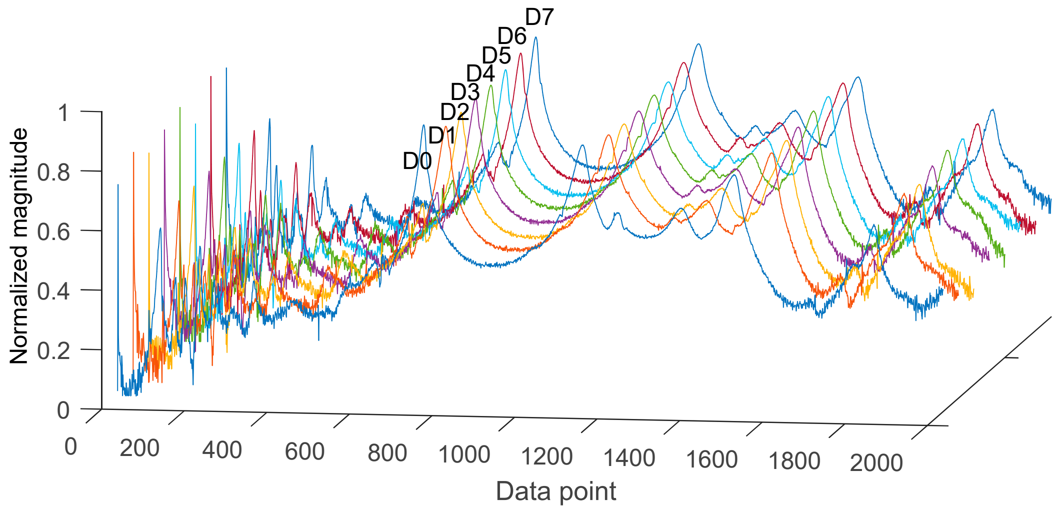


Figure 11. SVD responses of five accelerometer sensors.

5. Development of 1-D CNN Models

5.1. 1-D CNN Model Using Time-History Response

5.1.1. Databank of Time-History Responses

The concatenated time-history responses were processed with controlled noise contamination to improve the CNN model's adaptability to real-world conditions. Noise was introduced at 1–10% standard deviation levels across ten iterations. Figure 12 illustrates examples of contaminated responses compared to their original responses.

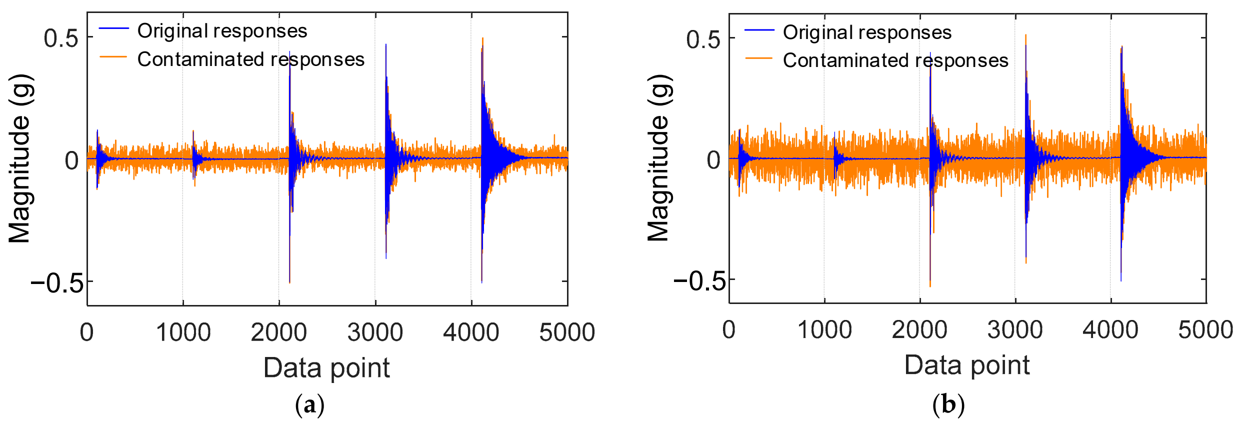


Figure 12. Illustration of noise contamination to time-history responses: (**a**) 5% noise; (**b**) 10% noise.

The time-history responses were used to generate datasets for CNN model development. The responses were acquired in 15 ensembles for each damage level, of which 80% were used for training and validation and 20% reserved for evaluation. A total of 120 responses obtained from eight levels (D0–D7) were divided into 96 responses for training/validation and 24 responses for evaluation. The noise-contamination process generated 4800 responses from the 96 original responses, resulting in a dataset of 4896 responses for training and validation. The evaluation dataset was constructed using the same method to maintain the consistency of the databank structure. The evaluation dataset had 1224 responses, consisting of 24 original responses and 1200 responses generated through noise contamination.

Figure 13 shows the 4896 responses from the foundation damage levels (D0–D7) used for training and validation. Each response consisted of 5000 data points with magnitudes ranging from –0.6 to 0.6. The time-history responses from eight damage levels were labeled accordingly before input into the 1-D CNN model.

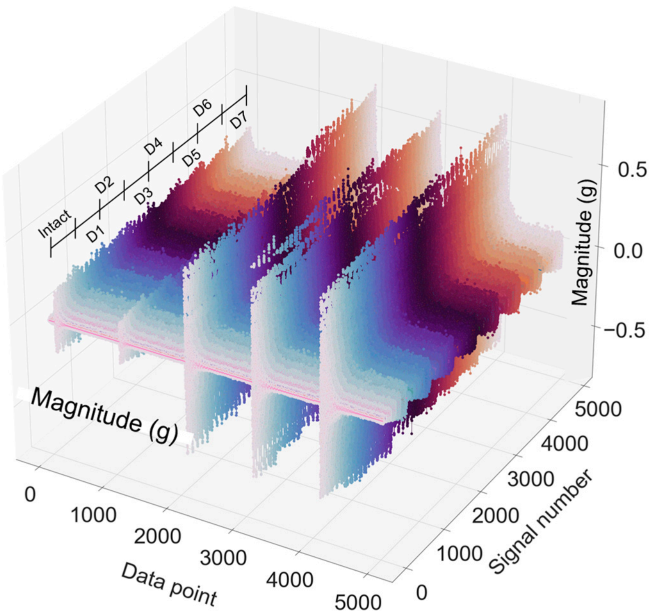


Figure 13. Visualization of time-history responses for training and evaluation of CNN model.

5.1.2. Training Procedures of Time-History Responses

Figure 14 shows the loss values of the CNN model trained by time-history responses. Training loss decreased in the first five iterations and then converged until the end of training. Validation losses sharply decreased in the initial five iterations and maintained stable status until the end at 40th iteration. In general, the training and validation losses performed well and converged after only a few epochs.

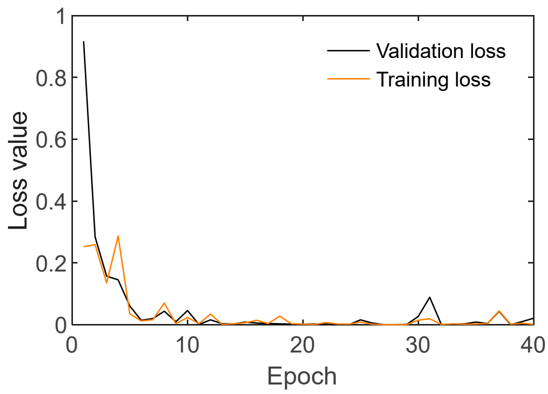


Figure 14. Loss values of the 1-D CNN model trained by time-history responses.

5.1.3. Evaluation of 1-D CNN Model Using Time-History Responses

The 1-D CNN model was evaluated using 20% of the time-history responses from the eight damage levels. As shown in Figure 15, confusion matrices were used to quantify the classification performance. The elements on the diagonal positions represent correct classifications (true positives, TP). The off-diagonal elements indicate incorrect classifications (false negatives, FN, and false positives, FP). True positive rate (TPR, so-called 'recall') was calculated as TPR = TP/(TP + FN), while false negative rate (FNR) was calculated as FNR = FN/(FN + TP). Precision was calculated as Precision = TP/(TP + FP), and accuracy was calculated as Accuracy = TP/(TP + FP + FN). F1-score was calculated as F1 = (2 × Precision × Recall)/(Precision + Recall).

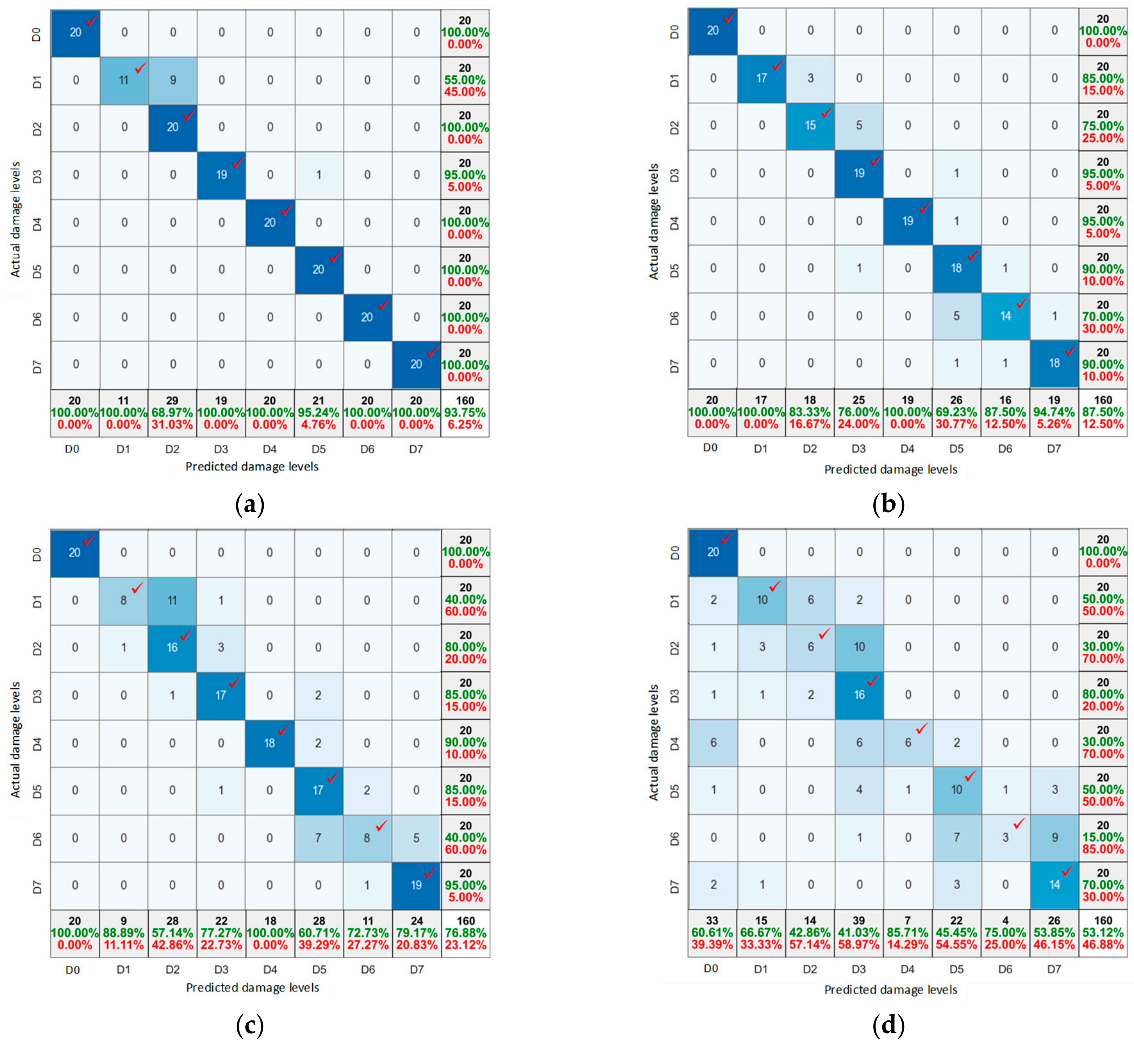


Figure 15. Damage classification results of the 1-D CNN model using the time-history databank: (**a**) trained 1% noise; (**b**) trained 5% noise; (**c**) untrained 6% noise; (**d**) untrained 10% noise. The red tick (✓) denotes the correct level of inflicted damage.

Figure 15a plots the damage classification results of the CNN model using the time-history databank for a trained noise level of 1%. The red tick (✓) denotes the correct level of inflicted damage. All data were classified correctly except in damage levels D1 (nine misclassifications) and D3 (one misclassifications), resulting in an overall accuracy of 93.75%. Figure 15b plots the damage classification results of the CNN model for a trained noise level of 5%. There were one to six misclassifications in each damage level, and the accuracy of the classification was reduced to 87.50%; nevertheless, the CNN model's reliability was demonstrated, with the majority of classifications being correct.

Figure 15c shows the damage classification results of the CNN model using time-history databank for an untrained noise level of 6%. Misclassifications appeared in all damage levels except for the intact level (D0). Noticeably, more than half of the data in damage levels D1 and D6 were misclassified. The overall accuracy of classification decreased to 76.88%. For an untrained noise level of 10%, more than half of the data in several damage levels were misclassified, as shown in Figure 15d. The accuracy of the results dropped to 53.12%. Despite this, most data fell in the nearby neighbor classes of the correct ones, suggesting that the CNN model can handle high noise levels with an acceptable error. However, there is a need to develop a technique that can minimize errors and improve the accuracy of the classification.

Figure 16 summarizes the accuracy metrics of the CNN model using time-history databank under the effect of noise. Accuracy, false positive rate (FPR), false negative rate (FNR), and F1-score were chosen to be analyzed (see Figure 16a–d). A better damage classification result was indicated by higher accuracy and F1-score value, and lower FPR and FNR. The accuracy values of the CNN model corresponding to noise levels of 0–5% ranged from 94–87%, followed by a constant decrease until they reached 53% at a noise level of 10%. The FPR value ranged from 4–11% for trained noise levels (0–5%) and abruptly increased to 20% at a noise level of 6%. The FPR continued to rise, reaching 41% at a noise level of 10%. Similarly, the FNR value was below 12.5% for noise levels 0–5% and then significantly increased for noise levels of 6–10%, with a maximum error of 47% at a noise level of 10%. The F1-score remained around 0.92 for trained noise levels 1–5%, then constantly decreased in untrained noise levels until reaching 0.79 at a noise level of 10%.

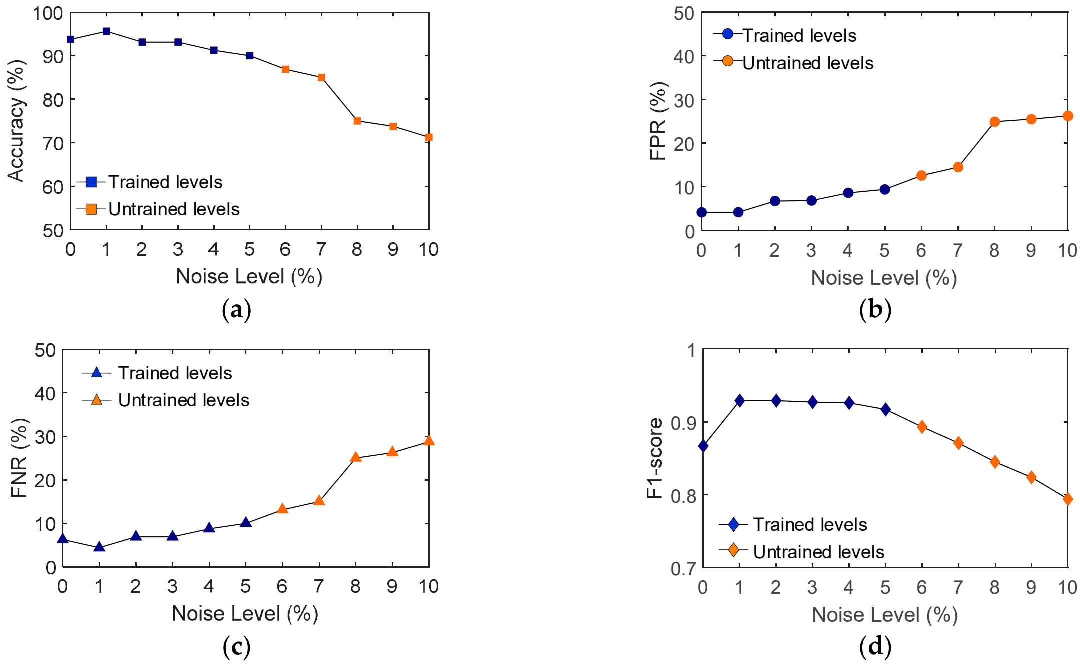


Figure 16. Noise effect on the 1-D CNN model trained by time-history data: (**a**) accuracy; (**b**) false positive rate; (**c**) false negative rate; (**d**) F1-score.

The CNN model trained by time-history response data demonstrated its feasibility in damage detection and monitoring. It maintained good performance with high noise levels (up to 5%). However, the model showed limitations when confronting untrained noise levels (6–10%).

5.2. 1-D CNN Model Using SVD Responses

5.2.1. Databank of SVD Responses

Figure 17 shows examples of contaminated SVD responses compared to their original ones. The training, validation, and evaluation datasets for the CNN model were generated from 120 SVD responses obtained from eight damage levels (D0–D7), for which 80% of the data (96 responses) were used for training and validation, and 20% of the data (24 responses) were used for evaluation. Noise was introduced at 1–10% standard deviation levels to SVD responses across ten generating iterations. After the noise contamination process, the training and validation datasets consisted of 4896 data, and the evaluation dataset consisted of 1224 data.

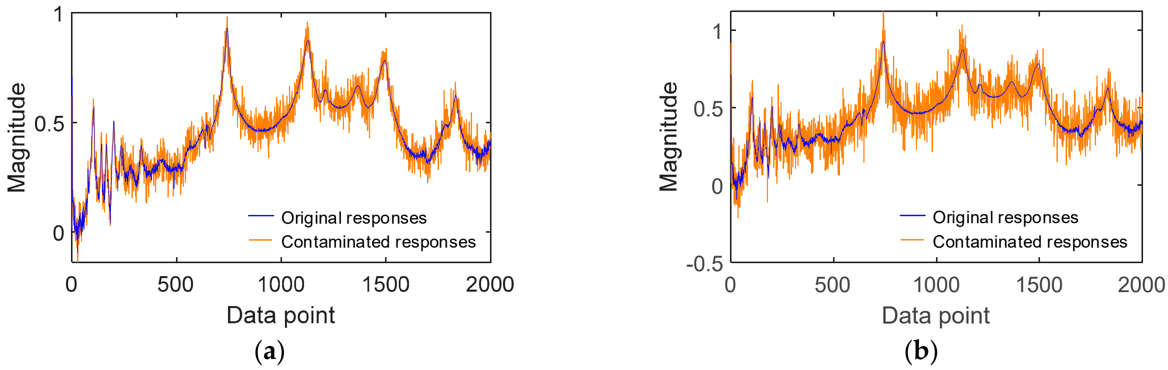


Figure 17. Illustration of noise contamination to SVD responses: (**a**) 5% noise; (**b**) 10% noise.

Figure 18 visualizes 4896 data points in training and validation datasets for eight foundation damage levels (D0–D7). Each response comprises 5000 data points ranging in magnitude from 0 to 1. These SVD responses were labeled accordingly before input into the 1-D CNN model.

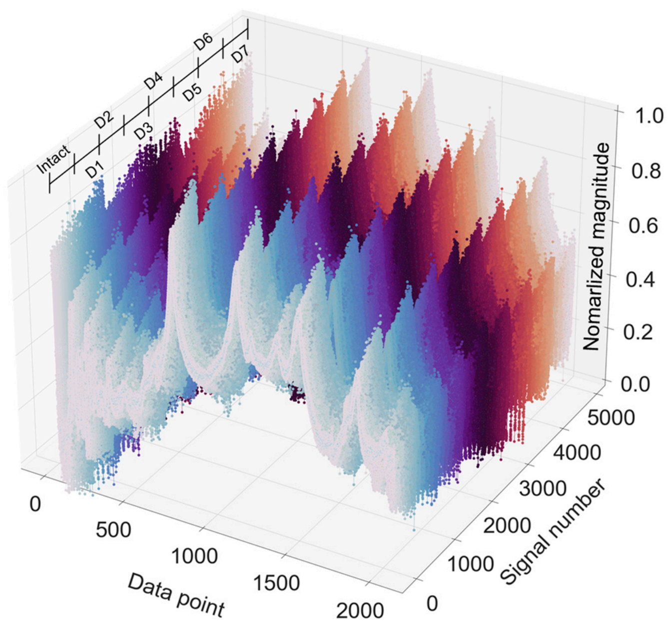


Figure 18. Visualization of SVD responses for training and evaluation of the CNN model.

5.2.2. Training Procedures of SVD Responses

Figure 19 shows loss values of the CNN model trained by SVD responses. Both the training and validation losses consistently decreased over 40 iterations and converged to nearly zero by the end of training. The CNN model designed with a compact architecture demonstrated efficient convergence after only a few epochs.

5.2.3. Evaluation of the 1-D CNN Model Using SVD Responses

Figure 20a plots the damage classification results of the CNN model using the SVD databank for the trained noise level of 1%. There were five misclassifications in damage level D3 and two misclassifications in damage level D4. The accuracy of classification reached 95.62%. As the noise level increased up to 5% (Figure 20b), the accuracy decreased to 90%, with one to nine misclassifications in damage levels D1–D4.

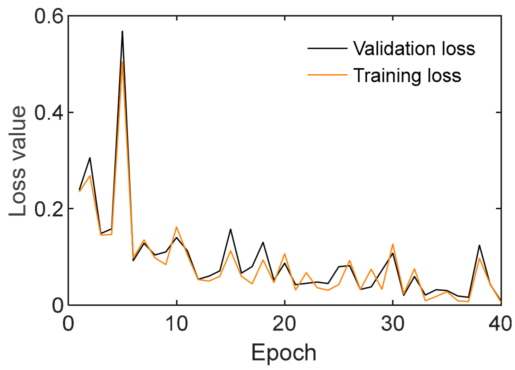


Figure 19. Loss values of the 1-D CNN model trained by SVD responses.

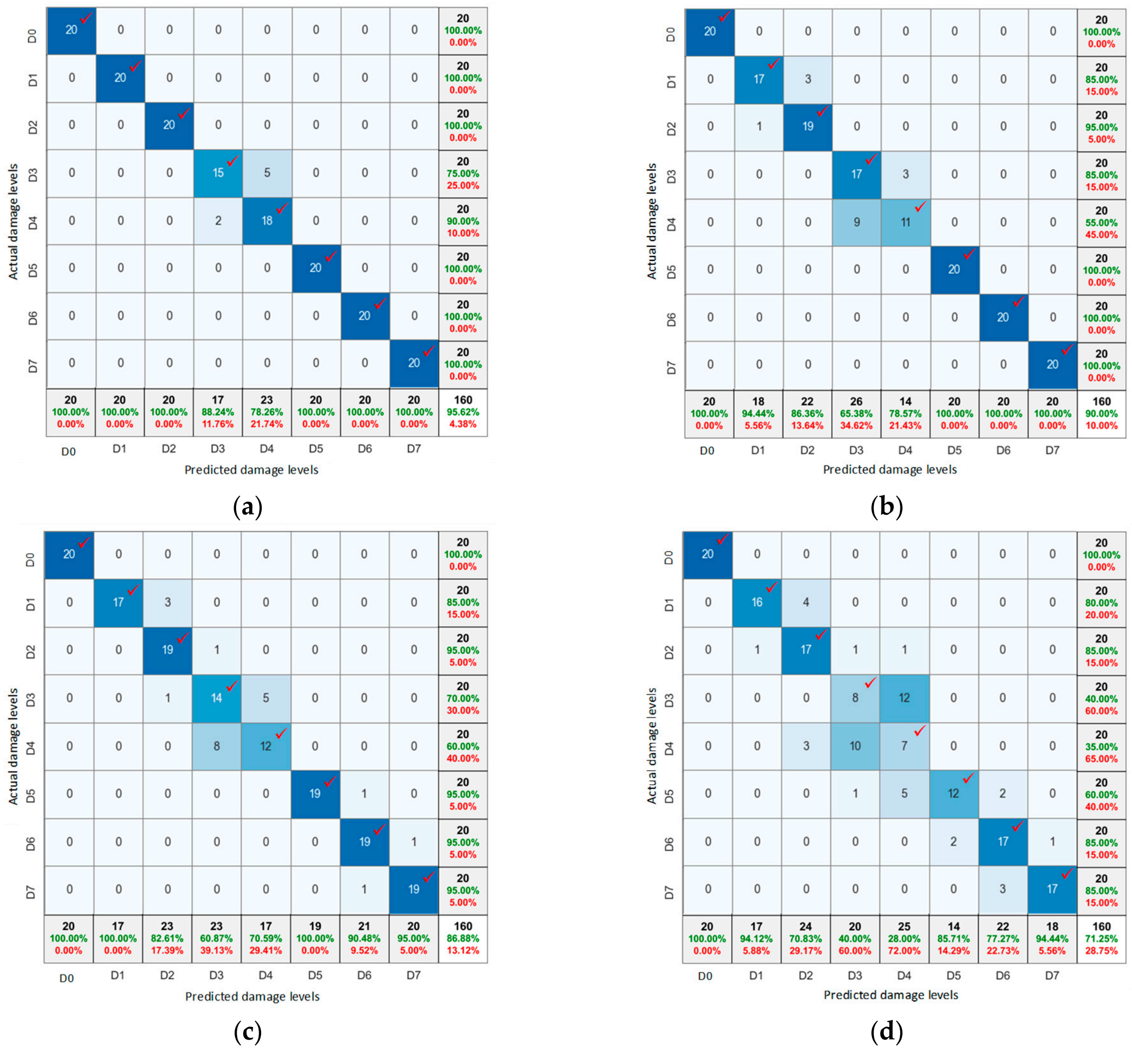


Figure 20. Damage classification results of the 1-D CNN model using SVD databank: (**a**) trained 1% noise; (**b**) trained 5% noise; (**c**) untrained 6% noise; (**d**) untrained 10% noise. The red tick (✓) denotes the correct level of inflicted damage.

Figure 20c shows the damage classification results of the CNN model using the SVD databank for the untrained noise level of 6%. Misclassifications appeared at all damage levels except for the intact level (D0). The overall accuracy of classification was 86.88%. As the noise level increased to 10% (see Figure 20d), the accuracy decreased to 71.25%. There were fewer than five misclassifications out of 20 data for each damage level, except for damage levels D3 and D4. The number of misclassifications was significantly smaller than the model trained using time-history responses at the same noise level.

Figure 21 summarizes the accuracy metrics of the CNN model trained using SVD responses under the effect of noise. Accuracy, false positive rate, false negative rate, and F1-score were analyzed (see Figure 21a–d), and the results were compared to the accuracy metrics of the CNN model trained using time-history responses.

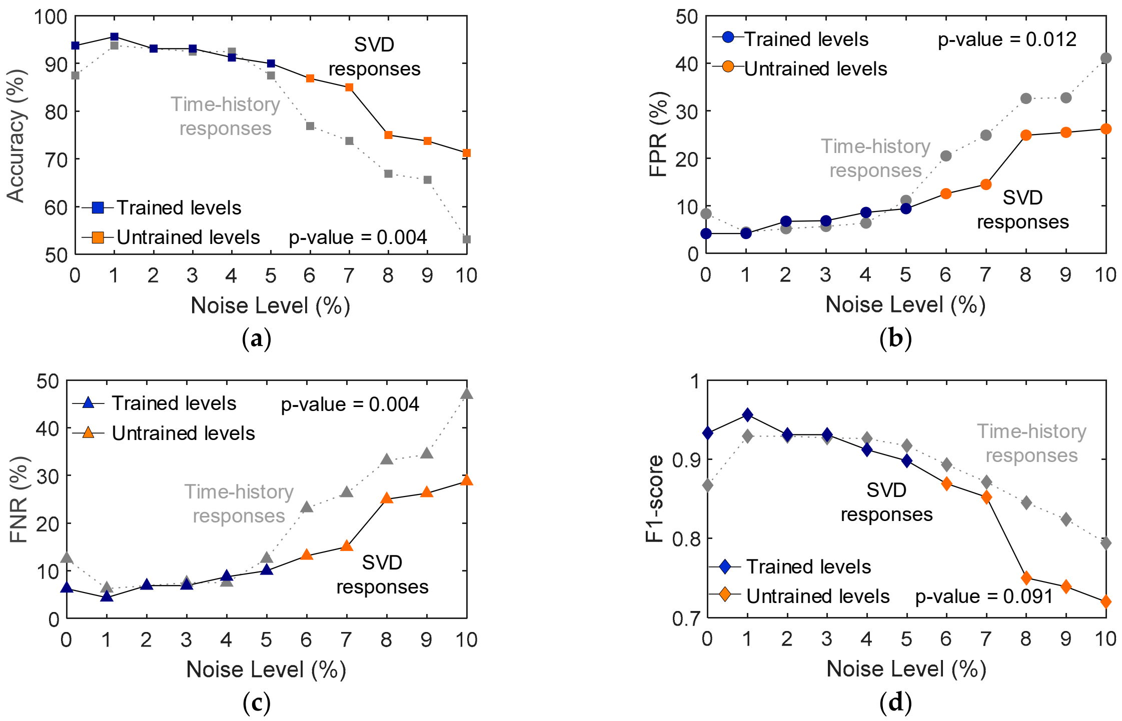


Figure 21. Noise effect on the 1-D CNN model trained using SVD responses compared to the model trained using time-history responses (grey lines): (**a**) accuracy; (**b**) false positive rate; (**c**) false negative rate; (**d**) F1-score.

The accuracy of the CNN model trained by SVD responses ranged from 91% to 96% for noise levels 0–5%. It slightly outperformed the CNN model trained using time-history responses, but the difference in accuracy between the two models was not significant. The performance divergence became clearer at higher noise levels (6–10%). At a noise level of 10%, the accuracy of the CNN model trained using SVD responses decreased to 71% compared to 53% using the time-history responses, suggesting that the CNN model trained with SVD responses exhibited improved robustness against untrained noise levels.

For trained noise levels (0–5%), FPR values ranged from 4% to 11% and increased significantly to 20% at a noise level of 6% and continued to rise to 41% at a noise level of 10%. Similarly, FNR values remained below 12.5% for noise levels 0–5%, but increased significantly to a maximum of 47% at noise level 10%.

In Figure 21d, the F1-score of the CNN model trained using SVD responses was above 0.9 for trained noise levels 0–5%. The F1-score rapidly declined for the untrained noise levels and dropped to 0.72 at a noise level of 10%. The F1-score indicated that the performance is not as strong as the CNN model trained using time-history responses.

The *p*-value of the one-tailed t-test [24] was employed to evaluate the difference in performance of two approaches. A low *p*-value (typically ≤ 0.05) indicates that the differences in performance are statistically significant. In Figure 21d, the *p*-value of the F1-score was greater than 0.05, indicating that the performance comparison based on the F1-score was not statistically significant. As shown in Figure 21a–c, the *p*-values of accuracy, FPR, and FNR were less than 0.05, indicating that using SVD data improved the CNN performance significantly.

In summary, the CNN model trained with SVD responses demonstrated its efficacy in damage detection and monitoring. It maintained good performance even with noise levels up to 5%. This matched the performance of the CNN model trained with time-history responses. However, the model trained with SVD responses demonstrated better performance when dealing with untrained noise levels (6–10%).

The computations were performed on a desktop computer (CPU—Intel Core i7-10700F 2.9 GHz). The training times for the CNN model using time-history and SVD data were 7 and 4 min, respectively. Both approaches demonstrate computational efficiency that is suitable for real-time monitoring applications. With the relatively short training times, models can be quickly updated or retrained when new data are available.

6. Evaluation of 1-D CNN Models for Untrained Damage Cases

6.1. Damage Classification by 1-D CNN Model Using Time-History Responses

Three untrained damage cases were established, as outlined in Table 3. To create partially untrained data scenarios, certain damage levels were excluded from the training and validation datasets. For Case 1, the damage levels D2, D4, and D6 were excluded from among the eight damage levels (D0–D7), resulting in 3060 responses remaining in the training and validation datasets. For Case 2, the training and validation datasets excluded four damage levels (D1, D3, D5, and D7). Case 3 excluded data from damage levels D1, D2, D5, and D6, leaving 2448 responses in the training and validation datasets.

Table 3. Untrained damage cases for evaluating the 1-D CNN model.

Case	Scenario	Data Type	
		Training & Validation Datasets	Evaluation Dataset
1	Excluding damage levels D2, D4, and D6	3060	1224
2	Excluding damage levels D1, D3, D5, and D7	2448	
3	Excluding damage levels D1, D2, D5, and D6	2448	

Figure 22 shows the loss values of the CNN models using partially untrained time-history databanks. For all three cases, the training and validation losses decreased sharply in a few initial iterations and converged by the end of the 40 iterations. The training and validation losses demonstrated good performance of all three CNN models.

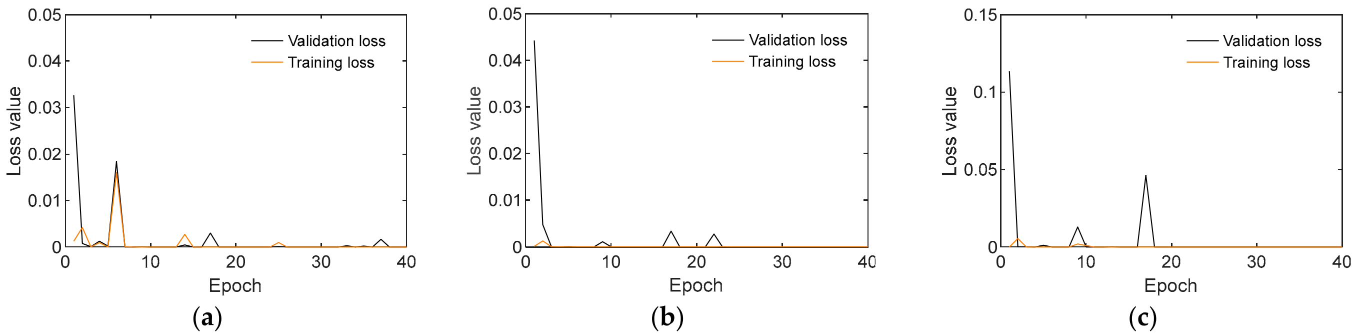


Figure 22. Loss values of the 1-D CNN model using partially untrained time-history databanks: (**a**) Case 1; (**b**) Case 2; (**c**) Case 3.

Figure 23 shows the damage classification results of the CNN model using partially untrained time-history databanks. The red ticks indicate the correct level of the inflicted damage. With untrained cases, the CNN model was limited in accessibility to certain damage levels. When encountering an unseen damage level, the CNN model could not directly classify data to untrained levels. Instead, it was expected to return the classification results in the most similar trained damage levels (the green ticks).

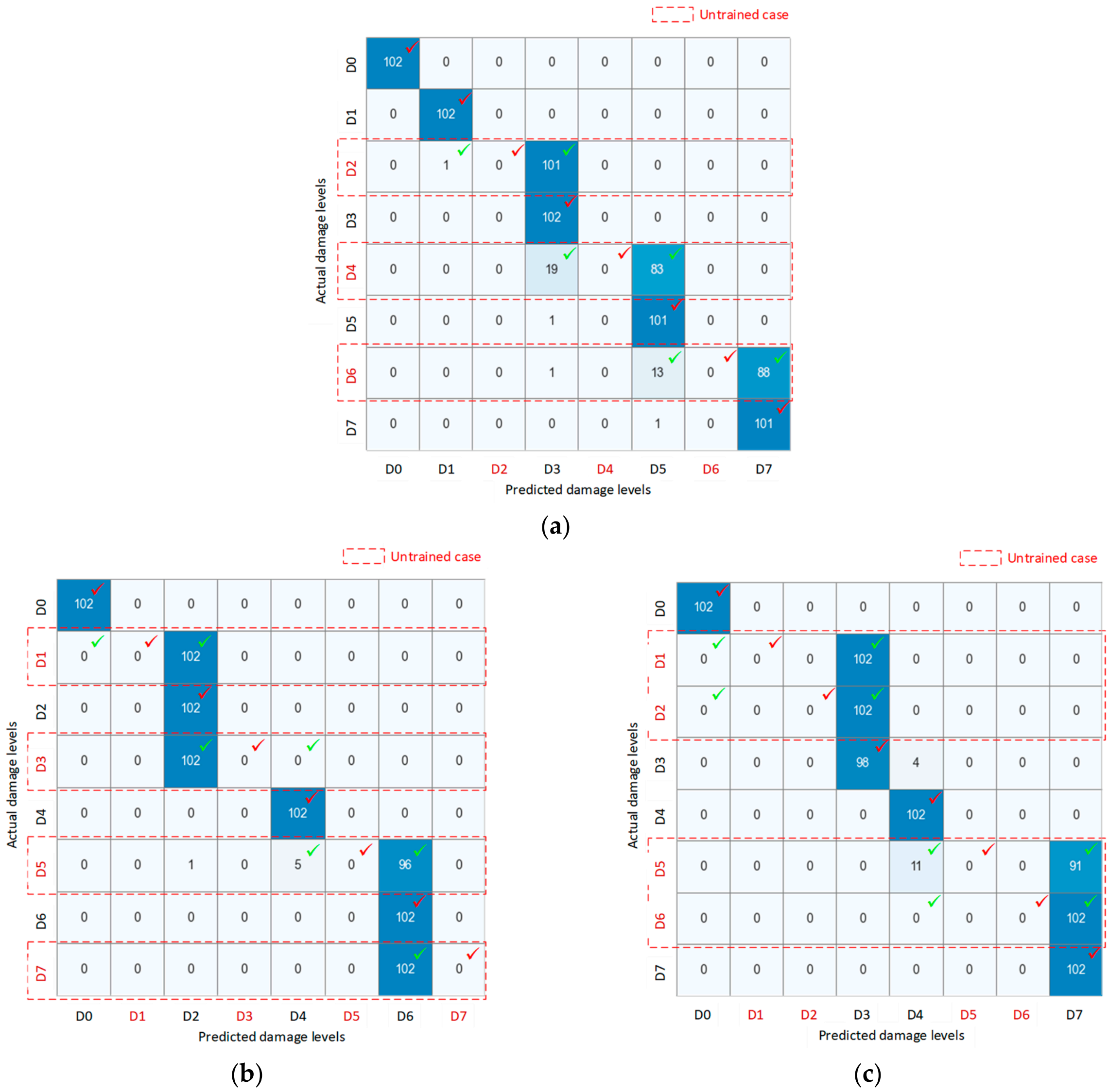


Figure 23. Damage classification results of the 1-D CNN model using partially untrained time-history databank: (**a**) Case 1; (**b**) Case 2; (**c**) Case 3. The red ticks indicate the correct level of the inflicted damage. When the CNN model could not directly classify data to untrained levels, it returned the classification results in the most similar trained damage levels (green ticks).

Figure 23a shows the damage classification results for the untrained Case 1. The data in all trained damage levels wasaccurate, with no misclassifications. The untrained damage levels were classified to nearby ones: D2 was classified to D3; D4 was classified to D3 and D5; D6 was classified to D5 and D7. Figure 23b shows the damage classification results for the untrained Case 2. There was no misclassification in the classification results for trained levels (i.e., D0, D2, D4, and D6). The untrained damage levels were classified into nearby ones: data from D1 and D3 were classified as D2; and data from D5 and D7 were classified as D6. Figure 23c shows the untrained Case 3. The distance between trained damage levels increased to two levels when D1, D2, D5, and D6 were neglected. The CNN model maintained its ability to make correct classifications for trained damage levels. The untrained damage levels D1 and D2 were classified as the nearest damage level D3, while untrained damage levels D5 and D6 were classified as the nearest damage level D7.

In practice, only a finite number of damage levels can be provided for model training. The results with partially untrained databank demonstrated the CNN model's ability to classify unseen damage levels into nearly similar damage levels. This reduced the need to provide many training cases, making the CNN model feasible for real applications in caisson systems.

6.2. Damage Classification by 1-D CNN Models Using SVD Responses

For SVD responses, three partially untrained datasets were generated similarly to time-history responses as shown in Table 3. Figure 24 shows the loss values of the CNN model trained using partially untrained SVD databanks. In both Case 1 and Case 2, the training and validation losses exhibited a sharp decrease in the first ten iterations, followed by convergence to near-zero values by the 40th iteration. For Case 3, the training and validation losses showed a slower initial decline over the first 20 iterations, followed by minor fluctuations until the end of training.

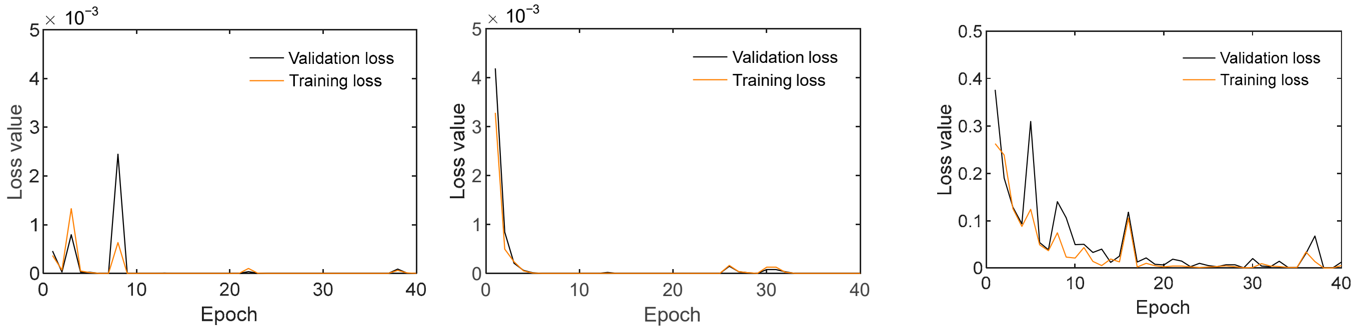


Figure 24. Loss values of the 1-D CNN model using partially untrained SVD databanks: (**a**) Case 1; (**b**) Case 2; (**c**) Case 3.

Figure 25 shows the damage classification results of the CNN model with partially untrained SVD databanks. For Case 1 (Figure 25a), all results for trained levels showed no misclassification. The damage levels D2, D4, and D6 were classified to nearby levels D1, D3, and D7, respectively. Case 2 (Figure 25b) demonstrated no misclassification in the results for trained levels (i.e., D0, D2, D4, and D6). The untrained damage levels D1, D3, and D7 were classified into nearby levels D2, D4, and D6, respectively. The untrained damage level D5 was classified into D4 and D6.

In Case 3 (Figure 25c), results for trained levels D0 and D7 showed no misclassification, but 50% of the data for D3 and D4 was misclassified. The untrained damage levels D1 and D2 were classified to the nearest damage level D3; untrained damage level D5 was classified to the nearest damage level D4; and D6 was classified to the nearest damage level D7.

6.3. Discussion on Damage Classification Results

In Figure 26, the damage classification results of CNN models (shown in Figures 23 and 25) are analyzed for untrained Case 1 using the normal probability density function (PDF) [25]. In Figure 26a, the classification results are depicted in blue (based on time-history data) and red (based on SVD data), while the combined results are shown in green. The shaded region represents the range within one standard deviation (σ) from the mean value (μ) and encompasses 68.8% of the observed values from the central tendency of the prediction.

For the untrained damage D2 in Case 1, the damage classification results (see Figures 23a and 25a) were assessed as shown in the PDF chart in Figure 26a. In the figure, the blue-shaded region indicates damage level D3 based on the time-history data. The red-shaded region indicates damage level D1 based on the SVD data. Combining the two results, the green-shaded region indicates that the damage level was classified to D2.

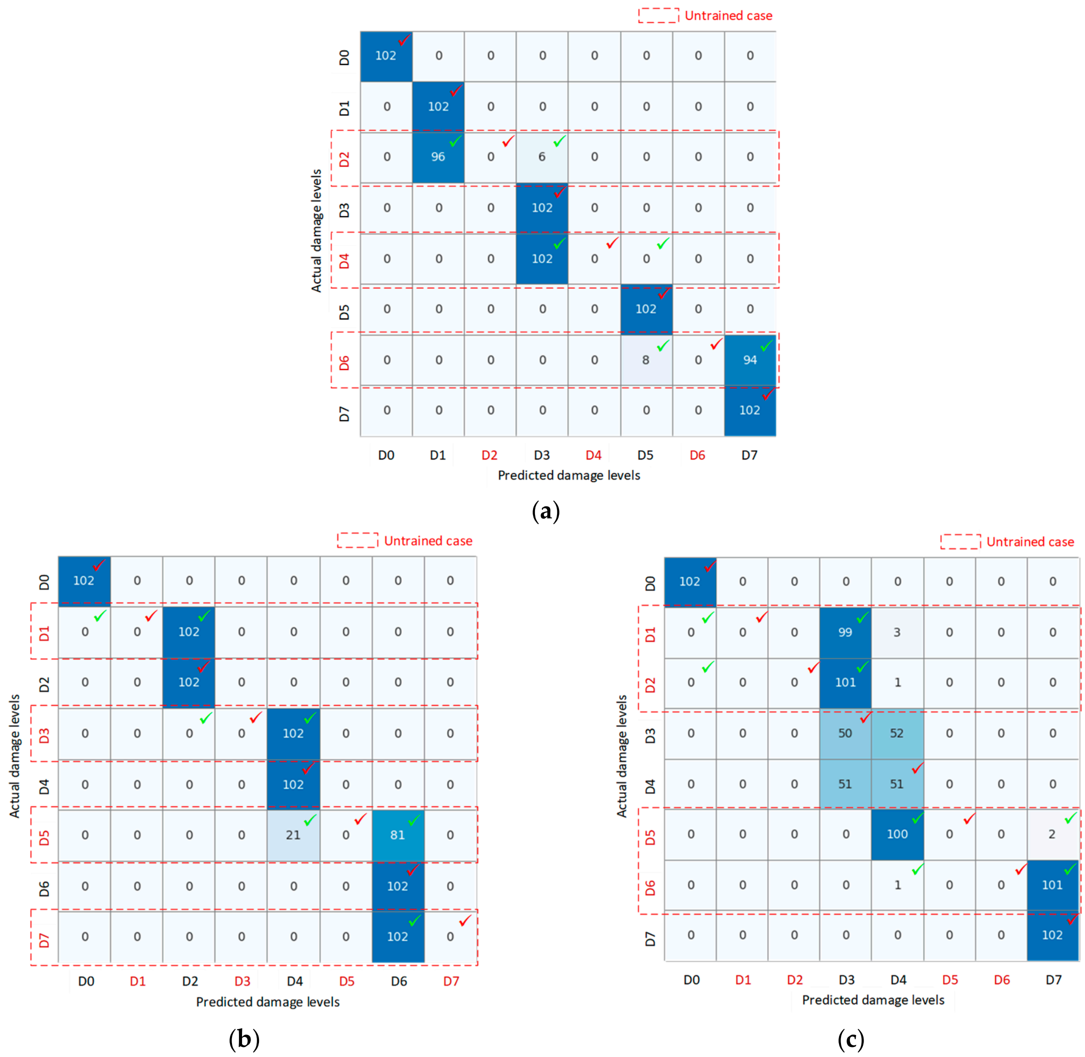


Figure 25. Damage classification results of the 1-D CNN model using partially untrained SVD databanks: (**a**) Case 1; (**b**) Case 2; (**c**) Case 3. The red ticks indicate the correct level of the inflicted damage. When the CNN model could not directly classify data to untrained levels, it returned the classification results in the most similar trained damage levels (green ticks).

For the untrained damage D4 in Case 1 (Figure 26b), the blue-shaded region indicated damage levels D4–D5 based on the time-history data, where the mean value leans towards D5. All classification results based on SVD data indicated damage level D3, hence their normal PDF chart is not shown. The combined results in the green-shaded region indicate that the damage level was classified to D4.

For the untrained damage level D6 in Case 1 (Figure 26c), both blue and red-shaded regions indicate damage level D7. Consequently, the green-shaded region for the combined results indicates that the damage level belonged to D7.

In Figure 27, the damage classification results (see Figures 23b and 25b) are analyzed for Case 2 using the PDF chart. For the untrained damage level D1 in Case 2, all results indicated the nearest trained damage level D3. Similar to untrained damage level D7 in Case 2 (Figure 27d), all results indicated the nearest trained damage level D6. The normal PDF chart is not shown for these cases.

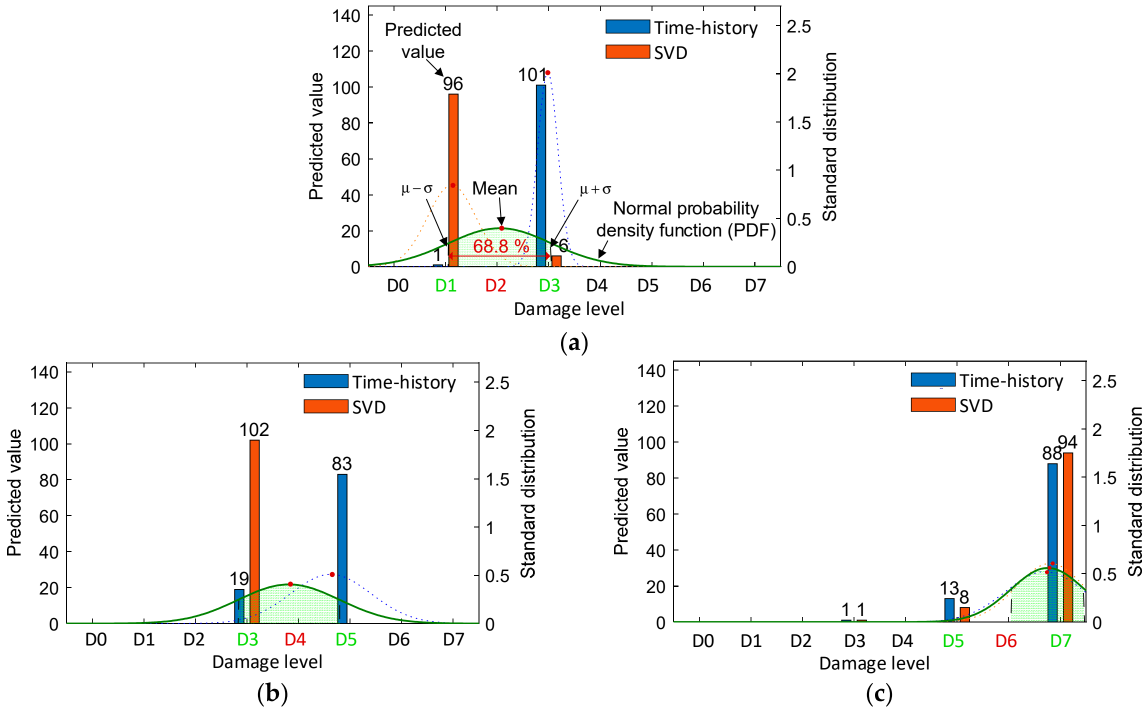


Figure 26. Probability assessment of damage classification results: Case 1: (**a**) untrained D2; (**b**) untrained D4; (**c**) untrained D6. The blue-shaded region shows results based on time-history data, the red-shaded region shows results based on SVD data, and the green-shaded region shows combined results.

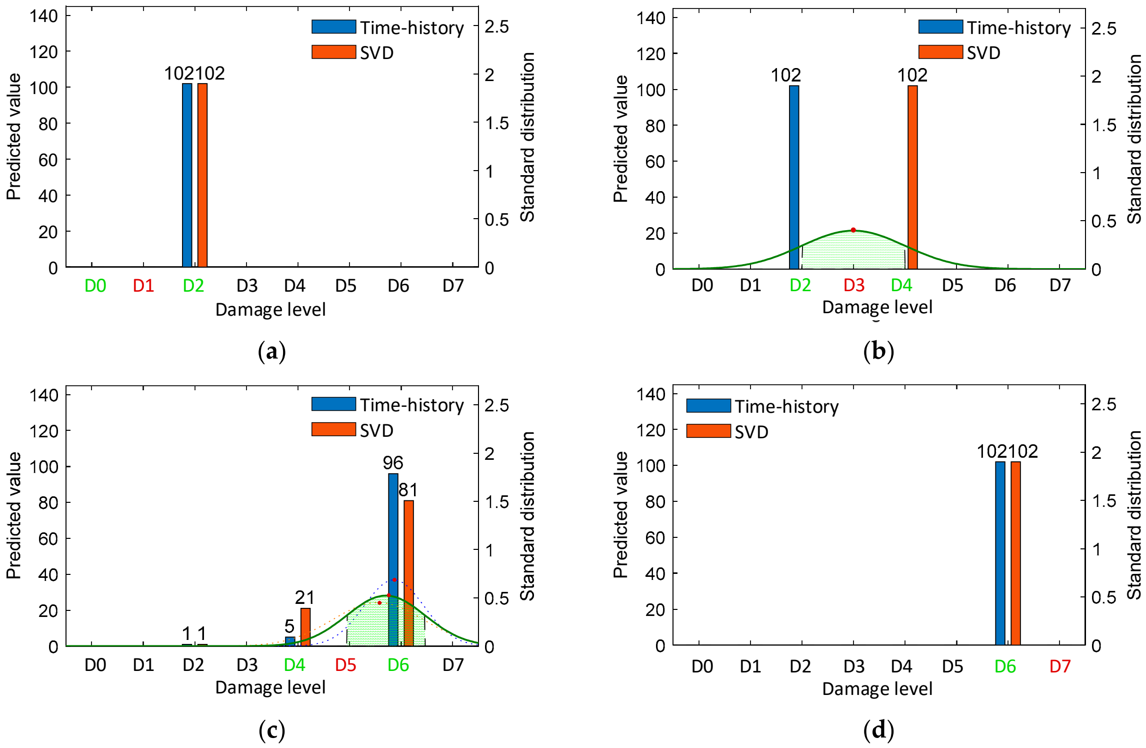


Figure 27. Probability assessment of damage classification results: Case 2: (**a**) untrained D1; (**b**) untrained D3; (**c**) untrained D5; (**d**) untrained D7. The blue-shaded region shows results based on time-history data, the red-shaded region shows results based on SVD data, and the green-shaded region shows combined results.

For the untrained damage level D3 in Case 2 (Figure 27b), the results based on time-history and SVD data indicated damage levels D2 and D4, respectively. Combining the two results, the green-shaded region indicates that the damage level was classified to D3. For the untrained damage level D5 in Case 2 (see Figure 27c), all three shaded regions indicate that the damage level fell between D5 and D6, with the mean value leaning towards D6.

In Figure 28, the damage classification results (see Figures 23c and 25c) are analyzed for Case 3. For the untrained damage levels D1 and D2 in Case 3 (see Figure 28a,b), almost all results based on time-history and SVD data indicated the nearest trained damage level D3. The green-shade region indicates that the damage belonged to D3.

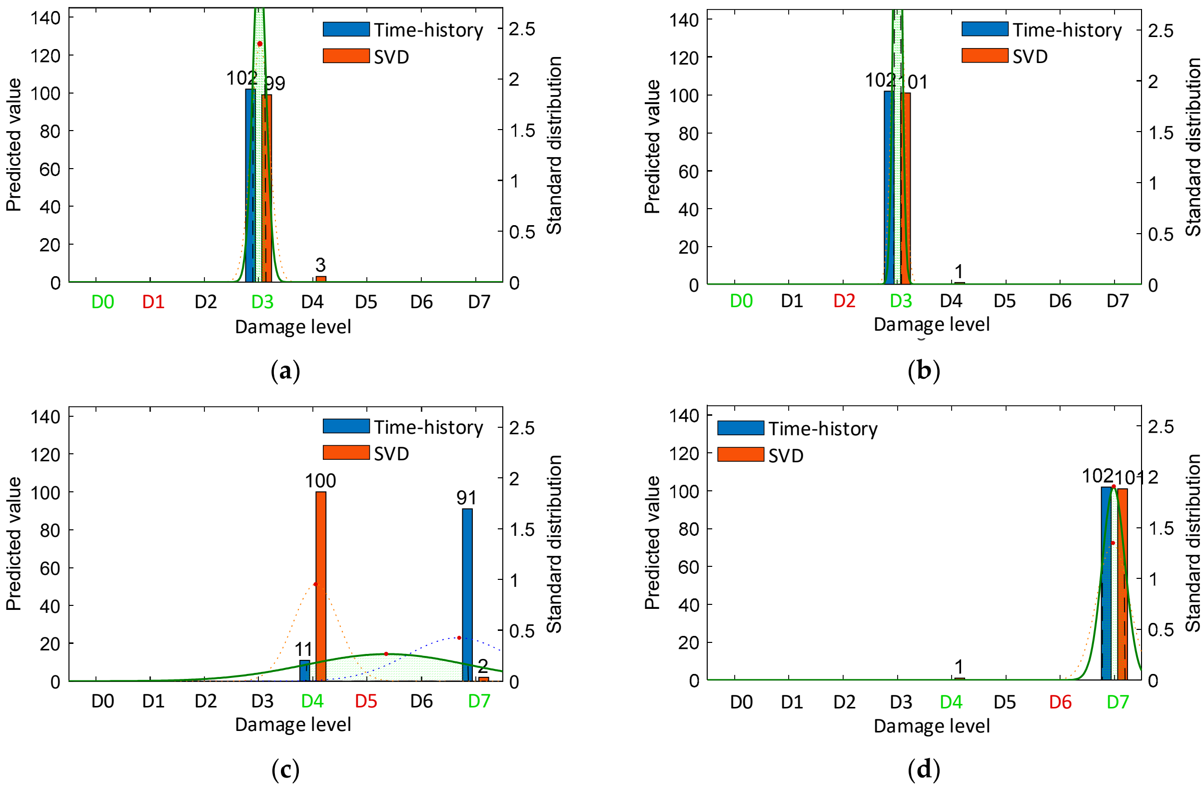


Figure 28. Probability assessment of damage classification results: Case 3: (**a**) untrained D1; (**b**) untrained D2; (**c**) untrained D5; (**d**) untrained D6. The blue-shaded region shows results based on time-history data, the red-shaded region shows results based on SVD data, and the green-shaded region shows combined results.

Figure 28c shows the probability assessment for the untrained damage level D5 in Case 3. In the figure, the blue-shaded region indicates the damage level D7 based on time-history data. The red-shaded region indicates the damage level D4 based on SVD data. Combining the two results, the green-shaded region indicates that the damage level belongs to D5 and D6, with the mean value leaning towards D5. For the untrained damage level D6 in Case 3 (see Figure 28d), all three shaded regions indicate that the damage level belonged to D7.

This visualization highlights the CNN models' performance with partially untrained data and demonstrates their capability to classify untrained damage levels into the most similar trained levels. Additionally, combining results from two approaches using time-history and SVD data could strengthen classification results and provide additional insights for investigators.

Figure 29 compares the accuracy of CNN models trained using time-history and SVD data under three untrained cases. All CNN models achieved high accuracy. For Case 1 and Case 2, the CNN models trained using time-history responses achieved accuracies of 99.6%

and 99.9%, respectively. Meanwhile, the CNN models trained using SVD responses had an accuracy of 100% for both cases. For Case 3, the CNN model trained using time-history data maintained high accuracy at 99.5%. However, the accuracy of the CNN model trained using SVD data fell to 86.9%. Note that the misclassification mainly came from the trained damage levels (i.e., D3 and D4). This comparison reveals that each approach adapted to particular situations. Therefore, using both approaches and integrating their results could provide better insights and enhance overall classification accuracy.

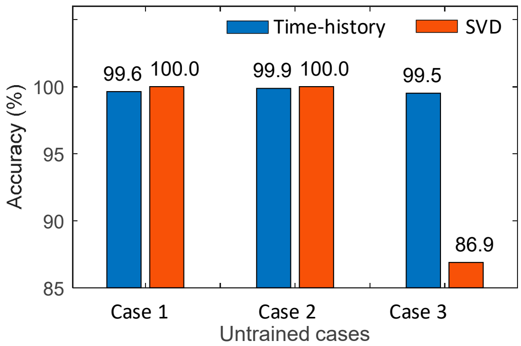


Figure 29. Accuracy of CNN models with untrained cases.

7. Conclusions

This study introduced a novel approach that applies CNN deep learning to both time-history and SVD responses for identifying damage in submerged structure-foundation systems. Firstly, various foundation damage levels were simulated in a lab-scale caisson-foundation system, and corresponding time-history responses were recorded using accelerometers. Secondly, the 1-D CNN deep learning model was trained using time-history responses. Thirdly, the 1-D CNN model was trained using the SVD responses derived from time-history responses. Finally, the trained CNN models were implemented to evaluate the damage within the system's foundation under noise contamination and partially untrained data.

Four main conclusions of the study are summarized as follows:

(1) The CNN deep learning model trained using time-history and SVD responses successfully classified the foundation damage of the caisson system.
(2) The t-test evaluation on performance metrics indicated that the CNN model trained using SVD responses outperformed the CNN model trained using time-history responses, particularly when dealing with untrained noise levels. The findings underscore the effectiveness of using additional vibration features such as SVD data.
(3) The performance of the CNN model was maintained with partially untrained cases. The CNN models trained using time-history and SVD responses successfully classified the untrained damage levels as the most similar trained damage levels. This outcome demonstrates the robustness and potential effectiveness of the CNN model under in situ conditions.
(4) Integrating the time-history and SVD responses strengthened the damage classification results and provided additional insights for investigators.

The proposed approach can be directly integrated into existing monitoring systems as a post-processing and decision-making phase. The CNN deep learning technique requires no additional hardware and only a one-time setup, with the ability to be automatically updated as new data become available. By reducing reliance on manual decision-making, this approach mitigates risks associated with human error and offers long-term benefits by saving expert labor costs. By collaborating with industrial partners, vibration data for training can be collected directly from in situ damaged caissons. The CNN model is continuously updated with variations in the shape, location, and severity of foundation damage. The ongoing refinement and expansion of the dataset enhances the reliability of the CNN model, improving its accuracy and robustness in real-world conditions.

Despite the effectiveness of the developed method for damage detection and monitoring, some aspects need to be resolved. Firstly, the feasibility of the method was proven with experiments on a lab-scaled caisson system. It now needs to be verified with the in situ caisson-foundation systems. Secondly, future investigations should consider different types of noise and the influence of environmental factors. Several other data augmentation techniques such as shifting, amplitude scaling, and window slicing could be applied besides Gaussian noise contamination. Thirdly, the verification of the method should be extended to other machine learning models. Although integrating various vibration features has proven effective for the CNN model, further evaluation is necessary to assess its compatibility with alternative machine learning models. Finally, the investigation showed the dependence of CNN models on the availability of vibration data (i.e., time-history and SVD responses). For real caisson systems, simulating damage to submerged foundations is challenging. Additional research on generating training data through indirect methods such as numerical models and pseudo-damage simulation is needed.

Author Contributions: Conceptualization, J.-T.K. and N.-L.P.; methodology, analysis, writing—original draft, N.-L.P.; data recording, N.-L.P. and Q.-B.T.; investigation and data curation, N.-L.P.; writing—review and editing, J.-T.K.; supervision, J.-T.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Basic Science Research Program through the National Research Foundation of Korea: 2022R1A2C1003889136178206434O1O3.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available on reasonable request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Brunn, P. Port structures, wharves, quays, terminals, and mooring devices. *J. Coast. Res.* **2005**, *SI*, 139–158.
2. Matsui, T.; Oda, K. Foundation damage of structures. *Soils Found.* **1996**, *36*, 189–200. [CrossRef] [PubMed]
3. Oumeraci, H.; Kortenhaus, A. Analysis of the dynamic response of caisson breakwaters. *Coast. Eng.* **1994**, *22*, 159–183. [CrossRef]
4. Yu, C. Numerical simulation of pounding damage to caisson under storm surge. *E3S Web Conf.* **2018**, *38*, 03046. [CrossRef]
5. Catbas, F.N.; Ciloglu, S.K.; Hasancebi, O.; Grimmelsman, K.; Aktan, A.E. Limitations in structural identification of large constructed structures. *J. Struct. Eng.* **2007**, *133*, 1051–1066. [CrossRef]
6. Ho, D.D.; Kim, J.T.; Stubbs, N.; Park, W.S. Prestress-force estimation in PSC girder using modal parameters and system identification. *Adv. Struct. Eng.* **2012**, *15*, 997–1012. [CrossRef]
7. Li, H.N.; Yi, T.H.; Ren, L.; Li, D.S.; Huo, L.S. Reviews on innovations and applications in structural health monitoring for infrastructures. *Struct. Monit. Maint.* **2014**, *1*, 1. [CrossRef]
8. Lee, S.Y.; Nguyen, K.D.; Huynh, T.C.; Kim, J.T.; Yi, J.H.; Han, S.H. Vibration-based damage monitoring of harbor caisson structure with damaged foundation-structure interface. *Smart Struct. Syst.* **2012**, *10*, 517–546. [CrossRef]
9. Huynh, T.C.; Lee, S.Y.; Kim, J.T.; Park, W.S.; Han, S.H. Simplified planar model for damage estimation of interlocked caisson system. *Smart Struct. Syst.* **2013**, *12*, 441–463. [CrossRef]
10. Huynh, T.C.; Lee, S.Y.; Dang, N.L.; Kim, J.T. Vibration-based structural identification of caisson-foundation system via in situ measurement and simplified model. *Struct. Control Health Monit.* **2019**, *26*, e2315. [CrossRef]
11. Lee, S.Y.; Huynh, T.C.; Kim, J.T. A practical scheme of vibration monitoring and modal analysis for caisson breakwater. *Coast. Eng.* **2018**, *137*, 103–119. [CrossRef]
12. Lee, S.Y.; Huynh, T.C.; Dang, N.L.; Kim, J.T. Vibration characteristics of caisson breakwater for various waves, sea levels, and foundations. *Smart Struct. Syst.* **2019**, *24*, 525–539.
13. Pham, N.L.; Ta, Q.B.; Kim, J.T. Pseudo-Damage Simulation and CNN Deep Learning for Damage Identification of Submerged Structure-Foundation System. *Struct. Health Monit.* 2024, *under review*.
14. Teng, S.; Chen, G.; Yan, Z.; Cheng, L.; Bassir, D. Vibration-based structural damage detection using 1-D convolutional neural network and transfer learning. *Struct. Health Monit.* **2023**, *22*, 2888–2909. [CrossRef]
15. Huang, J.; Yin, X.; Kaewunruen, S. Quantification of dynamic track stiffness using machine learning. *IEEE Access* **2022**, *10*, 78747–78753. [CrossRef]

16. Li, F.; Wang, L.; Wang, D.; Wu, J.; Zhao, H. An adaptive multiscale fully convolutional network for bearing fault diagnosis under noisy environments. *Measurement* **2023**, *216*, 112993. [CrossRef]
17. Ming, G.; Guanying, D.; Jihua, Y. Dynamic studies on caisson-type breakwaters. *Coast. Eng.* **1988**, 2469–2478. [CrossRef]
18. Khodabandehlou, H.; Pekcan, G.; Fadali, M.S. Vibration-based structural condition assessment using convolution neural networks. *Struct. Control Health Monit.* **2019**, *26*, e2308. [CrossRef]
19. Lin, Y.Z.; Nie, Z.H.; Ma, H.W. Structural damage detection with automatic feature-extraction through deep learning. *Comput.-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 1025–1046. [CrossRef]
20. Brincker, R.; Zhang, L.; Andersen, P. Modal identification of output-only systems using frequency domain decomposition. *Smart Mater. Struct.* **2001**, *10*, 441. [CrossRef]
21. Yi, J.H.; Yun, C.B. Comparative study on modal identification methods using output-only information. *Struct. Eng. Mech. Int. J.* **2004**, *17*, 445–466. [CrossRef]
22. Ewins, D.J. *Modal Testing: Theory, Practice and Application*; John Wiley & Sons: Hoboken, NJ, USA, 2009.
23. Sarawgi, Y.; Somani, S.; Chhabra, A.; Sangwan, D. Nonparametric vibration-based damage detection technique for structural health monitoring using 1D CNN. In Proceedings of the Computer Vision and Image Processing: 4th International Conference, Jaipur, India, 27–29 September 2019.
24. Pearson, K. Contributions to the mathematical theory of evolution. *Philos. Trans. R. Soc. London* **1894**, *185*, 71–110.
25. Loeve, M.M. *Probability Theory*, 2nd ed.; Springer: New York, NY, USA, 1960.

applied sciences

MDPI

Article

Engine Fault Detection by Sound Analysis and Machine Learning

Ferit Akbalık [1,*], Abdulnasır Yıldız [2], Ömer Faruk Ertuğrul [3] and Hasan Zan [4]

1 Social Sciences Vocational School, Batman University, 72040 Batman, Turkey
2 Department of Electrical and Electronics Engineering, Dicle University, 21280 Diyarbakir, Turkey; abnayil@dicle.edu.tr
3 Department of Electrical and Electronics Engineering, Batman University, 72040 Batman, Turkey; omerfaruk.ertugrul@batman.edu.tr
4 Vocational School, Mardin Artuklu University, 47200 Mardin, Turkey; hasanzan@artuklu.edu.tr
* Correspondence: ferit.akbalik@batman.edu.tr

Abstract: Traditional vehicle fault diagnosis methods rely heavily on the expertise of mechanics or diagnostic tools available at service centers, which can be costly, time-consuming, and may not always provide accurate results. This study presents a comprehensive vehicle fault diagnosis framework, which utilized Mel-Frequency Cepstral Coefficients (MFCCs), Discrete Wavelet Transform (DWT)-based features, and the Extreme Learning Machine (ELM) classifier. To address the limitations of previous works, the proposed framework leverages a large, diverse dataset encompassing various vehicle models and real-world operating conditions. Significantly improved robustness and generalizability of the fault diagnosis system were achieved. The results of the experiments demonstrate the superiority of the MFCC-based features combined with the ELM classifier, achieving the highest performance metrics in terms of accuracy, precision, recall, F1-score, macro F1-score, and weighted F1-score, which are 92.17%, 92.24%, 92.22%, 92.10%, and 92.06%, respectively. Slightly lower performance was obtained while employing the DWT-based features compared to employing MFCC-based features. Additionally, frequency analysis was conducted to identify specific frequency bins, which are the most indicative of different fault types in providing valuable guidance for future diagnostic efforts. Overall, the proposed framework provides a reliable and practical solution for accurate vehicle fault detection, paving the way for future advancements in automotive diagnostics.

Keywords: vehicle fault detection; extreme learning machines; mel-frequency cepstral coefficients; wavelet transform

1. Introduction

Car faults are an unfortunate reality for many drivers, with unexpected vehicle repairs being a common occurrence in the automotive industry. These faults can range from minor issues (e.g., a blown fuse, a flat tire) to more serious problems that can impact vehicle safety and performance. For instance, a faulty brake system or a malfunctioning engine can pose significant risks to drivers and passengers and can result in costly repairs. In fact, according to a report by the American Automobile Association, the average expenditure on unexpected vehicle repairs in the United States was between USD 500 and USD 600 per vehicle in 2017 [1]. However, this figure can vary widely based on several factors, such as the age, and model of the vehicle, as well as driving habits and conditions.

Traditional car fault diagnosis methods have relied heavily on the expertise of mechanics or diagnostic tools, which may be available at service centers [2,3]. These methods often involve manual inspections, computerized scanning, or diagnostic tests to identify and address potential issues in vehicles, such as engine faults, front-end assembly faults, airflow faults, spark plug, and electrical faults. However, these traditional methods are often costly, time-consuming, and may not always provide accurate results [4]. Recently, there has been a surge of interest in machine learning-based fault diagnosis systems [5–7]. Leveraging advancements in artificial intelligence and data analytics, these systems aim to automate fault detection processes, enhance accuracy, and enable proactive maintenance strategies [8,9].

The general structure of machine learning-based fault diagnosis methods encompasses several stages, i.e., data acquisition, preprocessing, feature extraction, and classification [10,11]. Data acquisition involves gathering relevant information from vehicles, such as sensor readings and diagnostic codes, and recording relevant signals [12]. Preprocessing focuses on cleaning and organizing the data to eliminate noise and inconsistencies [13]. Feature extraction aims to identify key parameters or characteristics indicative of potential faults [14]. Finally, classification algorithms are applied to categorize the data into different fault types or states [15].

Two primary types of signals, which are vibration and sound signals, are employed in fault diagnosis [16]. Vibration signals capture mechanical movements and dynamics within the vehicle and provide insights into structural integrity and component performance. On the other hand, sound signals reflect acoustic emissions associated with engine operation and component interactions and offer valuable clues about the health and functionality of various vehicle parts.

Several studies have explored fault diagnosis using vibration signals and demonstrate the versatility and effectiveness of this approach. For example, Jegadeeshwaran and Sugumaran [17] presented a method of vibration-based continuous monitoring system and analysis using a machine learning approach. Their study focused on fault diagnosis in hydraulic braking systems by acquiring vibration signals from a piezoelectric transducer under both good and faulty brake conditions. They employed decision tree algorithms to identify the most relevant features among different faulty conditions and they achieved a classification accuracy of 97.45%. Similarly, Barbieri et al. [18] aimed to identify damages and diagnose damaged components in automotive gearboxes by comparing vibration signals of damaged and undamaged systems. They employed various signal analysis techniques (e.g., wavelet transform and mathematic morphology) to verify damage presence and used a signal processing technique combining pattern spectrum and selective filtering for component failure identification. Jafarian et al. [19] explored vibration analysis for fault detection in an internal combustion engine and focused on detecting faults related to poppet valve clearance and incomplete combustion. They utilized four accelerometers on the engine body, applied the Principal Component Analysis (PCA) technique for data analysis, and achieved high efficiency in fault classification and detection. These studies collectively highlight the wide applicability and effectiveness of vibration-based fault diagnosis in diverse automotive systems. A summary of studies employing vibration signals is given in Table 1.

Table 1. Summary of studies on vehicle fault detection using vibration signals.

Reference	Dataset	Methodology	Key Findings
Jegadeeshwaran and Sugumaran [17]	550 signals recorded from a brake system in a controlled environment, 8 brake fault types	Statistical feature extraction and decision trees	Overall accuracy of 97.5%
Barbieri et al. [18]	Multi-step recording of signals from 13 gearboxes in a lab environment, 3 gearbox fault types	Wavelet transform-based feature extraction and comparison-based classification	Significant differences between signals recorded from gearboxes with and without damage

Table 1. *Cont.*

Reference	Dataset	Methodology	Key Findings
Jafarian et al. [19]	Signals recorded from a car engine in a controlled environment	Statistical features of PCA components and comparison-based classification	High accuracy in fault classification
Ahmed et al. [20]	600 signals recorded from a car engine in a lab environment, 7 engine fault types	Time–frequency domain feature extraction and artificial neural networks	Overall accuracy of 97%
Taghizadeh-Alisaraei and Mahdavian [21]	Signals recorded from a car engine in a lab environment, in cases involving faulty and healthy injectors	Time–frequency analysis	Significant differences between signals recorded from the engine with and without faulty injectors
Wang et al. [22]	411 signals recorded from 8 valve train states in a controlled environment	Time–frequency analysis and probabilistic neural networks	Overall accuracy of 97.6%

Although high success rates have been reported in vibration signals, it is hard to implement them in a real-word system. Therefore, a practical method is required to distinguish the faults. Sound signals offer distinct advantages in fault diagnosis due to their ease of recording using commonly available devices such as cell phones or microphones. Therefore, this makes sound analysis a practical and cost-effective approach to diagnosing vehicle faults. Studies conducted by various researchers further exemplify the potential of sound-based fault diagnosis in automotive systems. For instance, Madain et al. [23] identified distinct sounds associated with specific engine malfunctions and developed an algorithm using sound techniques in diagnosis. They reported high error detection rates through analysis of engine sound samples collected from a laboratory environment. Similarly, Jian-Da Wu and Chiu-Hong Liu [24] developed a fault diagnosis system for internal combustion engines based on the discrete wavelet transform technique applied to sound emission signals and showcased its effectiveness in fault recognition under diverse engine operating conditions. Another study that delved into the application of acoustic signal processing methods for assessing internal combustion engine technical conditions proposed new algorithms for automatic detection of valve clearance issues based on acoustic signal components [25]. Additionally, Mofleh et al. [26] conducted a study aimed at detecting faults in spark-ignition engines using acoustic signals and an Artificial Neural Network (ANN) system. It highlighted the high potential of ANN-based fault detection in internal combustion engines using acoustic signals, particularly in identifying simulated spark plug and misfire faults. These studies collectively underscore the practicality and efficacy of sound-based fault diagnosis methods in the automotive industry and offer valuable insights for developing reliable diagnostic systems. A summary of such studies employing sound signals is provided in Table 2.

Despite recent advancements, current studies in vehicle fault diagnosis often encounter significant drawbacks. These limitations include the reliance on data recorded in controlled laboratory settings, which may not fully represent real-world vehicle conditions. Furthermore, many studies are constrained to specific car models or fault types, limiting the generalizability and applicability of their findings. There is a pressing need for a comprehensive dataset that encompasses diverse vehicle models, real-world vehicle conditions, and a wide range of fault scenarios to enhance the effectiveness and accuracy of fault diagnosis systems.

Table 2. Summary of studies on vehicle fault detection using sound signals.

Reference	Dataset	Methodology	Key Findings
Madain et al. [23]	Few signals recorded from two cars, 2 types of faults (shell bearing and exhaust)	RPM-based dominant frequency and comparison-based classification	100% accuracy for the first car and 90% for the second
Wu and Liu [24]	300 signals recorded from one engine, 5 engine-related faults	Wavelet transform-based feature extraction and artificial neural networks	Overall accuracy of 99%
Figlus et al. [25]	Signals recorded from two engines, one fault type (excessive valve clearance)	Wavelet transform-based feature extraction and comparison-based classification	Efficiency of the algorithm presented
Mofleh et al. [26]	60 signals recorded from one engine, 2 spark-ignition faults	Frequency domain feature extraction and artificial neural networks	Overall accuracy of 73.3%
Nevea and Sybingco [27]	36 signals recorded from the 1996–2000 model Honda Civic, 3 engine-related faults	Fourier transform and power spectrum density for feature extraction and fuzzy logic inference for classification	Overall accuracy of 56%
Wang et al. [5]	140 signal recordings from Santana 2000 model vehicle	Hilbert–Huang transform for feature extraction and support vector machines for classification	Overall accuracy of 90%
Siegel et al. [28]	992 2.5-long signals recorded from 4 cars, one type of engine fault	Fourier, Wavelet, and MFCC-based feature extraction and support vector machines	Overall accuracy of 99%
Yılmaz et al. [29]	100 signals recorded from various cars, 2 engine-related faults	Wavelet transform-based feature extraction and k-nearest neighborhood	Overall accuracy of 91.8%

Our study directly addresses these challenges by leveraging a diverse and extensive dataset comprising real-life vehicle sounds. This dataset captures a wide array of vehicle conditions various vehicle models, and an extensive range of fault scenarios encountered in everyday driving. We employ advanced signal processing techniques, including Mel-Frequency Cepstral Coefficients (MFCCs), Wavelet Transform, and Relief-F methods, for robust feature extraction and feature selection, while using Extreme Learning Machines (ELM) for the classification.

The summary of our study's approach and key contributions is as follows:

- Instead of employing a laboratory-collected dataset, comprehensive data were collected from real-life vehicle conditions and diverse vehicle models.
- In order to increase the success of the proposed approach, advanced signal processing techniques, which are MFCC, Wavelet Transform, and Relief-F, were employed.
- A thorough frequency analysis was conducted in each fault type, and specific frequency components, which are associated with different types of faults, were identified.

This paper is structured as follows. Section 2 gives details about utilized data collection methods and focuses on how sound signals were acquired from vehicular systems under various operating conditions. Section 3 elaborates on employed methodology encompassing employed signal processing techniques and feature extraction methodologies that are used in vehicle fault diagnosis based on sound signals. In Section 4, we present the obtained experimental results, which includes performance evaluations of the employed diagnostic system, comparisons with existing methods in the literature, and a detailed frequency analysis of each identified fault type. Furthermore, the implications of our findings and insights gained from the experimental outcomes are discussed. Finally, Section 5 serves as the conclusion of this study and summarizes key contributions made in this research, and proposes directions for future research and development in the field of vehicle fault diagnosis by sound signal analysis.

2. Dataset

Audio signal recordings were collected from vehicles, which were serviced at official Ford or Toyota service centers and ensured a diverse range of cars from these reputable brands. A cellphone served as the recording device, which captured sounds, while the cars were stationary, and their engines were idling at ideal operating temperatures. As seen in Figure 1, the cellphone was positioned 15 cm above the hood and centered, with the hood closed to mimic real-world conditions. Engine sounds were recorded for 30 s each, sampling at a frequency of 48 kHz to capture detailed acoustic information.

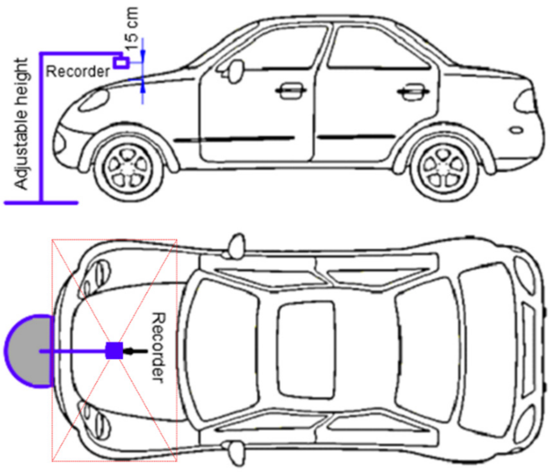


Figure 1. Setup for audio signal recording.

Professional mechanics diagnosed the cars as either healthy or with one of the following faults: spark plug issues, airflow irregularities, electrical malfunctions, engine/turbo problems, or front-end problems. The distribution of each diagnostic class is outlined in Table 3.

Table 3. Distribution of vehicle diagnostic classes.

Diagnostic Class	Number of Cases
Healthy	50
Spark Plug Issues	40
Airflow Irregularities	52
Engine/Turbo Problems	44
Front-End Problems	48
Electrical Malfunctions	46

Spark Plug Issue: Typically related to ignition problems, which result in misfires, rough idling, and decreased engine performance.

Airflow Irregularities: Pertaining to issues with the air intake system that affect engine combustion and efficiency.

Electrical Malfunctions: Encompassing faults within the vehicle's electrical system that directly impact engine performance. This may include issues with sensors, wiring, or other electrical components that affect engine operation and efficiency.

Engine/Turbo Problems: Referring to issues within the engine or turbocharger system that impact power delivery and overall engine performance.

Front-End Problems: Including issues with steering, suspension, or other components affecting the vehicle's front-end operation.

The collected dataset comprises audio signals, which were collected from a wide range of gasoline-engine vehicles such as Ford Focus (2014–2021), Ford Kuga (2020–2021), Ford Ecosport (2021), Ford Mondeo (2016), Toyota Corolla (2015), and Toyota Auris (2010). It was aimed to ensure diversity in vehicle models to capture a comprehensive range of engine sounds and fault types.

In addition, an example signal for each vehicle diagnostic class is provided in Figure 2 in order to demonstrate the characteristics of the recorded audio signals for each class.

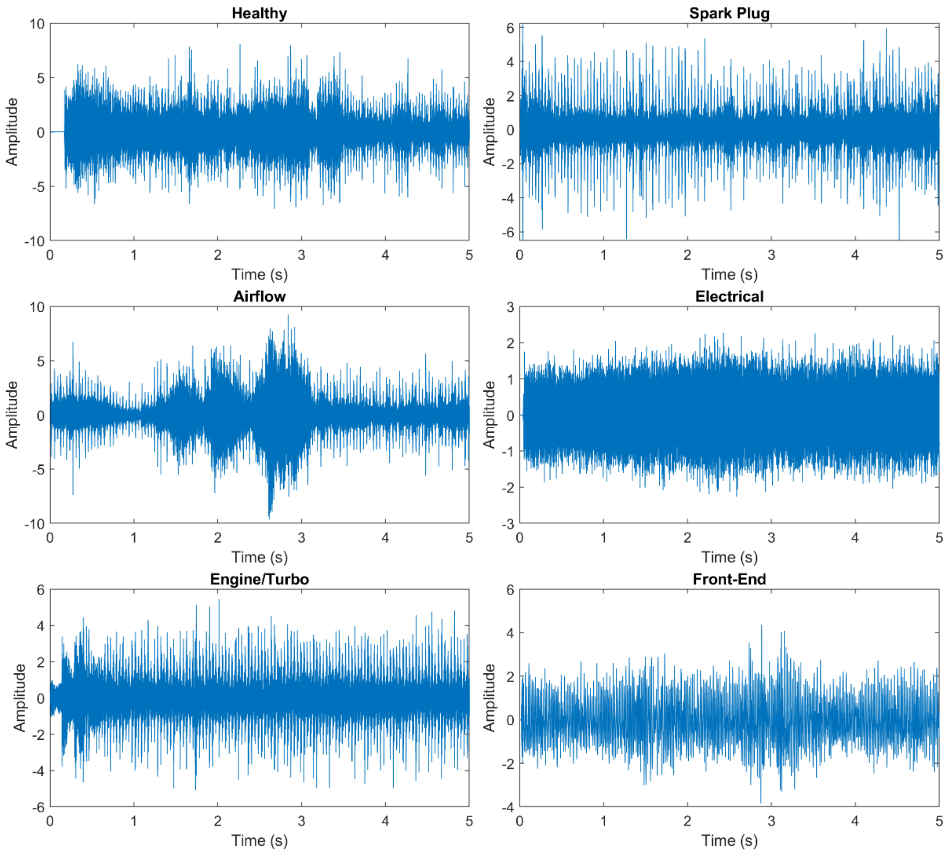


Figure 2. Example audio signals for each vehicle diagnostic class.

3. Methodology

3.1. Overview

This paper presents a framework comprising data acquisition, preprocessing, feature extraction, fine-tuning, and classification, as illustrated in Figure 3. Data acquisition details are explained in Section 2. In the preprocessing phase, each audio signal was normalized and trimmed to 5 s to ensure consistency. Feature extraction was performed using Mel-Frequency Cepstral Coefficients (MFCCs) and Discrete Wavelet Transform (DWT) techniques, capturing important acoustic characteristics for fault diagnosis. The fine-tuning process involves Grid Search optimization to find optimal hyperparameters for machine learning models, while Extreme Learning Machines (ELM) are utilized for efficient and accurate classification of vehicle fault diagnostic classes. Additionally, Fourier transform-based feature extraction and Relief-f feature selection algorithm are employed for frequency analysis of each fault type, aimed at uncovering crucial frequency components associated with different fault types.

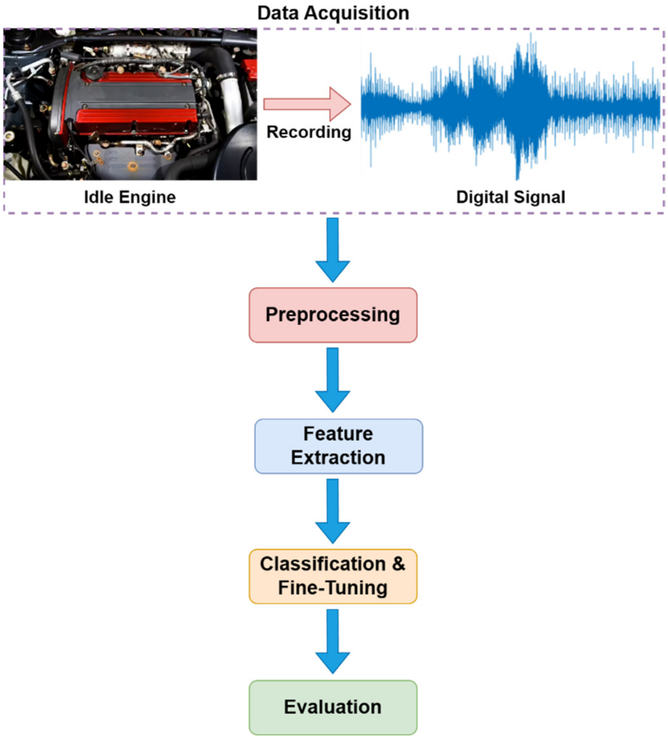


Figure 3. Methodology block diagram for vehicle fault diagnosis using sound signals and machine learning.

3.2. Feature Extraction

Feature extraction is a critical step in audio signal processing for fault diagnosis in vehicle systems. This section outlines two widely used techniques: MFCC and DWT.

MFCC is a prominent technique in audio signal processing. The process involves several steps [30]:

- Windowing the audio signal into short segments using Equation (1).

$$W(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \qquad N-1 \geq n \geq 0 \tag{1}$$

- Applying the Discrete Fourier Transform (DFT) to each frame to convert the time-domain signal into the frequency domain. DFT can be defined for a frame X comprising of N samples as in Equation (2).

$$X_n = \sum_{k=0}^{N-1} X_k e^{-\frac{2\pi jkn}{N}} \quad , \qquad n = 0, 1, 2, 3 \ldots, N-1 \tag{2}$$

- Mapping the frequency spectrum to the Mel-scale to approximate human auditory perception using Equation (3).

$$f_{mel} = 2595 \log_{10}\left(1 + \frac{f_{lineer}}{700}\right) \tag{3}$$

- Applying logarithmic compression to the Mel spectrum.
- Computing the Discrete Cosine Transform (DCT) to obtain the MFCC coefficients as follows:

$$C_k = \sum_{n=0}^{N-1} \log(S_n) \cos\left[k\pi \frac{(2n+1)}{2N}\right] \tag{4}$$

where C_k represents the k-th MFCC coefficient, N represents the total number of mel-frequency filters, S_n represents the energy of the n-th mel-frequency filter bank, and k is the index of the MFCC coefficient, usually ranging from 0 to $k-1$.

These MFCC coefficients capture essential spectral features of the audio signal, such as pitch, timbre, and formants, which are crucial for fault diagnosis in vehicle systems. In this study, the mean of each coefficient over each frame was calculated as a feature. Details regarding MFCC parameters are provided in Section 4.

DWT is a powerful tool for analyzing signals in both time and frequency domains simultaneously [31]. The process involves the following:

- Decomposing the audio signal into different frequency bands using wavelet functions, such as Daubechies and Symlets wavelets. For an input signal $x(t)$, the approximation coefficients A_j and detail coefficients D_j at level j are computed using:

$$A_{j+1}(k) = \sum_n h(n-2k)A_j(n) \tag{5}$$

$$D_{j+1}(k) = \sum_n g(n-2k)A_j(n) \tag{6}$$

  where h and g are the low-pass and high-pass filter coefficients, respectively, corresponding to the wavelet function.
- Extracting features from each level of decomposition.

These features provide insights into the time–frequency characteristics of the signal, aiding in fault detection and classification in vehicle systems. In this study, we calculated the energy, standard deviation, and entropy of each coefficient to serve as features for the classification task. Details regarding the decomposition levels and wavelets employed are given in Section 4.

3.3. Classification and Fine-Tuning

For the classification of vehicle fault diagnostic classes, we employed ELM, a machine learning algorithm based on a single hidden layer feedforward neural network architecture [32–35]. In the ELM algorithm, the input layer weights and thresholds are assigned randomly, while the output layer weights are calculated based on these assignments. ELM training consists of two parts: (1) generating random hidden layer parameters from a predefined range, and (2) calculating the generalized inverse output weight matrix [36]. ELM is popular due to its fast learning speed, generalization ability, and simplicity. As illustrated in Figure 4, the ELM inputs map the features to the hidden layer, which are then passed on to the output layer. The output from ELM learning can be used for various tasks such as classification, regression, and clustering.

ELM transforms input vector $x = [x_1, x_2, \ldots, x_n]$ into the hidden layer representation using the weight matrix W and the bias vector b. Each neuron in the hidden layer uses an activation function $g(.)$. The connections between the hidden layer outputs and the output layer are represented by the weights β. In ELM, the weights and bias values for the hidden layer are randomly assigned and kept fixed. The weights β for the output layer are learned using the least squares method. The relationship between the input and output of the ELM is calculated as summarized in Equations (7)–(9).

$$H = g(W \cdot X + b) \tag{7}$$

$$Y = H \cdot \beta \tag{8}$$

$$\beta = H^{+} \cdot T \tag{9}$$

where H is the matrix of hidden layer outputs, Y denotes the output vector, H^+ represents the Moore–Penrose pseudoinverse of the matrix H, and T is the target vector.

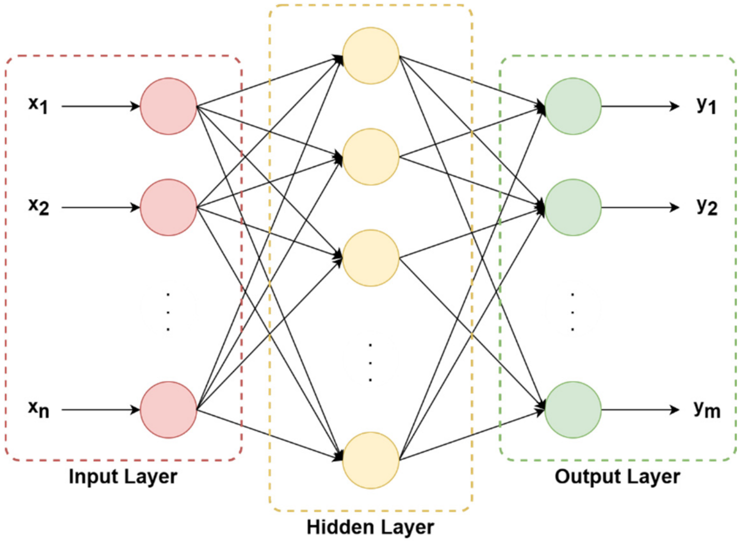


Figure 4. Structure of an ELM model.

To optimize the performance of our ELM classifier, we utilized Grid Search, a hyperparameter optimization technique [37]. Grid Search systematically explores a predefined set of hyperparameters, evaluating each combination using 5-fold cross-validation to determine the optimal parameters that yield the highest classification accuracy. This method allows for a thorough examination of the model's performance across various settings, ensuring the selection of the most effective parameter set. The specific parameters used for Grid Search, such as the number of hidden neurons and activation functions, are outlined in Table 4. This exhaustive search process ensures that our model is finely tuned, enhancing its robustness and reliability in delivering accurate fault diagnosis results.

Table 4. Grid search parameters for optimizing ELM classifier.

Hyperparameter	Values
Number of hidden neurons	2, 3, 4, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100
Activation Function	sigmoid, sine, triangular basis, radial basis, and hard limit function

3.4. Performance Evaluation

The performance of the classification model is assessed using standard metrics, including accuracy, precision, recall, macro, and weighted averaged F1-scores. These metrics provide a comprehensive understanding of the model's effectiveness in correctly diagnosing vehicle faults. Accuracy measures the overall correctness of the model, precision indicates the proportion of true positive diagnoses out of all positive diagnoses, recall (or sensitivity) assesses the model's ability to identify true positives, and F1-score is the harmonic mean of precision and recall, offering a balanced evaluation metric [38]. Each metric is calculated as outlined in Equations (10)–(14). Five-fold cross-validation is employed in the performance analysis to provide a reliable estimate of the model's effectiveness across different subsets of the data. This method helps in ensuring that the evaluation is not biased by any particular partitioning of the data.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{10}$$

$$Precision\ (PR) = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i} \tag{11}$$

where N is the number of classes, TP_i is the number of true positives for class i and FP_i is the number of false positives for class i.

$$Recall\ (RE) = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FN_i} \tag{12}$$

where FN_i is the number of false negatives for class i.

$$Macro\ F1\text{-}Score = \frac{1}{N}\sum_{i=1}^{N}\frac{2 \cdot PR_i \cdot RE_i}{PR_i + RE_i} \tag{13}$$

$$Weighted\ F1\text{-}Score = \sum_{i=1}^{N} w_i \frac{2 \cdot PR_i \cdot RE_i}{PR_i + RE_i} \tag{14}$$

where PR_i and RE_i are the precision and recall for class i, respectively. Additionally, confusion matrices, which display the model's classification results, were constructed to provide a detailed view of how well the classifier distinguishes between fault classes. This visual representation is crucial for understanding the classifier's strengths and areas needing improvement.

3.5. Relief-F Algorithm and Frequency Analysis

In this section, we detail the process of feature selection using the Relief-F algorithm and the subsequent frequency analysis conducted to identify the most relevant frequency components for each fault type.

The Relief-F algorithm is a powerful feature selection method that helps identify and rank the most relevant features in a dataset, making it particularly useful for high-dimensional data [39]. Relief-F works by iteratively sampling instances from the dataset and comparing the sampled instance with its nearest neighbors of the same and different classes. For each feature, it increases the relevance score if the feature value of the instance is similar to that of its nearest neighbor from the same class and decreases the score if the feature value is similar to that of its nearest neighbor from a different class. This process effectively highlights features that are consistently good at distinguishing between classes while ignoring irrelevant or redundant features.

To understand the frequency characteristics of the audio signals and identify the most diagnostically relevant frequencies for each fault type, we conducted a detailed frequency analysis using the following steps:

1. Frequency Spectrum Calculation: The frequency spectrum of each audio signal was calculated using the Fast Fourier Transform (FFT). FFT transforms the time-domain signal into its frequency-domain representation, providing insight into the signal's frequency components.
2. Spectrum Binning: The resulting frequency spectrum was divided into bins, each 100 Hz wide. This binning process organizes the frequency data into manageable segments. As there are very limited frequency components beyond 15 kHz, a frequency range of 0–15 kHz was considered.
3. Power Calculation: For each 100 Hz bin, the total power was calculated. This step involves summing the squared magnitudes of the frequency components within each bin, giving a measure of the signal's energy within that frequency range.
4. Relevance Determination: For each fault type, the Relief-F feature selection algorithm was applied to the binned frequency spectrum. By evaluating the relevance scores of the bins, Relief-F identified the 5 most relevant bins that contributed most significantly

to distinguishing between faulty and healthy conditions. These relevant bins highlight specific frequency components that are key indicators of each fault type.

4. Results and Discussion

In this section, the vehicle fault diagnosis framework was presented and discussed. The performance metrics and analysis are provided for models utilizing MFCC-based features and DWT-based features. Additionally, a detailed frequency analysis to identify the most relevant frequency components for each fault type, was given. Obtained results are also compared with existing studies to highlight the effectiveness and improvements achieved by the employed methodology.

4.1. Results for MFCC-Based Features

The performance of the vehicle fault diagnosis model using MFCC-based features is evaluated and summarized in Table 5. Various configurations of MFCC parameters, including the number of coefficients and window length, were tested to identify the optimal settings. The table also presents the best hyperparameters found through Grid Search for the ELM classifier, such as the activation function and number of neurons, alongside the resulting precision, recall, F1-score, and accuracy metrics. A 50% window overlap was used in all experiments to enhance feature extraction.

Table 5. Performance results for MFCC-Based features. Values given for window length are in seconds. 50% window overlap is used for all experiments. The bold indicates the highest score.

MFCC Parameters		Best ELM Hyperparameters		Performance Results				
Number of Coefs.	Window Length	Activation Function	Number of Neurons	Precision	Recall	Macro F1-Score	Weighted F1-Score	Accuracy
5	0.02	tribas	100	86.19	85.60	85.70	85.72	85.71
10	0.02	sig	100	90.09	89.24	89.34	89.31	89.29
20	0.02	sin	100	89.95	90.02	89.91	89.88	90.00
30	0.02	sin	100	89.81	89.73	89.51	89.45	89.64
5	0.03	radbas	80	85.74	84.59	84.73	84.73	84.64
10	0.03	sig	90	89.71	89.83	89.69	89.65	89.64
20	0.03	sin	100	**92.24**	**92.22**	**92.10**	**92.06**	**92.14**
30	0.03	sin	90	90.43	89.77	89.65	89.59	89.64
5	0.04	tribas	100	88.37	87.67	87.87	87.98	87.86
10	0.04	sig	90	91.45	91.12	91.11	91.11	91.07
20	0.04	sig	100	91.14	90.79	90.69	90.61	90.71
30	0.04	sig	80	89.67	89.37	89.16	89.00	89.29

The results, which are given in Table 5, indicate that using 20 MFCC coefficients with a window length of 0.03 s and a *sine* activation function yielded the highest performance with a precision of 92.24%, recall of 92.22%, F1-score of 92.10%, and accuracy of 92.14%. This demonstrates that the choice of MFCC parameters and ELM hyperparameters significantly impacts the classification performance.

From the table, it can be observed that the number of MFCC coefficients and window length significantly affect the performance of the proposed method up to a certain point. Specifically, increasing the number of MFCC coefficients from 5 to 20 generally leads to an improvement in precision, recall, F1-score, and accuracy. However, further increasing the number of coefficients to 30 does not seem to result in any significant improvement. Similarly, increasing the window length from 0.02 to 0.03 or 0.04 generally leads to an improvement in performance, with the best results achieved at a window length of 0.03 for most MFCC parameter configurations. However, the choice of activation function and the number of neurons in the ELM classifier also seem to play a role in achieving the best performance.

The confusion matrix for the model with 20 MFCC coefficients and a window length of 0.02 s, shown in Figure 5, provides additional insights into the model's performance. Each cell in the matrix represents the number of instances for which the true class is represented by the row and the predicted class is represented by the column.

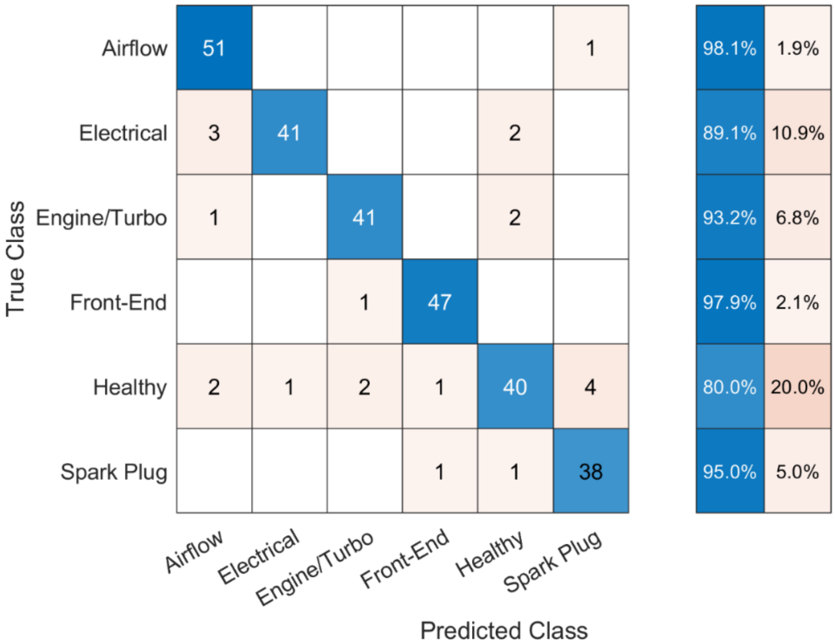


Figure 5. Confusion matrix for 20 MFCC coefficients and window length of 0.02 s using MFCC-based features.

Overall, the confusion matrix confirms that the model performs well across most fault types, with particularly high accuracy for airflow, front-end, and spark plug faults. Specifically, the model achieves the highest performance in detecting airflow faults with an accuracy of 98.1%, correctly identifying 51 out of 52 cases. However, the model shows the lowest performance in identifying healthy cases, with an accuracy of 80.0%, correctly predicting 40 out of 50 instances. This suggests that healthy cases are more prone to being misclassified as faults, indicating an area for further refinement.

The results of the experiments using MFCC-based features demonstrate the effectiveness of the proposed method for fault diagnosis in vehicles. The method can accurately diagnose different types of faults using the extracted MFCC features and the ELM classifier. The results also provide insights into the optimal combination of MFCC parameters and ELM hyperparameters, which can be used to improve the performance of the method in future studies. These findings demonstrate the model's effectiveness in vehicle fault diagnosis and highlight areas for further optimization.

4.2. Results for DWT-Based Features

The performance of the vehicle fault diagnosis model using DWT-based features is evaluated and summarized in Table 6. Various configurations of DWT parameters, including the decomposition level and widely used wavelets such as db4, db8, db20, sym3, and sym8, were tested to identify the optimal settings [24,40]. The table also presents the best hyperparameters found through Grid Search for the ELM classifier, such as the activation function and number of neurons, alongside the resulting precision, recall, F1-score, and accuracy metrics.

Table 6. Performance results for DWT-based features. The bold indicates the highest score.

DWT Parameters		Best ELM Hyperparameters		Performance Results				
Level	**Wavelet**	**Activation Function**	**Number of Neurons**	**Precision**	**Recall**	**Macro F1-Score**	**Weighted F1-Score**	**Accuracy**
2	db4	sig	80	82.11	81.09	81.30	81.69	81.43
3	db4	sin	90	82.84	80.63	81.11	81.43	81.07
4	db4	sin	100	78.46	77.44	77.24	77.25	77.50
5	db4	sin	100	81.72	80.95	80.79	80.98	81.07
2	db8	sig	90	82.22	80.99	81.19	81.47	81.43
3	db8	sin	70	83.51	82.42	82.64	82.77	82.86
4	db8	sin	100	77.57	77.16	76.94	77.09	77.14
5	db8	sin	100	79.62	78.05	78.29	78.25	78.21
2	db20	sin	100	**84.88**	83.27	83.54	83.52	83.57
3	db20	sin	80	83.00	82.37	82.45	82.46	82.50
4	db20	sin	100	77.68	75.93	76.00	76.04	76.07
5	db20	sin	100	80.57	79.93	79.97	80.21	80.00
2	sym3	sig	80	81.13	80.58	80.48	80.63	80.71
3	sym3	sin	90	84.36	83.32	83.46	83.52	83.57
4	sym3	sin	100	74.76	74.39	74.19	74.05	74.29
5	sym3	sig	50	78.48	77.46	77.43	75.55	77.50
2	sym8	sin	80	83.19	82.33	82.42	82.45	82.50
3	sym8	sin	70	84.17	**83.60**	**83.72**	**83.86**	**83.93**
4	sym8	sin	100	76.20	75.77	75.37	75.52	75.71
5	sym8	sin	100	77.10	75.98	76.07	76.26	76.07

From the table, it is clear that different combinations of DWT parameters and ELM hyperparameters result in varying levels of performance. The highest accuracy was achieved using a decomposition level of 3 and a *sym8* wavelet with a *sine* activation function, yielding a precision of 84.17%, recall of 83.60%, F1-score of 83.72%, and accuracy of 83.93%. This suggests that the choice of wavelet and decomposition level significantly influences the classification performance.

The confusion matrix in Figure 6 provides further insights into the classification performance for the best configuration (decomposition level of 3 and sym8 wavelet). The model achieves high accuracy for electrical faults (93.5%) and front-end faults (91.7%), indicating strong performance in these categories. However, the model shows lower accuracy for spark plug faults (65.0%), suggesting that distinguishing spark plug faults from other fault types remains a challenge. Additionally, healthy cases are identified with an accuracy of 74.0%, indicating some misclassification into fault categories, which highlights an area for further improvement.

Overall, the results of the experiments using DWT-based features demonstrate the effectiveness of the proposed method for fault diagnosis in vehicles. While the method shows strong performance in certain fault categories, it achieved lower performance overall compared to MFCC-based features. These findings provide insights into the optimal combination of DWT parameters and ELM hyperparameters and highlight areas for further optimization to enhance the performance of the method in future studies.

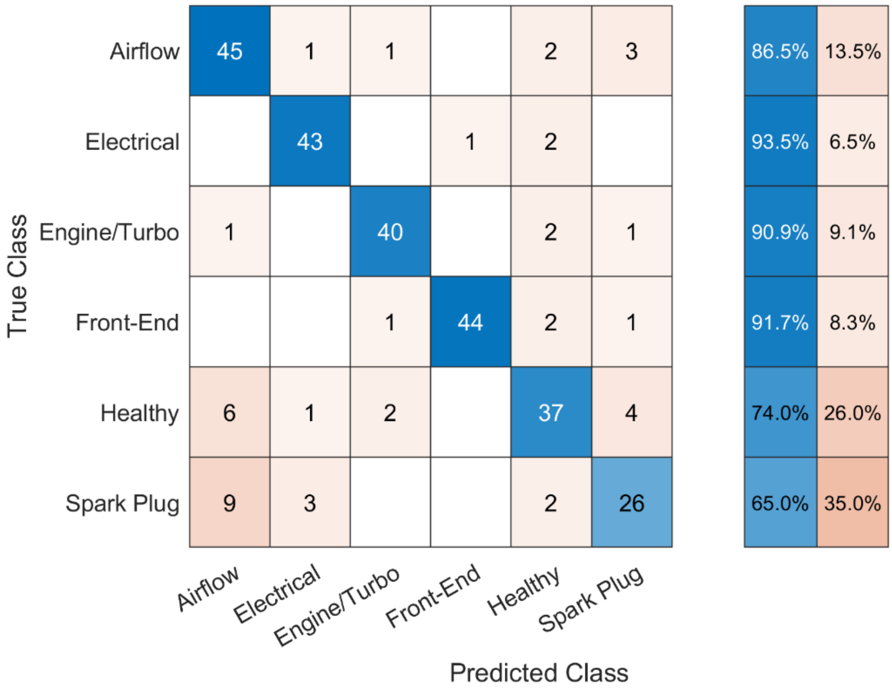


Figure 6. Confusion matrix for decomposition level of 3 and sym8 wavelet using DWT-based features.

4.3. Results for Frequency Analysis

In this section, the results of the frequency analysis, which was conducted to identify the most relevant frequency components for each fault type, are presented. Different types of engine faults often exhibit distinct frequency components. These components emerge due to the engine's physical structure and operational principles, with each fault generating unique sounds or vibrations at specific frequency ranges. Therefore, examining these frequency components is crucial for accurate fault diagnosis using sound analysis. Table 7 comprehensively outlines the relationship between the fault categories and their associated frequency groups.

Table 7. The most relevant frequencies for each fault type.

Fault Type	Most Relevant Frequency Bins (kHz)				
	Bin 1	**Bin 2**	**Bin 3**	**Bin 4**	**Bin 5**
Spark Plug Issues	5.2–5.3	9.4–9.5	12.4–12.5	13.2–13.3	14.1–14.2
Airflow Irregularities	3.0–3.1	9.1–9.2	9.2–9.3	10.0–10.1	12.2–12.3
Engine/Turbo Problems	0.7–0.8	4.8–4.9	9.5–9.6	12.1–12.2	12.5–12.6
Front-End Problems	6.6–6.7	8.5–8.6	8.9–9.0	11.6–11.7	13.5–13.6
Electrical Malfunctions	2.1–2.2	11.2–11.3	11.4–11.5	12.1–12.2	14.3–14.4

For spark plug issues, the most relevant frequency bin is 5.2 to 5.3 kHz, indicating that monitoring this high-frequency range is crucial for accurate detection. Airflow irregularities are most prominently indicated by the 3.0 to 3.1 kHz bin, highlighting the need for precise analysis in this low-frequency range. Engine/turbo problems are best identified by the 0.7 to 0.8 kHz bin, suggesting these faults manifest through specific low-frequency sounds. Front-end problems are primarily associated with the 6.6 to 6.7 kHz bin, essential for identifying issues related to components such as the suspension or chassis. Electrical malfunctions affecting the engine show the highest relevance at the 2.1 to 2.2 kHz bin, possibly due to distinctive noise patterns. These findings underscore the importance of focusing on these key frequencies for accurate fault detection.

Overall, the frequency analysis confirms that different fault types are associated with specific and most relevant frequency bins. The identification of these relevant frequency bins provides valuable guidance for future diagnostic efforts, suggesting that including these specific frequencies in the analysis can lead to better and more reliable fault detection outcomes. This insight into the frequency components of various faults enhances our understanding and ability to diagnose vehicle issues more accurately and efficiently.

4.4. Comparison with Other Studies

Numerous publications have addressed the diagnosis of vehicle malfunctions through engine sound analysis, employing a diverse array of methods. For instance, Navea and Sybingco [27] used Fourier transform and power spectral density to detect engine starting issues, drive belt problems, and valve-related faults, achieving a detection accuracy of 56% using recordings from the 1996–2000 Honda Civic model. Siegel et al. [28] focused on misfire faults in four vehicles, using Fourier transform, wavelet transform, and MFCC with SVM, resulting in a 99% accuracy rate. Wang et al. [5] recorded audio from a Santana 2000's engine cylinder head, using Hilbert–Huang transform and SVMs, achieving up to 90% accuracy. Kemalkar and Bairagi [41] studied lubrication, chain, crank, and valve faults in Honda Unicorn and Bajaj Pulsar motorcycles, employing MFCC and achieving accuracy rates between 50% and 75%. The reported results in the literature are summarized in Table 8.

Table 8. Reported results in the literature.

Reference	Dataset	Methodology	Key Findings
Wang et al. [5]	140 signal recordings from Santana 2000 model vehicle	Hilbert–Huang transform for feature extraction and support vector machines for classification	Overall accuracy of 90%
Nevea and Sybingco [27]	36 signals recorded from the 1996–2000 model Honda Civic, 3 engine-related faults	Fourier transform and power spectrum density for feature extraction and fuzzy logic inference for classification	Overall accuracy of 56%
Siegel et al. [28]	992 2.5-long signals recorded from 4 cars, one type of engine fault	Fourier, wavelet transform, and MFCC-based feature extraction and support vector machines	Overall accuracy of 99%
Kemalkar and Bairagi [41]	Sound samples of motorcycles under idling conditions are recorded using a voice recorder with 44.1-kHz sampling frequency and 16-bit quantization.	Liner predictive coding, hidden Markov model, artificial neural network	Accuracy rates between 50% and 75

A significant drawback of previous works is their reliance on data from controlled laboratory settings, often using the same brand of vehicles or a limited number of fault types. These controlled conditions do not adequately capture the variability and complexity of real-world scenarios, leading to models that may not perform well outside the specific conditions under which they were trained. Furthermore, the homogeneity of vehicle models in these studies limits the generalizability of their findings, as the diagnostic methods may not be applicable to different vehicle makes and models.

This dataset includes various types of faults and different environmental settings, thereby enhancing the robustness and generalizability of our fault diagnosis framework. By utilizing data from multiple vehicle brands and real-world conditions, the employed approach is designed to reflect the true complexity and variability of vehicle faults.

5. Conclusions

The proposed study presents a comprehensive vehicle fault diagnosis framework utilizing MFCC-based features and DWT-based features. The results demonstrate the efficacy of MFCC features combined with an ELM classifier, achieving the highest performance metrics. DWT-based features, while effective, showed slightly lower performance compared to MFCC features. Frequency analysis identified specific frequency bins most indicative of different fault types, providing valuable guidance for future diagnostic efforts. Additionally, by addressing the limitations of previous studies through the introduction of a large, diverse dataset encompassing various vehicle models and real-world operating conditions, we have significantly improved the robustness and generalizability of our fault diagnosis system. This framework provides a reliable and practical solution for accurate vehicle fault detection, paving the way for future advancements in automotive diagnostics.

Author Contributions: Methodology, F.A. and H.Z.; Software, H.Z.; Data curation, F.A.; Writing—review & editing, A.Y. and Ö.F.E.; Supervision, A.Y. and Ö.F.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets generated during and analyzed during the current study are available from the corresponding author upon reasonable request. The data are not publicly available due to confidentiality constrains.

Acknowledgments: The numerical calculations reported in this paper were fully performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Edmonds, E. One-in-Three U.S. Drivers Cannot Pay for an Unexpected Car Repair Bill. American Automobile Association, 2017. Available online: https://newsroom.aaa.com/2017/04/one-three-u-s-drivers-cannot-pay-unexpected-car-repair-bill (accessed on 28 April 2024).
2. Karaman, E.; Rende, H.; Akşahin, M.F. Recognition of Vehicles from Their Engine Sound. *Mühendis Ve Makina* **2019**, *60*, 148–164.
3. Xu, L.; Wang, T.; Xie, J.; Yang, J.; Gao, G. A Mechanism-Based Automatic Fault Diagnosis Method for Gearboxes. *Sensors* **2022**, *22*, 9150. [CrossRef] [PubMed]
4. Liu, Y.; Zhang, J.; Ma, L. A fault diagnosis approach for diesel engines based on self-adaptive WVD, improved FCBF and PECOC-RVM. *Neurocomputing* **2016**, *117*, 600–611. [CrossRef]
5. Wang, Y.; Ma, Q.; Zhu, Q.; Liu, X.; Zhao, L. An intelligent approach for engine fault diagnosis based on Hilbert–Huang transform and support vector machine. *Appl. Acoust.* **2014**, *75*, 1–9. [CrossRef]
6. Feng, Z.; Zhang, D.; Zuo, M.J. Planetary Gearbox Fault diagnosis via Joint Amplitude and Frequency Demodulation Analysis Based on Variational Mode Decomposition. *Appl. Sci.* **2017**, *7*, 775. [CrossRef]
7. López-Torres, C.; Riba, J.-R.; Garcia, A.; Romeral, L. Detection of Eccentricity Faults in Five-Phase Ferrite-PM Assisted Synchronous Reluctance Machines. *Appl. Sci.* **2017**, *7*, 565. [CrossRef]
8. Qu, Y.; He, M.; Deutsch, J.; He, D. Detection of Pitting in Gears Using a Deep Sparse Autoencoder. *Appl. Sci.* **2017**, *7*, 515. [CrossRef]
9. Gao, C.; Xue, W.; Ren, Y.; Zhou, Y. Numerical Control Machine Tool Fault Diagnosis Using Hybrid Stationary Subspace Analysis and Least Squares Support Vector Machine with a Single Sensor. *Appl. Sci.* **2017**, *7*, 346. [CrossRef]
10. Lupea, I.; Lupea, M.; Coroian, A. Helical Gearbox Defect Detection with Machine Learning Using Regular Mesh Components and Sidebands. *Sensors* **2024**, *24*, 3337. [CrossRef]
11. Moshrefi, A.; Tawfik, H.H.; Elsayed, M.Y.; Nabki, F. Industrial Fault Detection Employing Meta Ensemble Model Based on Contact Sensor Ultrasonic Signal. *Sensors* **2024**, *24*, 2297. [CrossRef]
12. Morenas, J.d.L.; Moya-Fernández, F.; López-Gómez, J.A. The Edge Application of Machine Learning Techniques for Fault Diagnosis in Electrical Machines. *Sensors* **2023**, *23*, 2649. [CrossRef]
13. Qu, N.; Wei, W.; Hu, C. Series Arc Fault Detection Based on Multimodal Feature Fusion. *Sensors* **2023**, *23*, 7646. [CrossRef] [PubMed]

14. Yang, X.; Yang, J.; Jin, Y.; Liu, Z. A New Method for Bearing Fault Diagnosis across Machines Based on Envelope Spectrum and Conditional Metric Learning. *Sensors* **2024**, *24*, 2674. [CrossRef]
15. Abid, A.; Khan, M.T.; Iqbal, J. A review on fault detection and diagnosis techniques: Basics and beyond. *Artif. Intell. Rev.* **2021**, *54*, 3639–3664. [CrossRef]
16. Baydar, N.; Ball, A. Detection of Gear Failures via Vibration and Acoustic Signals Using Wavelet Transform. *Mech. Syst. Signal Process.* **2003**, *17*, 787–804. [CrossRef]
17. Jegadeeshwaran, R.; Sugumaran, V. Method and Apparatus for Fault Diagnosis of Automobile Brake System Using Vibration Signals. *Recent Patents Signal Process.* **2013**, *3*, 2–11. [CrossRef]
18. Barbieri, N.; Barbieri, G.d.S.V.; Martins, B.M.; Barbieri, L.d.S.V.; de Lima, K.F. Analysis of automotive gearbox faults using vibration signal. *Mech. Syst. Signal Process.* **2019**, *129*, 148–163. [CrossRef]
19. Jafarian, K.; Darjani, M.; Honarkar, Z. Vibration analysis for fault detection of automobile engine using PCA technique. In Proceedings of the 2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA), Qazvin, Iran, 27–28 January 2016.
20. Ahmed, R.; El Sayed, M.; Gadsden, S.A.; Tjong, J.; Habibi, S. Automotive Internal-Combustion-Engine Fault Detection and Classification Using Artificial Neural Network Techniques. *IEEE Trans. Veh. Technol.* **2015**, *64*, 21–33. [CrossRef]
21. Taghizadeh-Alisaraei, A.; Mahdavian, A. Fault detection of injectors in diesel engines using vibration time-frequency analysis. *Appl. Acoust.* **2019**, *143*, 48–58. [CrossRef]
22. Wang, C.; Zhang, Y.; Zhong, Z. Fault diagnosis for diesel valve trains based on time–frequency images. *Mech. Syst. Signal Process.* **2008**, *22*, 1981–1993. [CrossRef]
23. Madain, M.; Al-Mosaiden, A.; Al-khassaweneh, M. Fault diagnosis in vehicle engines using sound recognition techniques. In Proceedings of the 2010 IEEE International Conference on Electro/Information Technology, Normal, IL, USA, 20–22 May 2010.
24. Wu, J.-D.; Liu, C.-H. Investigation of engine fault diagnosis using discrete wavelet transform and neural network. *Expert Syst. Appl.* **2008**, *35*, 1200–1213. [CrossRef]
25. Figlus, T.; Liščák, Š.; Wilk, A.; Łazarz, B. Condition monitoring of engine timing system by using wavelet packet decomposition of a acoustic signal. *J. Mech. Sci. Technol.* **2014**, *28*, 1663–1671. [CrossRef]
26. Mofleh, A.; Shmroukh, A.; Ghazaly, N. Fault Detection and Classification of Spark Ignition Engine Based on Acoustic Signals and Artificial Neural Network. *Int. J. Mech. Prod. Eng. Res. Dev.* **2020**, *10*, 5571–5578.
27. Navea, R.; Sybingco, E. Design and Implementation of an Acoustic-Based Car Engine Fault Diagnostic System in the Android Platform. In Proceedings of the International Research Conference in Higher Education, Manila, Philippines, 26–28 October 2013.
28. Siegel, J.; Kumar, S.; Ehrenberg, I.; Sarma, S. Engine Misfire Detection with Pervasive Mobile Audio. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Turin, Italy, 19–23 September 2016.
29. Yılmaz, G.; Mete, N.F.; Umugabekazi, U.; Aydemir, Ç. Dalgacık Dönüşümü ve Özbağlanım Model Parametreleri Öznitelikleri ile Otomobil Motor Seslerinden Arıza Tespiti. *J. Investig. Eng. Technol.* **2020**, *3*, 48–54.
30. Abdul, Z.K.; Al-Talabani, A.K. Mel Frequency Cepstral Coefficient and its Applications: A Review. *IEEE Access* **2022**, *10*, 122136–122158. [CrossRef]
31. Weeks, M.; Bayoumi, M. Discrete Wavelet Transform: Architectures, Design and Performance Issues. *J. VLSI Signal Process. Syst. Signal Image Video Technol.* **2003**, *35*, 155–178. [CrossRef]
32. Huang, G.; Huang, G.-B.; Song, S.; You, K. Trends in extreme learning machines: A review. *Neural Netw.* **2015**, *64*, 32–48. [CrossRef] [PubMed]
33. Chegni, A.M.; Ghavami, B.; Eftekhari, M. A GPU-based accelerated ELM and deep-ELM training algorithms for traditional and deep neural networks classifiers. *Intell. Syst. Appl.* **2022**, *15*, 200098. [CrossRef]
34. Qureshi, S.A.; Hussain, L.; Alshahrani, H.M.; Abbas, S.R.; Nour, M.K.; Fatima, N.; Khalid, M.I.; Sohail, H.; Mohamed, A.; Hilal, A.M. Gunshots Localization and Classification Model Based on Wind Noise Sensitivity Analysis Using Extreme Learning Machine. *IEEE Access* **2022**, *10*, 87302–87321. [CrossRef]
35. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
36. Zhang, J.; Li, Y.; Xiao, W.; Zhang, Z. Robust extreme learning machine for modeling with unknown noise. *J. Frankl. Inst.* **2020**, *357*, 9885–9908. [CrossRef]
37. Bashir, M.B.; Latiff, M.S.B.A.; Coulibaly, Y.; Yousif, A. A survey of grid-based searching techniques for large scale distributed data. *J. Netw. Comput. Appl.* **2016**, *60*, 170–179. [CrossRef]
38. Górny, K.; Kuwałek, P.; Pietrowski, W. Increasing Electric Vehicles Reliability by Non-Invasive Diagnosis of Motor Winding Faults. *Energies* **2021**, *14*, 2510. [CrossRef]
39. Kira, K.; Rendell, L.A. A Practical Approach to Feature Selection. In *Machine Learning Proceedings 1992*; Sleeman, D., Edwards, P., Eds.; Morgan Kaufmann: San Francisco, CA, USA, 1992; pp. 249–256.

40. Azadi, S.; Soltani, A. Fault detection of vehicle suspension system using wavelet analysis. *Veh. Syst. Dyn.* **2009**, *47*, 403–418. [CrossRef]
41. Kemalkar, A.K.; Bairagi, V.K. Engine fault diagnosis using sound analysis. In Proceedings of the 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), Pune, India, 9–10 September 2016.

applied sciences

Article

Superficial Defect Detection for Concrete Bridges Using YOLOv8 with Attention Mechanism and Deformation Convolution

Tijun Li [1], Gang Liu [1,2,*] and Shuaishuai Tan [1]

[1] School of Civil Engineering, Chongqing University, Chongqing 400045, China; litijun@cqu.edu.cn (T.L.); shuaishuai_tan@cqu.edu.cn (S.T.)
[2] Key Laboratory of New Technology for Construction of Cities in Mountain Area, Chongqing University, Ministry of Education, Chongqing 400045, China
* Correspondence: gliu@cqu.edu.cn

Abstract: The accuracy of detecting superficial bridge defects using the deep neural network approach decreases significantly under light variation and weak texture conditions. To address these issues, an enhanced intelligent detection method based on the YOLOv8 deep neural network is proposed in this study. Firstly, multi-branch coordinate attention (MBCA) is proposed to improve the accuracy of coordinate positioning by introducing a global perception module in coordinate attention mechanism. Furthermore, a deformable convolution based on MBCA is developed to improve the adaptability for complex feature shapes. Lastly, the deformable convolutional network attention YOLO (DCNA-YOLO) detection algorithm is formed by replacing the deep C2F structure in the YOLOv8 architecture with a deformable convolution. A supervised dataset consisting of 4794 bridge surface damage images is employed to verify the proposed method, and the results show that it achieves improvements of 2.0% and 3.4% in mAP and R. Meanwhile, the model complexity decreases by 1.2G, increasing the detection speed by $3.5/\text{f}\cdot\text{s}^{-1}$.

Keywords: concrete surface defects; deep learning; YOLO; attention mechanism; deformable convolution

1. Introduction

Bridges, which are crucial to national economic development, play an important role in the transport infrastructure. The number of constructed and operational bridges has exceeded 1.2 million in China, of which more than 70% are concrete bridges. As the service life increases, concrete bridges are inevitably subjected to various superficial defects such as cracking, concrete spalling, and rebar corrosion due to the combined effects of material aging, external loads, and the working environment [1]. These defects can affect the load-bearing capacity, service life, and overall safety of bridge structures [2]. Due to the large number and scale of concrete bridges, traditional manual inspections are inefficient to meet the demands of routine inspections. In addition, structures such as high piers and long spans require expensive auxiliary equipment for close-up inspection, which further increases inspection costs. With the improvements in resolution and accuracy of cameras, there is growing interest in using computer vision pattern recognition, including machine learning and deep learning, [3] to automatically detect bridge defects [4–7].

Machine learning methods mainly rely on template matching for defect detection, so such methods rely heavily on manual experience for sample feature extraction. In addition, single-layer features constructed by these methods have limited recognition ability when dealing with complex features and noise. Therefore, deep learning methods that can adaptively learn and extract image features have become mainstream, which are categorized into one-stage and two-stage methods based on the overall training conditions. One-stage detection methods extract features directly in the network to predict object classification and location, including the You Only Look Once (YOLO) [8] and Single Shot Multibox Detector (SSD) algorithms [9]. Two-stage methods typically use a Region Proposal Network (RPN) to extract potential target area which typically exhibit higher detection accuracy and stability as well as more computing resources, such as Regions with Convolutional Neural Network (RCNN) [10], Fast RCNN [11], Faster RCNN [12], and Mask RCNN [13].

The YOLO algorithm has been continuously improved since it was proposed by Joseph Redmon et al. in 2016, whose detection accuracy and computing speed have been further improved [8,14–21]. The main feature of the YOLO algorithm is that it transforms object detection into a regression problem, dividing the image into s $\times$ s grids and directly predicting the class probabilities and position information within the corresponding bounding box of each grid. YOLOv8 [21] adopts an anchor-free approach, combines the Task-Aligned assigner positive sample assignment strategy [22] and decoupled head, and introduces a C2F structure with a richer gradient flow and distribution focal loss, further improving detection accuracy and speed. However, the YOLO algorithm was primarily designed for general image classification. To improve its effectiveness in identifying concrete cracks, Zhang XB et al. [23] proposed a YOLOv5 model enhanced with a fusion of spatial pyramid pooling cross-stage partial connections (SPPCSPCs) and a transposed convolution to detect cracks on bridge surfaces from different angles, demonstrating superior detection performance compared to other models on the ZJU SYG dataset (a crack data set for object detection based on deep learning provided by Zhejiang University). Yu Z et al. [24] introduced a concrete structure crack detection method based on an improved YOLOv5, using a threshold segmentation method based on Otsu's maximum inter-class variance to remove background noise in images, and optimizing the initial anchor box sizes with the K-means method. The improved average accuracy in complex environments increased by 6.87%. Jin Q et al. [25] proposed an improved YOLOv5 algorithm based on transformer heads and the self-attention mechanism, which effectively improved the detection and classification capabilities of concrete cracks, with a mean accuracy (mAP) of up to 99.5%. WU Y et al. [26] presented a lightweight LCANet backbone and a novel efficient prototype mask branch for crack detection based on the YOLOv8 instance segmentation model, reducing model complexity. Specifically, under conditions of 129 frames per second (FPS), the results of the case study showed the accuracy reached 94.5%, while the computational complexity decreased by 51%, compared to the original model.

The improvements made to the YOLO algorithm focus on optimizing the model structure, improving the loss functions and the feature extractors, and optimizing the data pre- and post-processing methods. These improvements have enhanced detection accuracy, speed, and robustness. However, in engineering applications, the underside of concrete bridge structures, where significant forces are applied, is prone to cracking, but these areas often have inadequate lighting conditions. In addition, structural corners and edges are susceptible to defects such as honeycombing and exposed rebar due to casting problems, but these areas have complex backgrounds and weak surface textures. In such cases, the YOLO algorithm can suffer from missed detections and false positives. To address these challenges, the deformable convolutional network attention YOLO (DCNA-YOLO) algorithm based on YOLOv8s is proposed in this study. A multi-branch coordinate attention mechanism (MBCA) is introduced to simultaneously incorporate spatial position information and global information. The attention weights for direction perception, position sensitivity, and global awareness are optimized to comprehensively improve the accuracy of coordinate localization. This effectively highlights features of target defect areas with uneven reflections and weak textures, thereby improving the representation and detection effects of the target detection algorithm. Thus, the balance between detection performance, speed, and model parameter size is achieved using MBCA. Furthermore, a deformable convolution method, named MBCADC, based on MBCA is presented. By embedding MBCA attention, this method improves the adaptability of the deformable convolution to significant illumination changes and complex feature shapes in regions with uneven reflections. As a result, it better accommodates different image structures and texture features. The complexity of the model is reduced, while recall (R) and average precision are improved, and missed detections and false positives are reduced.

The paper is organized as follows. Firstly, an overview of the YOLO algorithm is given for the problem under study, and the improvements of existing methods are compared. Subsequently, an improved framework that incorporates deformable convolution (DC) modules with the multi-branch coordinate attention (MBCA) mechanism is introduced. Finally, this new framework is validated with a dataset containing 4794 damage images and compared with other algorithms. Flowchart of study as shown in Figure 1.

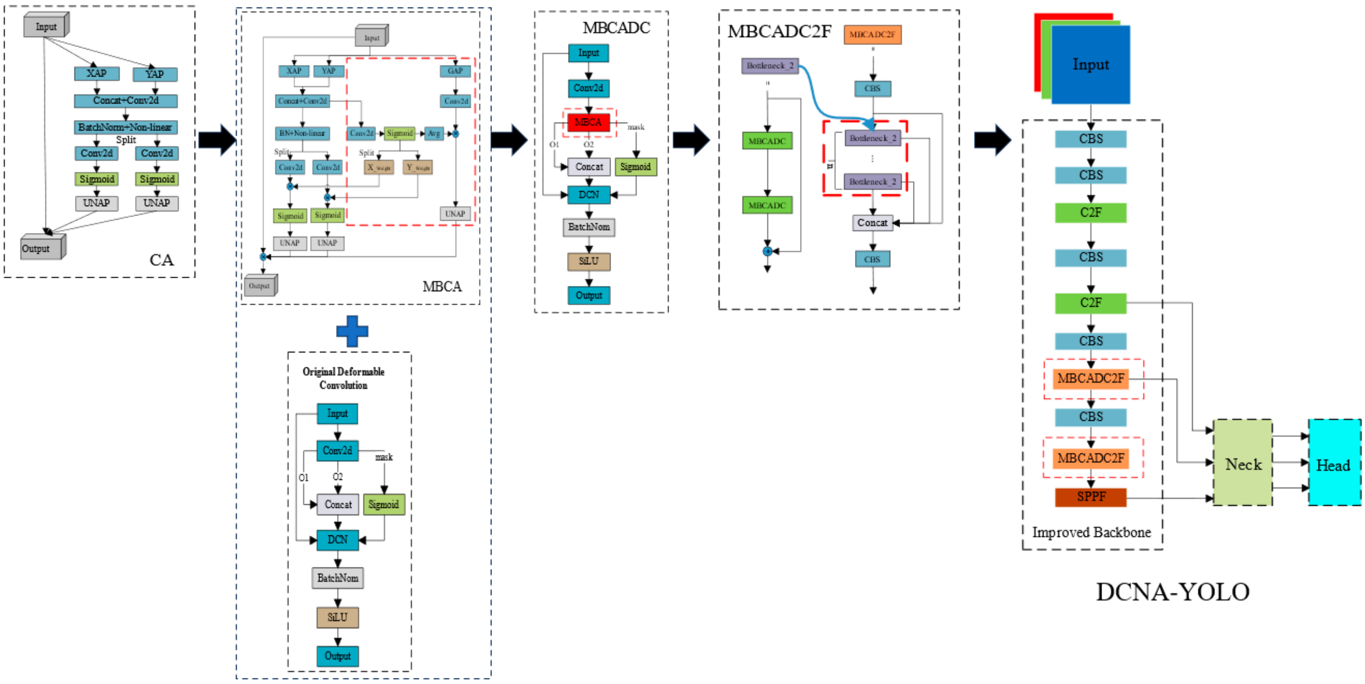


Figure 1. Flowchart of study.

2. Basic Theory of the YOLOv8 Network

YOLOv8s was introduced in 2023 as the latest version of YOLO, which supports image classification, object detection, and instance segmentation tasks. The model structure consists of three main components: backbone, neck, and head, as shown in Figure 2. The received training data are pre-processed using mosaic data augmentation before entering the backbone network for feature extraction. Then, the backbone outputs three feature maps of different scales to the neck structure for bidirectional feature fusion. Finally, the head uses convolutional layers to scale the fused feature maps, producing outputs at three different scales.

The backbone network consists of convolution batch normalization sigmoid linear unit (CBS), cross-stage partial fusion (C2F), and spatial pyramid pooling fast (SPPF) modules [27], where the CBS module is primarily used to extract features from the input image, the C2F module retains lightweight properties while capturing richer gradient flow information, and the SPPF module employs spatial pyramid pooling by serially computing three MaxPool2d operations with 5×5 convolutional kernels. The optimizations made in this paper focus on this component; further information on YOLOv8s can be found in reference [21].

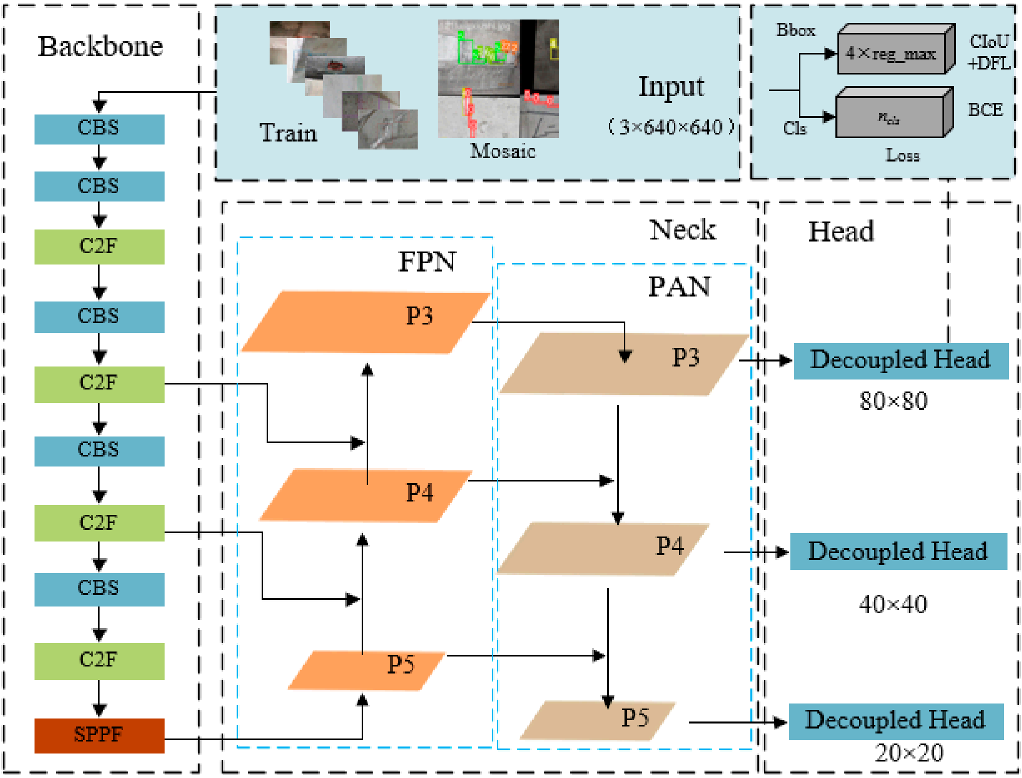


Figure 2. Network structure diagram of YOLOv8s.

3. DCNA-YOLO Method Construction

3.1. Multi-Branch Coordinate Attention

Critical information about objects may be obscured by noise, image backgrounds, and uneven lighting due to complex, blurry, and poorly lit environments. Therefore, enhancing the positional information of features is a significant challenge. To address this issue, attention mechanisms [28] have been introduced in recent years. Among these methods, the coordinate attention mechanism (CAM) [29] effectively enhances the extraction of structural information about objects by using two one-dimensional average pooling operations to aggregate the feature maps vertically and horizontally into two separate orientation-aware feature maps, which are subsequently encoded into an attention tensor containing orientation–position information, and ultimately decomposed into a pair of attention maps that are both orientation- and position-aware.

Although the CAM is effective in capturing long-range dependencies in local spatial information, it overlooks the global dependencies necessary for understanding spatial features. To address this limitation, a global context perception module is introduced into the CAM, aiming to help the network acquire global information by considering the overall context comprehensively. This results in more precise and comprehensive image representations for processing tasks. Additionally, by optimizing attention weights for direction awareness, position sensitivity, and global perception, the network can selectively focus on key areas of the target, significantly improving coordinate localization accuracy.

As shown in Figure 3, the multi-branch coordinate attention (MBCA) principle is described, which consists of two steps:

Step 1: In the information embedding phase, each channel of the input feature map X is encoded using two spatial range pooling kernels: (H, 1) and (1, W) to embed coordinate information. The kernels operate along the horizontal (width W) and vertical (height H) coordinates, respectively.

$$\begin{cases} z_c^h(h) = \dfrac{1}{W} \sum\limits_{0 \le i < W} x_c(h, i) \\ z_c^w(w) = \dfrac{1}{H} \sum\limits_{0 \le j < h} x_c(j, w) \end{cases} \tag{1}$$

where h and w represent the height and width of the current input feature map, respectively. c denotes the current input feature map's channel. $z_c^h(h)$ denotes the output of channel c at height h, and $z_c^w(w)$ denotes the output of channel c at width w.

The global information is additionally embedded in the CAM by encoding each channel of the input feature map X through global average pooling (GAP).

$$Z_c = \frac{1}{h \times w} \sum_{i=1}^{h} \sum_{j=1}^{w} x_c(i, j) \tag{2}$$

where z_c represents the global information output of channel c.

Step 2: At the coordinated and global attention generation phase, the aggregated feature maps Z^h and Z^w generated from Equation (1) are firstly concatenated, and then a 3 × 1 convolution operation is applied for information fusion, compressing feature channels. After batch normalization and non-linear activation functions, the intermediate feature map is split along the spatial dimension into two independent tensors f^h and f^w. Subsequently, two 1 × 1 convolutions F_h and F_w are used to transform f^h and f^w into tensors with the same number of channels as the input X. Furthermore, to fully utilize the expressive representation of the aggregated feature maps and accurately highlight the regions of interest, a feature map optimization module is proposed. This involves passing the intermediate feature map f through a 1 × 1 convolutional transformation function F1 to adjust the number of channels. After applying a non-linear activation function and a sigmoid function, a feature map optimization weight matrix g_1 is obtained. The optimization weight matrix is then split along the spatial dimension into independent weights g_1^h and g_1^w for the vertical and horizontal directions, respectively. Finally, the two independent weights g_1^h and g_1^w are applied to the corresponding tensors, and after passing through a sigmoid function, the outputs g^h and g^w are expanded and used as attention weights.

$$\begin{cases} f = \delta\left(F_{3\times1}\left(\left[Z^h, Z^w\right]\right)\right) \\ g_1 = \sigma(\delta(F_1(f))) \\ g_1^h = F_h(g_1) \\ g_1^w = F_w(g_1) \\ g^h = \sigma\left(F_h\left(f^h\right) \times g_1^h\right) \\ g^w = \sigma\left(F_w(f^w) \times g_1^w\right) \end{cases} \tag{3}$$

where $[\cdot,\cdot]$ represents the concatenation operation along the spatial dimension. $F_{3\times1}(\cdot)$ denotes the 3 × 1 convolution transformation function, $\delta(\cdot)$ represents the non-linear activation function hard_swish. f represents the intermediate feature map with horizontal and vertical spatial information, where $f \epsilon R^{C/r\times(W+H)}$. r is the reduction ratio, taken as 16. F_h and F_w represent 1 × 1 convolution operations in the vertical and horizontal directions, respectively. f^h and f^w represent intermediate feature maps in the vertical and horizontal directions, where $f^h \epsilon R^{C/r\times1\times H}$ and $f^w \epsilon R^{C/r\times1\times W}$. σ denotes the sigmoid activation function. g_1 represents the feature map optimization weight matrix. g_1^h and g_1^w represent optimization weights in the vertical and horizontal directions, respectively. g^h and g^w represent attention weights in the vertical and horizontal directions, respectively.

For global attention generation, the global information feature map generated from Equation (2) is multiplied element-wise by the average-weighted optimization weight matrix g_1. Subsequently, the result passes through a sigmoid function to obtain the global attention weights g^G.

$$\begin{cases} f_c = \delta(F_1(Z_c)) \\ g^G = \sigma(mean(g_1) \cdot f_c) \end{cases} \tag{4}$$

where $F_1(\cdot)$ represents the 1×1 convolution transformation function. $mean(\cdot)$ represents the mean function. f_c represents the intermediate feature map with global information, where $f_c \epsilon R^{C\times1\times1}$. g_1 represents the feature map optimization weight matrix. g^G represents the global attention weights.

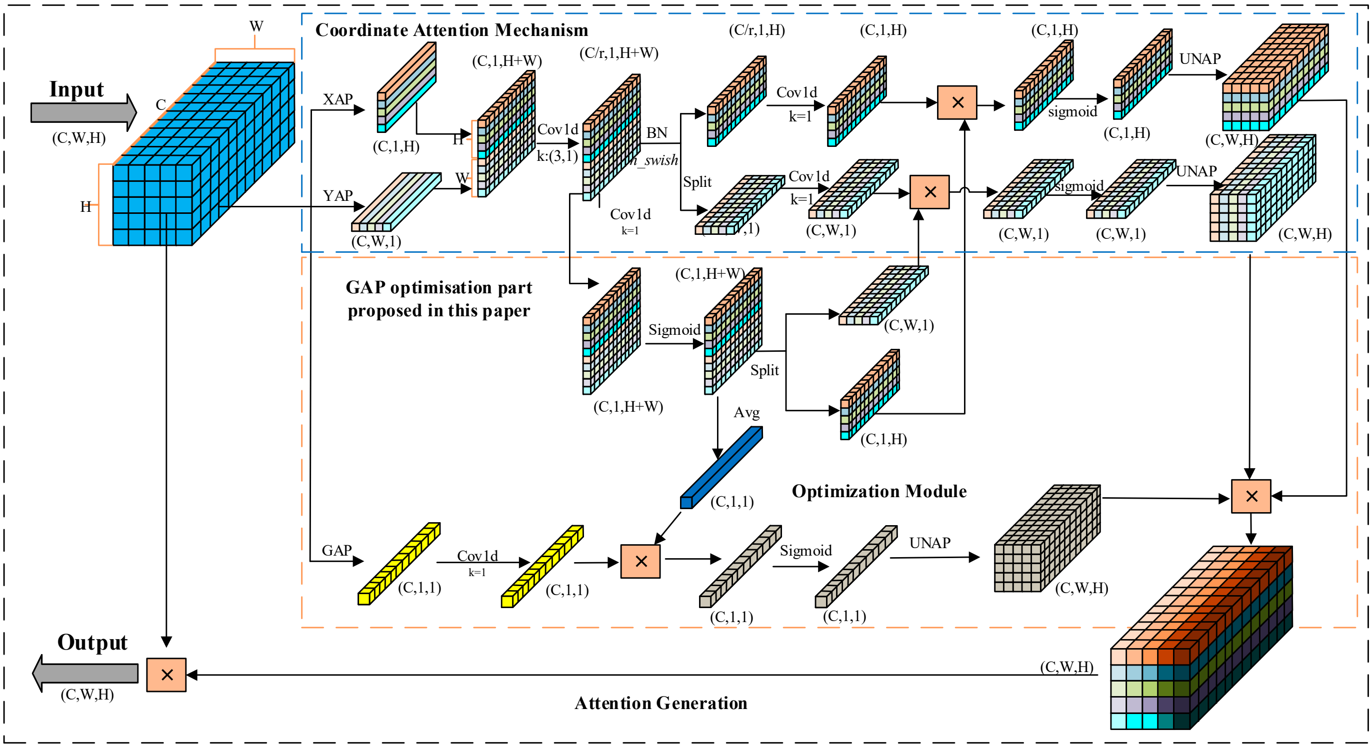


Figure 3. The principle of the multi-branch coordinate attention (MBCA) algorithm.

The attention weights g are calculated by multiplying the vertical and horizontal direction attention weights by the global attention weights.

$$g_c = g_c^h(i) \times g_c^w(j) \times g_c^G(i,j) \tag{5}$$

where g_c represents the attention weight at channel c. g_c^h represents the vertical direction attention weight at channel c. g_c^w represents the horizontal direction attention weight at channel c. g_c^G represents the global attention weight at channel c.

It computes the average of all elements within each feature map of the input, resulting in a feature map with a size of 1×1. When direct multiplication or addition operations are needed for the original input, reverse average pooling (UNAP) can be used to expand the feature map to the desired size. The specific pooling operations are illustrated in Figure 4.

The multi-branch coordinate attention (MBCA) mechanism is established by the two steps above, introducing global-level information and optimizing the perception of specific target positions at the local level.

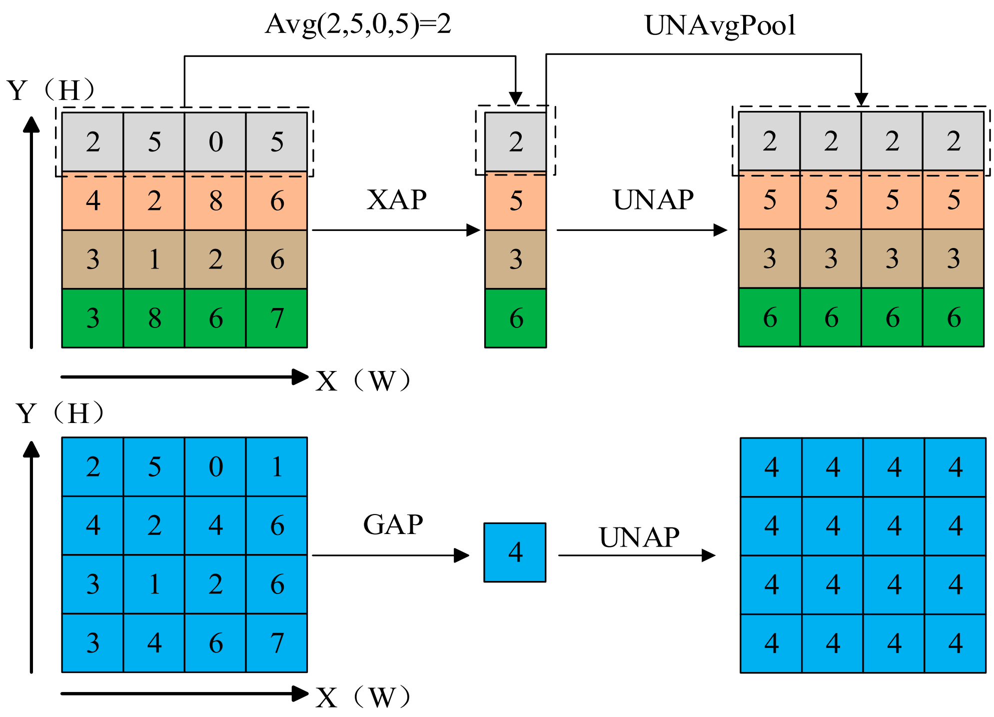


Figure 4. Schematic diagram of XAP, GAP, and UNAP.

3.2. Deformable Convolution Based on MBCA

The features of the image become more complex and irregular due to weak texture or lighting changes on the target surface with the detection of weak texture areas and uneven reflection areas, making the fixed receptive field kernel insufficient to adapt complex features. To address this issue, deformable convolution (DCNv2) [30] is applied to learn the offset and modulation parameters for each pixel to better adjust to the sampling position of the convolution kernel and to adapt to different image structures and texture features.

For each position p_0 in the output feature map y, the deformable convolution structure is defined as follows:

$$y(P_0) = \sum_{p_n \in R} w(p_0) \cdot x(p_0 + p_n + \Delta p_n) \cdot \Delta m_n \quad (6)$$

where the grid R defines the size and expansion rate of the receptive field. For a receptive field of size 3×3 and a dilation rate of 1, $R = \{(-1,-1),(1,0)\cdots(0,1),(1,1)\}$. p_0 corresponds to mapping each point of the output feature map y to the center of the convolution kernel, and then mapping it to the coordinates in the input feature map x; p_n represents the relative coordinates in R for p_0. $\omega(\cdot)$ denotes the sampling point weight, and $x\,(\cdot)$ denotes the mapping of the coordinates in the input feature map x to feature vectors. The offset $\{\Delta p_n | n = 1, 2 \cdots N\}$, $N = |R|$, and the modulation parameter Δm_n lies within [0, 1].

The offsets Δp_n and modulation parameters Δm_n of a deformable convolution are obtained by applying a separate standard convolution layer to the same input feature map, which results in insufficient spatial support range. As a consequence, the effective receptive field of foreground nodes and the prominent region constrained by errors may include background areas irrelevant to detection. The proposed multi-branch coordinate attention (MBCA) is embedded during the process of generating the offsets Δp_n and modulation parameters Δm_n, which is named MBCADC, to further enhance the ability of deformable convolution DCNv2 to manipulate spatial support regions, as illustrated in Figure 5.

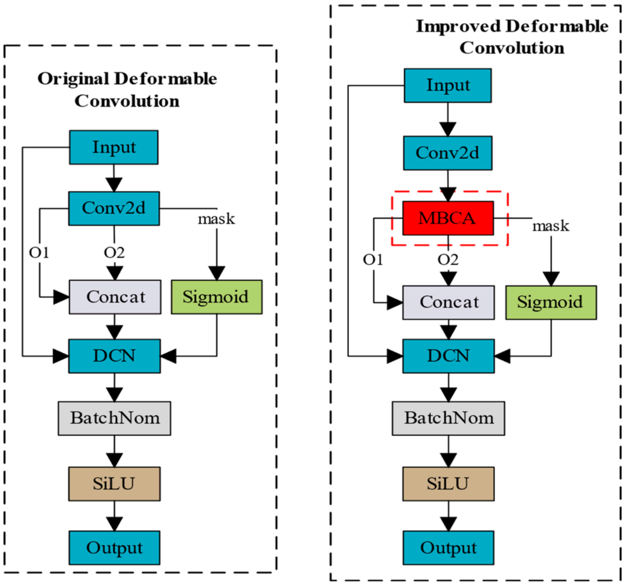


Figure 5. Structure of original deformable convolution (DCNv2) and improved deformable convolution (MBCADC).

The MBCADC module consists of a Conv2d, MBCA attention, a DCN, BatchNorm2d, and a SiLU activation function, where o1 and o2 represent learned offsets in the x- and y-coordinate directions, respectively, and mask denotes the sampling weights at different positions. The structure of the deformable convolution MBCADC is defined as follows:

$$y(P_0) = \sum\nolimits_{p_n \in R} w(p_0) \cdot x(p_0 + p_n + \Delta p_n \cdot g) \cdot \Delta m_n \cdot g \tag{7}$$

where g represents the attention weights generated by multi-branch coordinate attention (MBCA).

Figure 6 illustrates the process of MBCADC deformable convolution. From Figure 6, it can be observed that the sampling matrix of deformable convolution is non-fixed and deformable, with the offsets determined by algorithms that can better learn the geometric properties of the objects to be detected.

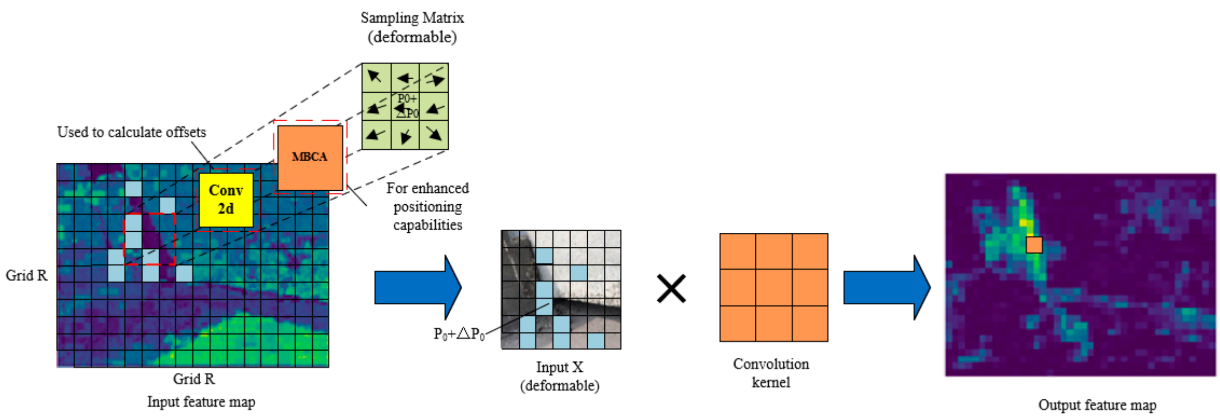


Figure 6. Process of MBCADC deformable convolution.

3.3. MBCADC2F Module

The MBCAC2F module is an improvement over the YOLOv8s backbone network's C2F module, integrating the multi-branch coordinate attention deformable convolution (MBCADC) module. This module comprises two CBS modules and n Bottleneck modules, where the Bottleneck module contains a residual structure with two MBCADC modules, as illustrated in Figure 7. By learning the parameters of deformable convolution, the model can dynamically adjust the sampling positions of the convolution kernel based on the actual shape and positional information of the target, allowing for more precise capture of target features.

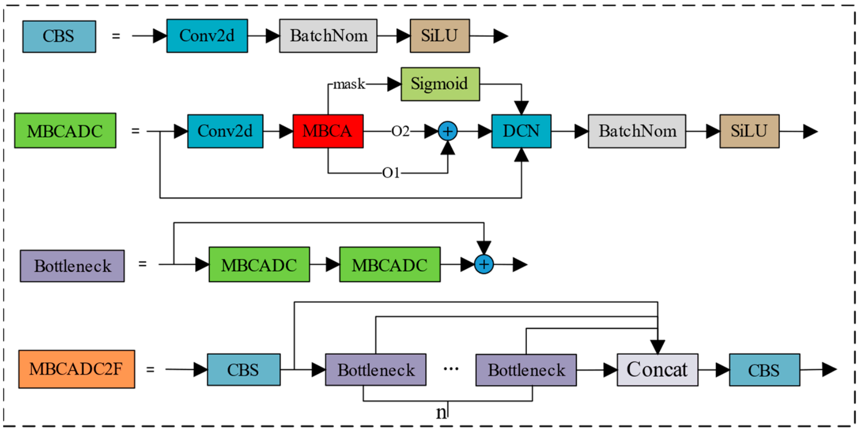


Figure 7. The structure of the MBCADC2F module.

3.4. Deformable Convolutional Network Attention YOLO Object Detection Network

The deformable convolutional network attention YOLO (DCNA-YOLO) algorithm improves upon the YOLOv8s baseline model by modifying the backbone network. The neck and head structures of the DCNA-YOLO model remain the same as those of the baseline model. The model's overall structure is illustrated in Figure 8. The DNCA-YOLO backbone network comprises five CBS modules, two C2F modules, two MBCADC2F modules, and one SPPF module. The structures of modules such as CBS, C2F, and SPPF in the MBCADC2F module are identical to those of the corresponding modules in the YOLOv8s baseline model. Each Bottleneck unit in the MBCADC2F module contains a residual connection structure with two MBCADC modules.

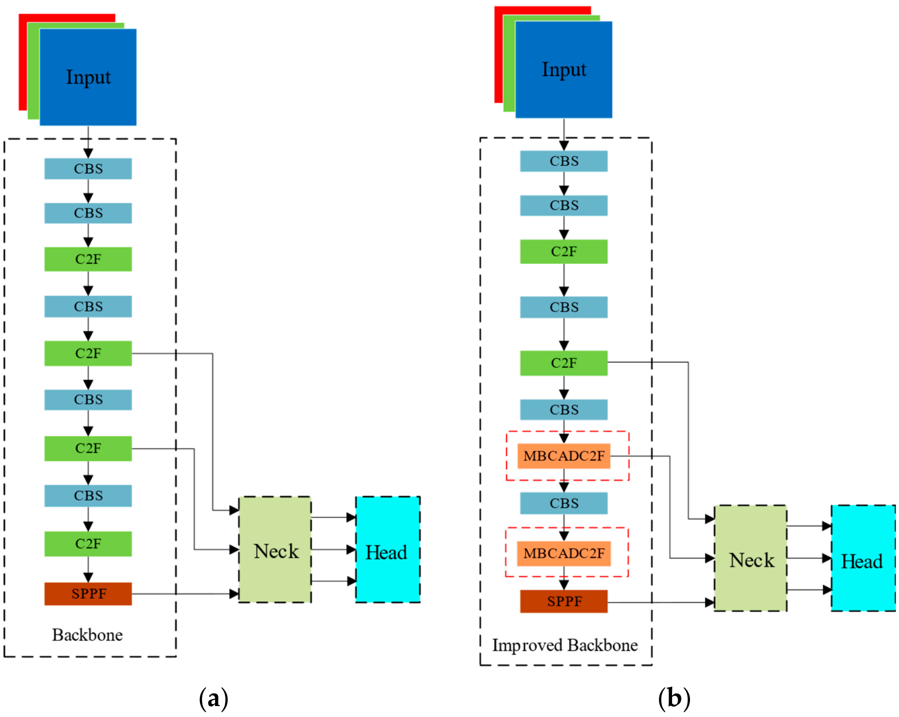


Figure 8. Original YOLOv8s and proposed DCNA-YOLO overall framework. (**a**) Original YOLOv8s overall framework; (**b**) proposed DCNA-YOLO overall framework.

4. Example Verification

4.1. Experimental Dataset

We created a dataset of apparent damage to concrete bridges, consisting of 4794 images. Domain experts annotated the dataset, which we used to evaluate the effectiveness of the proposed method in identifying apparent damage to concrete bridges. According to the regulations of China's road and bridge maintenance standards and inspection standards, apparent concrete damage was classified into seven types: cracks, spalling, honeycombing, exposed reinforcement, water seepage, and voids. The constructed dataset included at least one type of damage in each image. Augmenting the dataset enhances its diversity and richness, making the model's detection more effective. The original dataset was randomly divided into training, validation, and testing sets with a ratio of 8:1:1. Data augmentation techniques, such as flipping, rotation, and HSV (hue, saturation, and value) enhancement, were then applied to the divided dataset. The dataset sizes were as follows: 14,528 images in the training set, 1816 images in the validation set, and 1816 images in the testing set. Table 1 shows the statistical results of the number of annotated boxes for each type of damage.

Table 1. Number of Labels for Each Damage in the Dataset.

Labels (Damage)	Number	Labels (Damage)	Number
liefeng (crack)	17,636	shenshui (seepage)	9244
boluo (spalling)	11,875	fengwo (comb surface)	8330
kongdong (cavity)	7082	lujin (steel exposed)	6584
mamian (pockmark)	8274		

The images of the bridge's apparent damage collected in the experimental dataset were affected by the lighting environment, resulting in variations in image quality. Statistical analysis was conducted on the grayscale histograms of the images, which allowed for the categorization of the images into four lighting conditions:

(1) Well-lit images, which exhibit high clarity and rich details. The histogram of the grayscale image displays a unimodal distribution, with grayscale values primarily ranging from 150 to 250. The average grayscale value of the image is greater than 170.

(2) Partial shadow or occlusion images: The grayscale distribution is complex, with areas of varying brightness. The histogram of the grayscale image displays a bimodal distribution, with grayscale values primarily ranging from 50 to 100 and from 150 to 250. The image's average grayscale value is approximately 150.

(3) Low-lighting images: The overall low brightness can result in blurry or confusing areas in the apparent damaged regions. The grayscale histogram of the image displays a unimodal distribution, with grayscale values primarily distributed between 50 and 100. The image's average grayscale value is approximately 100.

(4) Dark-lighting images: The overall low brightness can make it challenging to distinguish details of the apparent damage. The image's grayscale histogram displays a unimodal distribution, with grayscale values ranging from 0 to 50. The average grayscale value of the image is below 30.

Figure 9 shows the grayscale histograms for the four distinct illumination conditions. Table 2 presents a statistical study of the number of well-lit images, partial shadow or occlusion images, low-lighting images, and dark-lighting images. Figure 10 illustrates some examples of picture data.

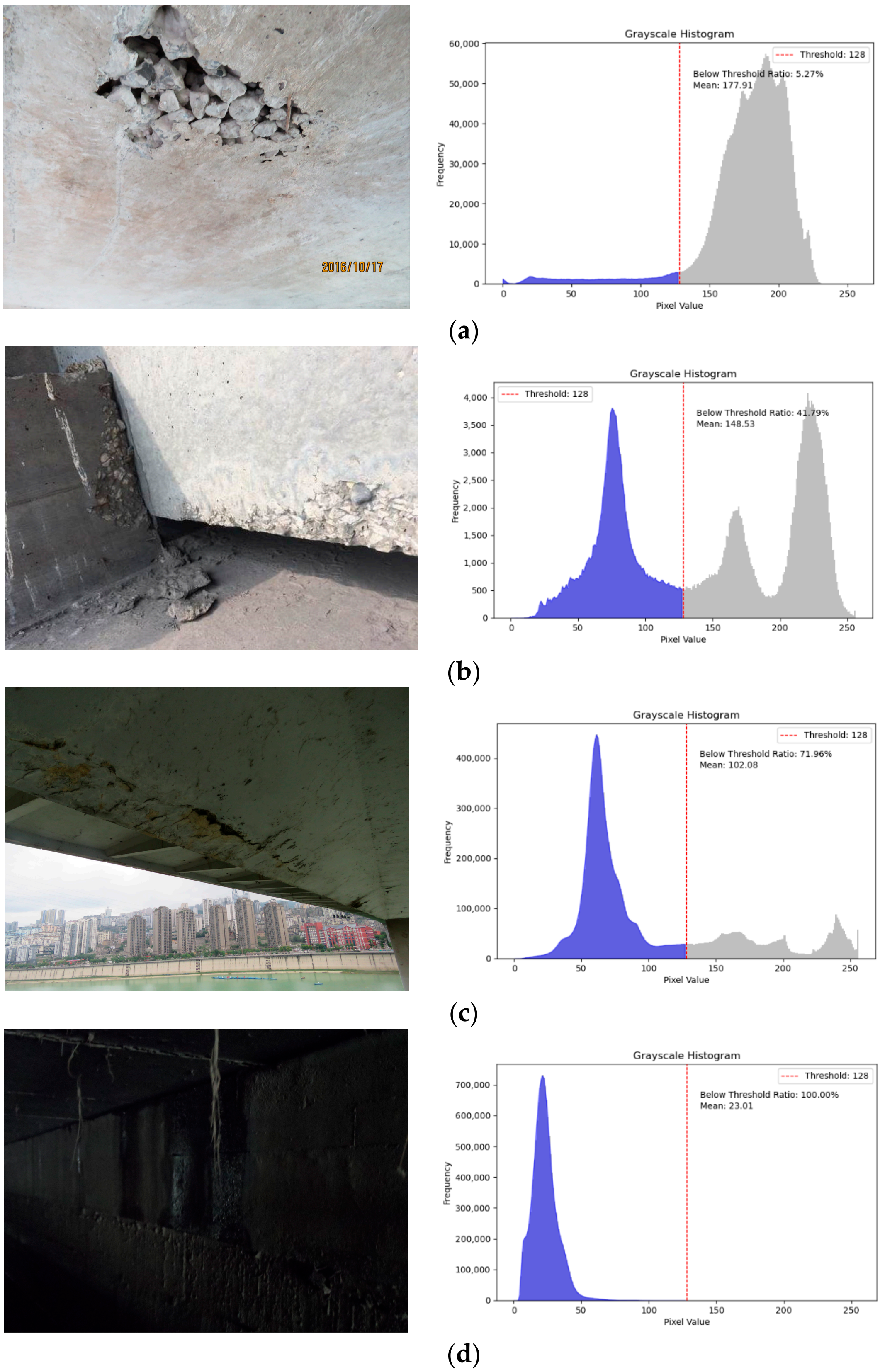


Figure 9. Images and grayscale histograms captured under varying lighting conditions. (**a**) Well-lit image and its gray histogram; (**b**) partial shadow or occlusion image and its gray histogram; (**c**) low-lighting image and its gray histogram; (**d**) dark-lighting image and its gray histogram.

Table 2. Summary of Image Quantities Under Different Illumination Conditions.

Data Set	Well-Lit Images	Partial Shadow or Occlusion Images	Low-Lighting Images	Dark-Lighting Images	Total
Train	6697	1798	2608	3425	14,528
Val	923	239	298	356	1816
Test	876	225	326	389	1816

(**a**)

(**b**)

(**c**)

(**d**)

Figure 10. Partial images of the training set. (**a**) Examples of well-lit images; (**b**) examples of partial shadow or occlusion images; (**c**) examples of low-lighting images; (**d**) examples of dark-lighting images.

4.2. Environmental Design and Evaluation Metrics

The computer hardware setup included an Intel Core i5-13600K CPU, 48 GB of RAM, and an NVIDIA GeForce RTX4070 with a 12,282 Mib GPU. The experimental environment consisted of Windows 10, CUDA 11.8, PyTorch 2.0.1, and Python 3.8.18. The training parameter settings had an initial learning rate of 0.01, with a learning rate strategy that employed cosine annealing. The number of training epochs was set to 300, with an initial input size of the model at 640 × 640 and a batch size of 16. The optimization algorithm used was SGD [31], with the loss function being cross-entropy. To prevent overfitting, early stopping criteria were employed. The network training was halted if the validation accuracy did not improve after 50 epochs.

To evaluate the model's detection performance for seven types of visual damage, we used precision (P), recall (R), mean average precision (mAP), model parameter quantities (parameters), floating-point operations (FLOPs), and frames per second (FPS) as the evaluation metrics.

4.3. Experimental Results and Analysis

4.3.1. Ablation Experiment

To further validate the effectiveness of the proposed improvements in terms of the number and placement of different enhancement modules, ablation experiments and quantitative and qualitative analyses were performed using the generated dataset in the same experimental environment to evaluate the benefits of key components in the model. Among them, "MBCA" refers to the addition of MBCA attention after the last layer of the backbone network SPPF; "DCNv2" refers to the replacement of the Bottleneck in the C2F module of the eighth layer of the backbone network with deformable convolution DCNv2; "Proposed method" refers to replacing the C2F module of the sixth and the eighth layer of the backbone network with the MBCADC2F module proposed in this paper. The experimental results are presented in Table 3.

Table 3. Results of ablation experiments.

Model	Parameters/M	FLOPs/G	FPS/f·s^{-1}	P/%	R/%	mAP$_{0.5}$/%	mAP$_{0.5:0.95}$/%
YOLOv8s	11.1	28.7	70.9	89.1	82.0	87.4	68.9
+MBCA	11.2	28.7	68.9	90.2	82.7	87.9	68.9
+DCNv2	11.2	27.5	73.8	90.0	82.5	88.1	70.2
proposed method	11.3	27.5	74.4	91.3	85.4	89.4	73.3

A comparison of the data in the table shows that the MBCA attention mechanism proposed in this study did not significantly increase the number of network parameters or model complexity, but improved the model's mAP0.5 value by 0.5%. The deformable convolution DCNv2 was able to reduce the model complexity and improve the model mAP0.5 without significantly increasing the model parameter count. The method proposed in this paper achieved the best experimental results, with only a 0.2 M increase in model parameters compared to the baseline model. Furthermore, floating-point operations were reduced by 1.2G, precision increased by 1.8%, recall improved by 3.4%, and mAP0.5 and mAP0.5:0.95 increased by 2.0% and 4.4%, respectively. In addition, the model's speed of detection increased by 3.5/f·s^{-1}.

Figure 11 shows a comparison between the test results and the heatmap analysis of the baseline model and the proposed method for well-lit images. The heatmaps were generated using the Grad-CAM method. They show that both models achieved good detection results for all seven types of surface defects in images with good lighting conditions. The proposed method detected all defects, while the baseline model missed a small piece of exposed rebar (fourth-row image). Furthermore, the accuracy of the detection results obtained by the proposed method was consistently higher than that of the baseline model. A comparison of the heatmaps shows that the proposed method provided a better representation, with the heatmaps better conforming to the shape of the target. These results indicate that the proposed method was effective in improving the accuracy of detecting images with good lighting conditions.

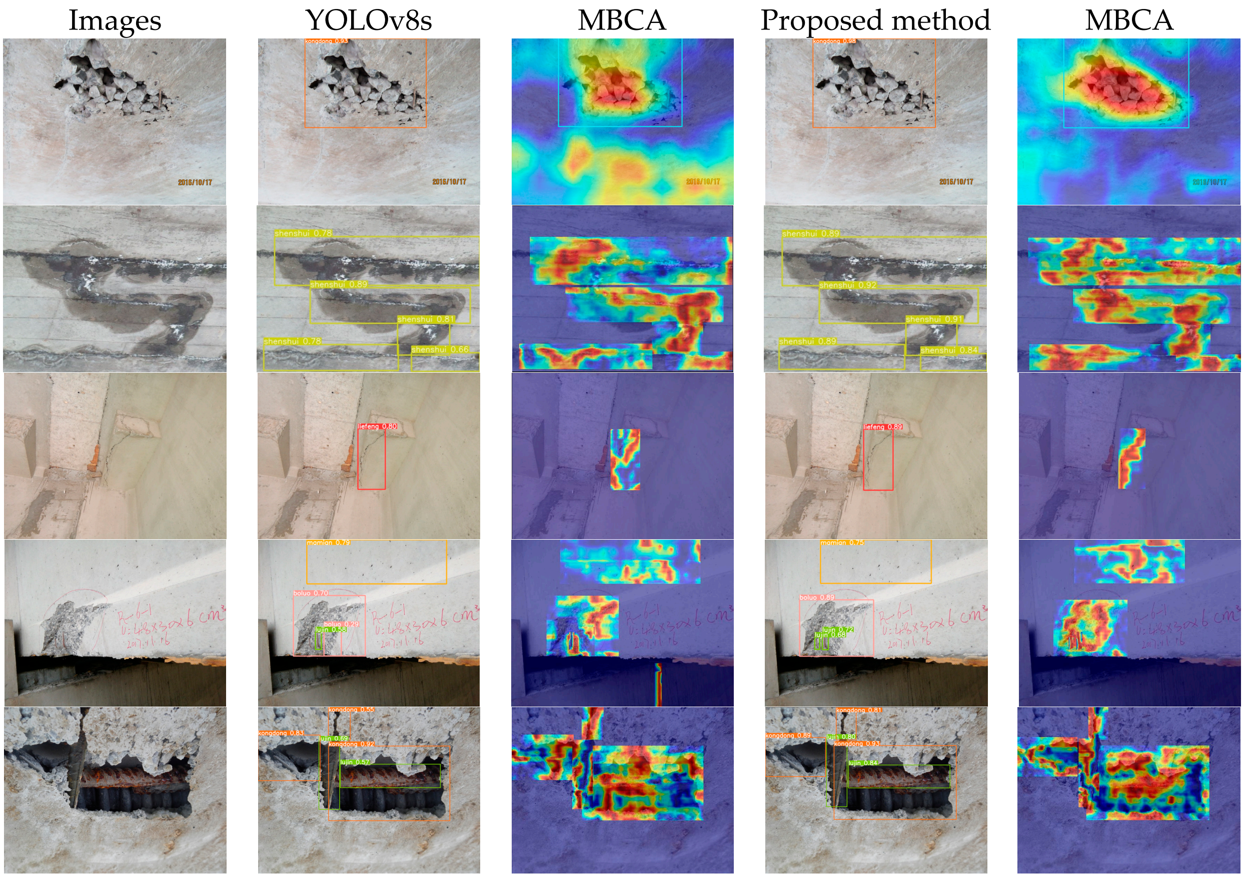


Figure 11. Detection results and heatmaps of seven types of damage in well-lit images.

Figure 12 shows a comparison between the test results and the heatmap analysis of the baseline model and the proposed method for images with partial shadows or occlusions. It can be seen that the proposed method achieved better detection results than the baseline model for images with partial shadows or occlusions. In images with partial shadows or occlusions, there were areas of varying brightness due to significant changes in illumination and uneven reflections on the target surface. The baseline model with fixed geometric structures of convolutional kernels was not effective in capturing the spatial information of the target in these regions. The proposed method used deformable convolution based on multi-branch coordinate attention, which allowed the model to dynamically adjust the sampling positions of the convolutional kernels according to the actual shape and position information of the target. A comparison of the heatmaps shows that the proposed method provided a better representation, with the heatmaps focusing more on the edge features of the target, which effectively reduced the rates of missed detections and false positives in images with partial shadows or occlusions.

Figure 13 shows the comparison between the test results and the heatmap analysis of the baseline model and the proposed method. It can be seen that both the baseline model and the proposed method could detect the class and location of defects in the image under low-light conditions. However, the detection accuracy of the proposed method was higher compared to the baseline model. The texture of the target region in the image may have been relatively weak, resulting in the lower detection accuracy of the model. A comparison of the heatmaps shows that the proposed method could effectively highlight the features of the defective region, thus improving the detection accuracy of the model.

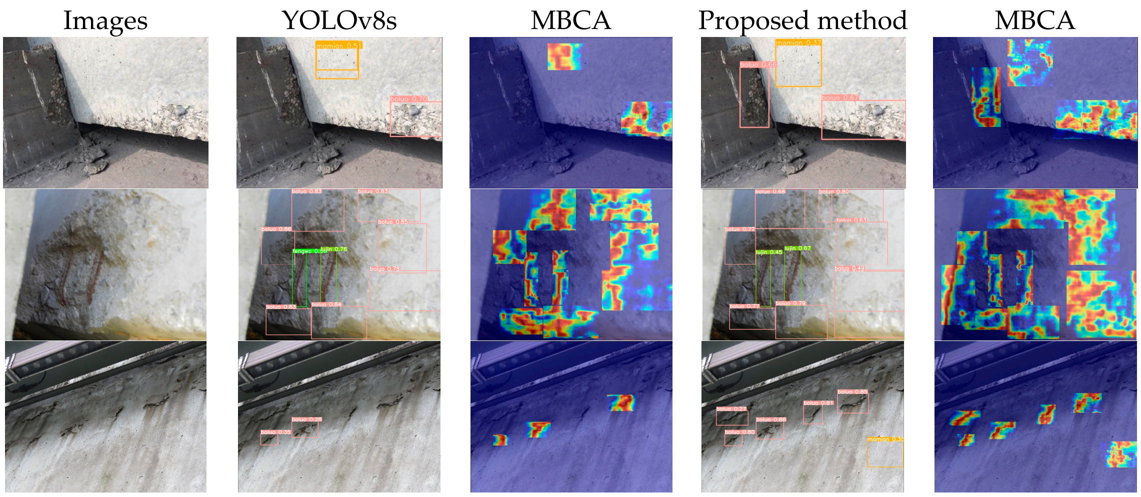


Figure 12. Detection results and heatmaps of damage in partially shaded or occluded images.

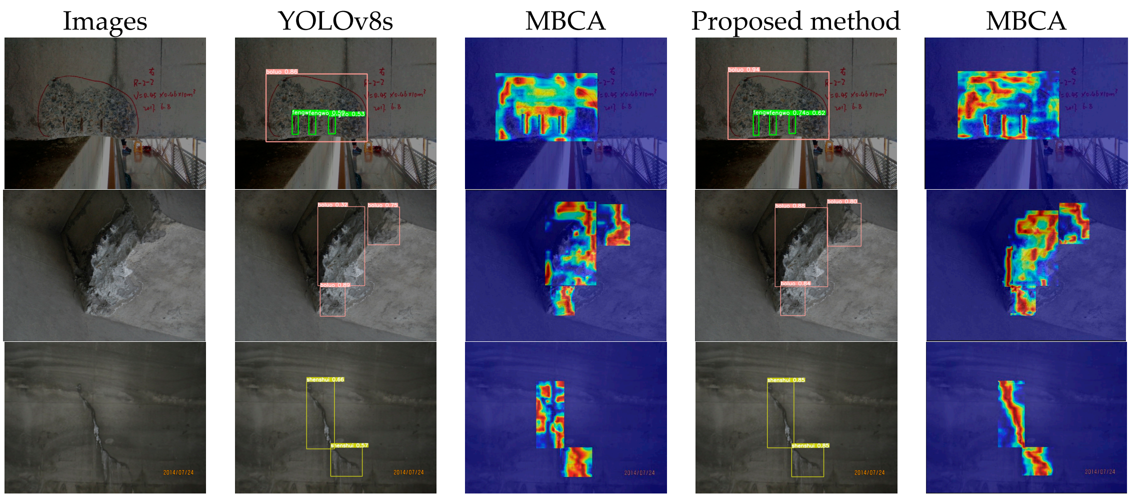


Figure 13. Detection results and thermograms of damage in low-light images.

Figure 14 illustrates a comparison between the test results and the heatmap analysis of the baseline model and the proposed method for images with dark lighting conditions. From Figure 14, it can be seen that both the baseline model and the proposed method performed poorly on images with dark lighting conditions. In images with dark lighting conditions, the overall brightness was low, making it difficult to detect surface defect details. The models failed to learn useful information from the images, resulting in incorrect target category detection or no defect detection.

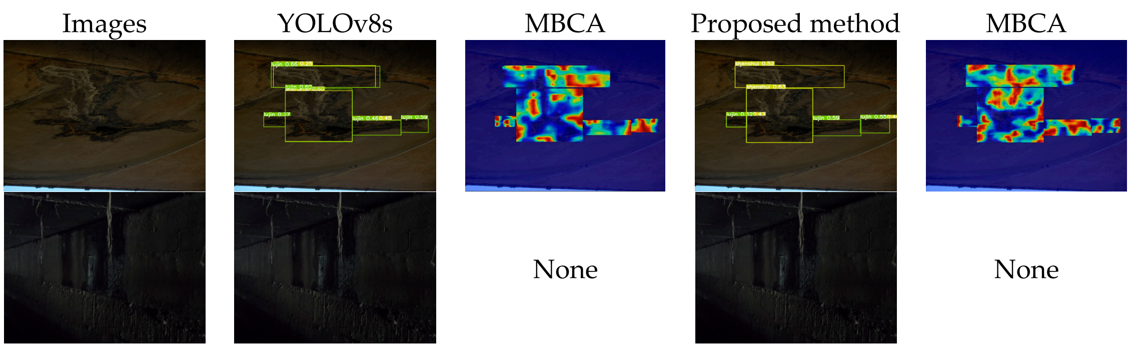


Figure 14. Detection results and thermograms of damage in dark and light images.

In summary, the proposed method effectively improved the accuracy of detecting images with good lighting conditions and those with poor lighting conditions, effectively mitigating the problems of missed detections and false positives.

4.3.2. Comparison Experiment of Different Detection Algorithms

Considering the real-time performance and accuracy requirements for concrete bridge surface defect detection tasks, the two-stage object detection models of the RCNN series and the outdated SSD models were not included in the comparative experiments. Instead, more widely used and advanced models from the YOLO series were selected as the benchmark models. The results are presented in Table 4.

Table 4. Comparative experimental results of different network models.

Model	Parameters/M	FLOPs/G	FPS/f·s^{-1}	P/%	R/%	mAP$_{0.5}$/%	mAP$_{0.5:0.95}$/%
YOLOv3-tiny	12.1	19.1	76.9	81.7	73.4	78.6	53.4
YOLOv5s	7.0	16.8	78.3	91.2	84.9	88.7	67.4
YOLOv6s	16.3	44.2	69.4	90.2	81.5	87.7	69.6
YOLOv8s	11.1	28.7	70.9	89.1	82.0	87.4	68.9
Proposed method	11.3	27.5	74.4	91.3	85.4	89.4	73.3

Comparing the data in Table 4, it is evident that the model proposed in this paper exhibited more effective performance in terms of detection accuracy compared to the current state-of-the-art (SOTA) models. The mAP0.5 value was improved by 11.1%, 0.7%, and 1.7% compared to the classical YOLOv3-tiny, YOLOv5s, and YOLOv6s models, respectively. Compared to the baseline model YOLOv8s, the proposed model achieved a 2.0% and 4.4% improvement in average precision (mAP0.5 and mAP0.5:0.95, respectively), a 3.4% increase in recall rate, an increase of 3.5/f·s^{-1} in detection speed, and a reduction in model complexity by 1.2G. These results demonstrate that the proposed model had better detection performance in concrete bridge surface defect detection.

5. Conclusions

The current work adopts a novel object detection algorithm termed DCNA-YOLO based on multi-channel attention mechanisms and deformable convolutions. It is proposed to address problems such as missed detections, false positives in regions with insufficient illumination, complex backgrounds, and weak surface textures on concrete bridge surfaces. The major findings of this work are concluded below:

(1) Multi-branch coordinate attention (MBCA) is adopted on the basis of CA with a supplementation of the global information branch. MBCA is applied to obtain spatial coordinate information and global information simultaneously, which improves the accuracy of coordinate information in the attention mechanism.

(2) The MBCA mechanism is embedded with a deformable convolution, then used to enhance the adaptability of the convolution kernel. This novel coordinated model contributes the coordinate localization ability for better adaptation to different image structures and texture features.

(3) The proposed framework (Figure 8) is validated through a self-constructed concrete surface defect dataset. Our results effectively highlight the accuracy of detecting regions with significant light variations, uneven reflections, and weak textures without increasing the complexity. All of these mitigate the problems of missed detections and false alarms.

(4) The next plan is to prune and distill the knowledge of the DCNA-YOLO model to develop a lighter model that can be deployed on resource-constrained concrete bridge health inspection drones for efficient real-time inspection. This will improve the safety, efficiency, and accuracy of the inspection and provide a scientific basis for the maintenance and management of concrete bridges.

Author Contributions: Conceptualization, T.L.; methodology, T.L.; software, T.L.; validation, T.L. and S.T.; formal analysis, G.L.; investigation, G.L.; resources, G.L.; data curation, T.L.; writing—original draft preparation, T.L.; writing—review and editing, S.T.; visualization, T.L.; supervision, G.L.; project administration, G.L.; funding acquisition, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (52221002, 52078084), Fundamental Research Funds for the Central Universities (2023CDJKYJH093), and the 111 project of the Ministry of Education and the Bureau of Foreign Experts of China (Grant No. B18062).

Data Availability Statement: The data used in this paper can be obtained through the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Xing, S.; Ye, J.; Jiang, C. Review the study on typical diseases and design countermeasures of China concrete curved bridges. In Proceedings of the 2010 International Conference on Mechanic Automation and Control Engineering, Wuhan, China, 26–28 June 2010; pp. 4805–4808.
2. Ni, F.; Zhang, J.; Chen, Z. Zernike-moment measurement of thin-crack width in images enabled by dual-scale deep learning. *Comput. Aided Civ. Infrastruct. Eng.* **2019**, *34*, 367–384. [CrossRef]
3. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
4. Cha, Y.-J.; Choi, W.; Büyüköztürk, O. Deep learning-based crack damage detection using convolutional neural networks. *Comput. Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [CrossRef]
5. Ni, F.; Zhang, J.; Chen, Z. Pixel-level crack delineation in images with convolutional feature fusion. *Struct. Control Health Monit.* **2019**, *26*, e2286. [CrossRef]
6. Dung, C.V.; Anh, L.D. Autonomous concrete crack detection using deep fully convolutional neural network. *Autom. Constr.* **2019**, *99*, 52–58. [CrossRef]
7. Ali, R.; Cha, Y.-J. Subsurface damage detection of a steel bridge using deep learning and uncooled micro-bolometer. *Constr. Build. Mater.* **2019**, *226*, 376–387. [CrossRef]
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
9. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016, Lecture Notes in Computer Science*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
10. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
11. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
12. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
13. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef] [PubMed]
14. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
15. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
16. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
17. Ultralytics YOLOv5. Available online: https://github.com/ultralytics/yolov5 (accessed on 20 August 2023).
18. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
19. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
20. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
21. Ultralytics YOLOV8. Available online: https://github.com/ultralytics/ultralytics (accessed on 12 October 2023).
22. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. TOOD: Task-aligned One-stage Object Detection. In Proceedings of the International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021.
23. Zhang, X.; Luo, Z.; Ji, J.; Sun, Y.; Tang, H.; Li, Y. Intelligent Surface Cracks Detection in Bridges Using Deep Neural Network. *Int. J. Struct. Stab. Dyn.* **2023**, *24*, 2450046. [CrossRef]
24. Yu, Z. YOLO V5s-based Deep Learning Approach for Concrete Cracks Detection. *SHS Web Conf.* **2022**, *144*, 03015. [CrossRef]

25. Jin, Q.; Han, Q.; Su, N.; Wu, Y.; Han, Y. A deep learning and morphological method for concrete cracks detection. *J. Circuits Syst. Comput.* **2023**, *32*, 2350271. [CrossRef]
26. Wu, Y.; Han, Q.; Jin, Q.; Li, J.; Zhang, Y. LCA-YOLOv8-Seg: An Improved Lightweight YOLOv8-Seg for Real-Time Pixel-Level Crack Detection of Dams and Bridges. *Appl. Sci.* **2023**, *13*, 10583. [CrossRef]
27. Elfwing, S.; Uchibe, E.; Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.* **2018**, *107*, 3–11. [CrossRef] [PubMed]
28. Tsotsos, J.K. *A Computational Perspective on Visual Attention*; The MIT Presse Books: Cambridge, MA, USA, 2011.
29. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 15–25 June 2021.
30. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable ConvNets v2: More Deformable, Better Results. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
31. Robbins, H.; Monro, S. A Stochastic Approximation Method. *Ann. Math. Stat.* **1951**, *22*, 400–407. [CrossRef]

applied sciences

MDPI

Article

Time Series Feature Selection Method Based on Mutual Information

Lin Huang [1], Xingqiang Zhou [2], Lianhui Shi [1] and Li Gong [1,*]

[1] Ship Comprehensive Test and Training Base, Naval University of Engineering, Wuhan 430033, China
[2] 91251 Army of PLA, Shanghai 200940, China
* Correspondence: 13971276201@139.com

Abstract: Time series data have characteristics such as high dimensionality, excessive noise, data imbalance, etc. In the data preprocessing process, feature selection plays an important role in the quantitative analysis of multidimensional time series data. Aiming at the problem of feature selection of multidimensional time series data, a feature selection method for time series based on mutual information (MI) is proposed. One of the difficulties of traditional MI methods is in searching for a suitable target variable. To address this issue, the main innovation of this paper is the hybridization of principal component analysis (PCA) and kernel regression (KR) methods based on MI. Firstly, based on historical operational data, quantifiable system operability is constructed using PCA and KR. The next step is to use the constructed system operability as the target variable for MI analysis to extract the most useful features for the system data analysis. In order to verify the effectiveness of the method, an experiment is conducted on the CMAPSS engine dataset, and the effectiveness of condition recognition is tested based on the extracted features. The results indicate that the proposed method can effectively achieve feature extraction of high-dimensional monitoring data.

Keywords: time series; feature extraction; mutual information; system operability; condition identification

1. Introduction

Time series data mining has extensive and important applications in the field of machine learning, such as data classification, clustering, or prediction, which can help explore the potential patterns in time series data that are useful for subsequent research. Currently, most researchers mainly focus on the processing of univariate time series. However, with the development of data collection technology, multidimensional time series data have become increasingly common and contain a large amount of potentially valuable information. Feature selection is an important step in data processing of high-dimensional data (such as regression and classification). Its main function is to reduce computational complexity, avoid the "curse of dimensionality" problem [1], reduce training time, and improve the performance of the predictor [2]. Therefore, how to effectively extract system features is one of the key issues in the field of time series analysis [3], which has been widely used in the following fields: image recognition [4,5], natural language processing [6,7], data mining [8,9], fault diagnosis [10–12], remaining useful life prediction [13,14], microbes classification [15], fatigue detection [16], image classification [17], intrusion detection [18–21], etc.

Feature selection methods can be roughly divided into three categories [22] including filter, wrapper, and embedded, according to their relationship with learning methods [23,24]. On the other hand, according to the utilized training data, feature selection methods can be divided into supervised [25], unsupervised, and semi-supervised models [26]. In this paper, we focus on the feature selection of time series. At present, the feature selection of time series, in most cases, is mainly conducted by observing the variable trends, amplitude, noise, and other characteristics [27]. This kind of method has problems such as strong subjectivity, inability to conduct quantitative analysis, and inaccuracy.

To solve these problems, we propose a time series feature extraction method based on mutual information. Mutual information is a measure of the interdependence between two variables, which indicates how much information is shared between two variables [28]. The greater the mutual information between variables, the stronger the correlation. Firstly, we determine the correlation between features and then select features with a strong correlation with the target variable. This is an unsupervised feature selection method and has become an important feature selection method [29]. In combination with other methods, MI-based feature selection has derived many other methods, such as MI with a correlation coefficient [30], variance impact factor [31], fisher score [32], binary butterfly optimization algorithm [33], conditional mutual information [34], deep neural network [35,36], etc. Among them, the authors of [37] proposed a deep generative network model for feature extraction of multivariate time series and introduced mutual information into the loss function to improve the expression capability and accuracy of the model. In the study [31], the authors proposed a variable selection method based on mutual information and the variance inflation factor (MI-VIF), which eliminated the variables based on MI and VIF, respectively, and showed good prediction performance. A feature selection algorithm based on MI and particle swarm optimization, which optimized the traditional particle update mechanism and swarm initialization strategy, was proposed in the work [23]. The study [38] proposed a novel fuzzy multi-information-based multi-label feature selection approach that was suitable for multi-label learning. The authors of [39] proposed a conditional multiple information-based feature selection algorithm for maximum reliability and minimum redundancy.

The above research mainly focuses on the improvement of the feature extraction algorithm based on mutual information. However, for the feature extraction analysis of time series, the main problem of the mutual information-based method is in finding the most appropriate target variables, that is, the key to the mutual information-based method is to find the target variables. In most cases, there is no intuitive or directly available target information. PCA is a multivariate statistical analysis technique for data compression and feature extraction, which can effectively remove linear correlations between data [40]. The purpose of PCA is to discard a small portion of information using linear transformation and replace the original variable with a few new comprehensive variables while ensuring minimal data loss. Therefore, it is required that the principal components fully reflect the information of the original variables, while PCA can eliminate the linear correlation between variables and suppress noise by fusing multiple variables [41,42]. Based on the above problems and the technical advantages of the PCA method, this paper uses the PCA method to reduce dimensions and fuse sensor data to obtain the target variables that can represent the system state.

Given the problem, this paper proposes a target information extraction method based on PCA dimension reduction, and on this basis, mutual information is used to distinguish and extract the suitable sensor signal. To validate the effectiveness of the method, analysis and validation are conducted on NASA's publicly available CMAPSS aircraft engine dataset, and the effectiveness of operating condition recognition based on the extracted features is tested. The specific content is as follows:

(1) We propose a time series feature selection method based on PCA dimension reduction and MI. MI is used to quantify the correlation between two variables, but for time series, it is difficult to obtain effective target variables. Therefore, PCA dimension reduction is used to extract the target variables from the time series, and feature selection of the time series is conducted based on this.
(2) We design a specific technical process for feature selection based on the above theoretical methods. In this process, we focus on the construction of target variables and the method of sensor selection and conduct experimental verification.
(3) The effectiveness of the proposed method is verified based on a publicly available aviation engine operation dataset. The experimental results are compared with

other methods to verify the feasibility of the proposed method, and experiments are conducted on condition recognition based on the selected features.

The rest of this paper is organized as follows: In Section 2, we describe the basic principle of the algorithm in detail and propose a feature extraction method flow based on MI. In Section 3, we analyze the experimental dataset and conduct comparative experiments using the algorithms, and in Section 4, we analyze the obtained result and compare the proposed idea with existing approaches. Section 5 is the conclusion of this paper, which includes recommendations for future work.

2. Methodology

2.1. Problem Description

This paper focuses on the feature selection of multivariate time series. Each sample in the dataset is a set of time series, and the length of each time series may vary. A set of univariate time series can generally be defined as follows:

$$T = t_1, t_2, \cdots, t_n \tag{1}$$

where n represents the length of the time series. Correspondingly, an m-dimensional multi-variable time series can be written as:

$$\begin{aligned} &T_1 = t_{11}, t_{12}, \cdots, t_{1n_1} \\ &T_2 = t_{21}, t_{22}, \cdots, t_{2n_2} \\ &\cdots \\ &T_m = t_{m1}, t_{m2}, \cdots, t_{mn_m} \end{aligned} \tag{2}$$

In the feature selection of time series, the dimension of the monitoring data is the number of features in the time series. When processing time series data, if the number of features in the dataset is too large, it can seriously affect the effectiveness of model training, leading to the so-called “curse of dimension” [43]. Effective feature selection can lay a good foundation for subsequent anomaly detection, fault diagnosis, condition identification, remaining useful life prediction, etc.

2.2. Feature Selection Based on Mutual Information

Mutual information is a measure of the interdependence between two variables, which mainly represents the correlation between them. Assuming that the joint probability distribution of two random variables X and Y is $p(x, y)$, and their marginal probability distribution is $p(x)$ and $p(y)$, respectively, then the mutual information $MI(X, Y)$ is the Kullback–Leibler divergence of the joint probability distribution $p(x, y)$ and the marginal probability distribution $p(x)$ and $p(y)$, which is defined as follows [28]:

$$MI(X, Y) = -\int_x \int_y f_{x,y}(x, y) \log \frac{f_{x,y}(x, y)}{f_x(x) f_y(y)} dxdy \tag{3}$$

From the above definition, when variables are completely independent or mutually independent, their mutual information is the smallest, and the result is 0. The greater the mutual information between variables, the stronger the correlation. In data processing, features with large mutual information with the target variable should be selected as much as possible to improve the prediction ability of the algorithm, and features with small mutual information should be eliminated to reduce data redundancy, that is, so-called feature selection based on maximum correlation and minimum redundancy [44].

In the specific application of mutual information, the probability density estimation function can be used to approximate the edge probability distribution to simplify the above formula and express it as follow:

$$MI(X,Y) = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\hat{f}_{x,y}(x_i,y_i)}{\hat{f}_x(x_i)\hat{f}_y(y_i)} \tag{4}$$

where $\hat{f}_x(x_i)$ and $\hat{f}_y(y_i)$ are the probability density estimations of X and Y, $\hat{f}_{x,y}(x_i,y_i)$ is the joint probability density estimation function, and N is the number of data samples.

According to Equation (4), the key to mutual information is the estimation of the probability density function. For the univariate edge probability distribution, the kernel probability density estimation method can be used for its estimation. In this way, without prior knowledge of the data distribution, the characteristics of the data distribution can be learned based on the data samples themselves. This is a non-parametric method for estimating probability density functions. For example, for variable X, suppose it has n sample points $x_1, x_2, \cdots, x_n$, which are independent and identically distributed random variables. Then, its kernel probability density estimation can be expressed as:

$$\hat{f}_x(x) = \frac{1}{n}\sum_{i=1}^{n}K\left(\frac{x-x_i}{h}\right) \tag{5}$$

where $K(\,)$ is the kernel function and h is the scale parameter of the kernel function, which defines the similarity between samples. Usually, Gaussian kernel functions are used, which can be expressed as:

$$K\left(\frac{x-x_i}{h}\right) = \frac{1}{\sqrt{2\pi}h}\exp\left(-\frac{(x-x_i)^2}{2h^2}\right) \tag{6}$$

Therefore, the probability density function of sample points can be estimated as:

$$\hat{f}_x(x) = \frac{1}{Nh_w}\sum_{i=1}^{N}\frac{1}{\sqrt{2\pi}\sigma_x}\exp\left(-\frac{(x-x_j)^2}{2\sigma_x^2 h^2}\right) \tag{7}$$

where σ_x is the variance in variable X, and the scale parameter h can be determined according to the following equation:

$$h_w = \left(\frac{4}{d+2}\right)^{1/(d+4)} N^{-1(d+4)} \tag{8}$$

where d is the dimension of variable X, and for one-dimensional random variables, d is equal to 1.

For the joint probability distribution $\hat{f}_{x,y}(x_i,y_i)$, assuming $z = [x\,y]^{\mathrm{T}}$, the estimation of the joint probability distribution can be expressed as:

$$\hat{f}_z(z) = \frac{1}{Nh}\sum_{i=1}^{N}K(u') \tag{9}$$

where:

$$u' = \frac{(z-z_i)^T C_z^{-1}(z-z_i)}{h^2} \tag{10}$$

$$K(u') = \frac{1}{(2\pi)^{d/2}\det(C_z)^{1/2}}\exp(-u'/2) \tag{11}$$

where C_z is the covariance matrix of z and $\det(C_z)$ is the value of the determinant of C_z.

2.3. Construction of the Time Series Target Variable Based on PCA

As mentioned before, for the feature selection problem of time series based on mutual information, one of the difficulties is finding appropriate target variables. In response to this issue, this paper proposes a time series target variable construction method based on PCA. By performing dimensionality reduction processing on multidimensional time series variables using PCA, variables that can characterize changes in system lifetime performance, which can be referred to as system operability, are obtained. On this basis, mutual information analysis is carried out between it and the monitoring variables of the system to conduct feature selection, so as to ensure that the selected features can maximize the prediction accuracy.

Assuming that the dataset $X = (x_1, x_2, \ldots, x_L)$ contains L samples and x is an N-dimensional variable, then the empirical mean of x can be expressed as:

$$u = (\mu_1, \mu_2, \ldots, \mu_N)^T \tag{12}$$

$$\mu_n = \frac{1}{L}\sum_{l=1}^{L} X[n, l] \tag{13}$$

The covariance matrix of the samples can be expressed as:

$$C = \frac{1}{L-1}(X - uh)\cdot(X - uh)^T \tag{14}$$

where h is an N-dimensional vector that is all 1. The covariance matrix C is decomposed into eigenvectors, and the first M eigenvectors are selected. The selection of the number of eigenvectors M depends on the desired data variance. In this paper, in order to obtain a one-dimensional objective variable that can characterize the system operability, M = 1 is chosen.

2.4. Workflow of the Proposed Feature Selection Method for Time Series

Figure 1 shows the proposed time series feature selection workflow. After obtaining the system time series monitoring data, the sensor signals are initially screened and then normalized. Afterward, in order to obtain the system target variables, PCA dimensionality reduction is performed on the normalized data to obtain one-dimensional time variables that can characterize the trend in the system operability. Then, the mutual information between each time series features with the target variable, as well as between each time series feature, is calculated. According to the value of mutual information, the time series features with large correlations and small redundancy with the target variables are selected.

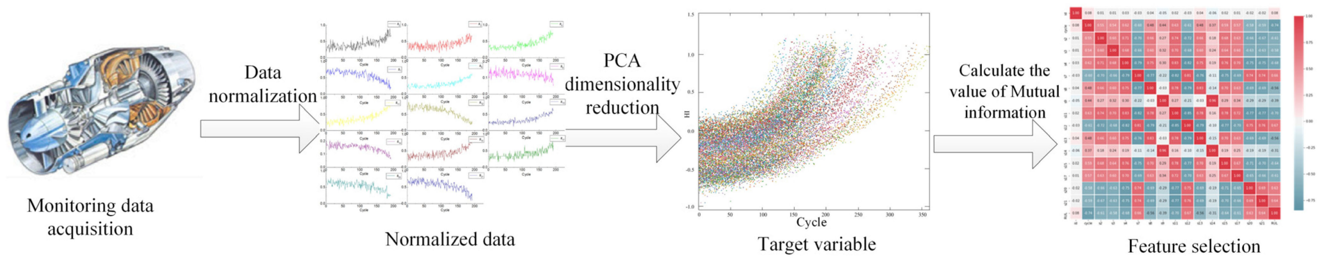


Figure 1. Workflow of the proposed feature selection method.

3. Experiments

To verify the effectiveness of the method proposed in this paper, the CMAPSS aircraft engine simulation time series dataset published by NASA [45] is used to validate the proposed method, which is then compared with different methods. All simulation experiments are carried out in the Pycharm 2021 (Community Edition) environment on a PC with an AMD Ryzen 7 3700X eight-core CPU and 16 GB memory.

3.1. C-MAPPS Datasets

As shown in Figure 2, the structure diagram of the aero-engine based on CMAPSS contains several components such as a fan, combustion chamber, high-/low-pressure booster, turbine, nozzle, etc.

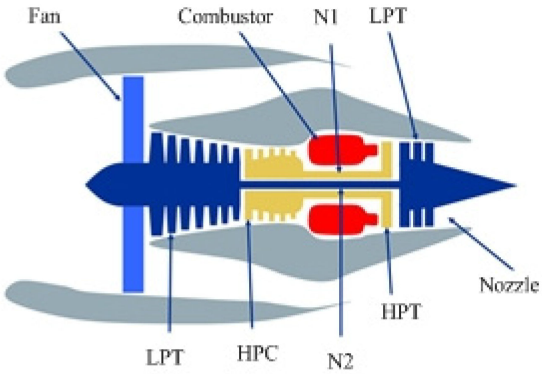


Figure 2. Structure diagram of the C-MAPPS aero-propulsion system.

Table 1 shows the relevant parameters of the C-MAPSS dataset and the monitoring data for each work cycle. Gaussian white noise is added to the monitoring data to simulate the actual sensor noise. This dataset can realistically simulate engine systems with high reliability.

Table 1. Engine sensor data description [45].

No.	Symbol	Description	Unit
1	T2	Total temperature at the fan inlet	°R
2	T24	Total temperature at the low-pressure booster outlet	°R
3	T30	Total temperature at the high-pressure booster outlet	°R °R
4	T50	Total temperature at the low-pressure turbine outlet	°R
5	P2	Pressure at the fan inlet	psia
6	P15	Total pressure of the external bypass	psia
7	P30	Total pressure at the high-pressure booster outlet	psia
8	Nf	Physical speed of the fan	rpm
9	Nc	Physical speed of the core engine	rpm
10	epr	Engine pressure ratio (P50/P2)	—
11	Ps30	Static pressure at the high-pressure booster outlet	psia
12	Phi	Ratio of fuel flow and P30	pps/psi
13	NRf	Corrected speed of the fan	ppm
14	NRc	Corrected speed of the core engine	rpm
15	BPR	Bypass ratio	—
16	farB	Gas–air ratio of the combustion chamber	—
17	htBleed	Bleed air entropy	—
18	Nf_dmd	Set speed of the fan	rpm
19	PCNfR_dmd	Set corrected speed of the core engine	rpm
20	W31	Cooling bleed air flow of the high-pressure turbine	lbm/s
21	W32	Cooling bleed air flow of the low-pressure turbine	lbm/s

In the CMAPSS dataset, some sensor signals are constant and do not vary with engine operation. After removing these data, the CMAPSS dataset has 14 sensor signals in total, that is, the data dimension of the original data space is reduced from 21 dimensions to 14 dimensions. Further, in order to reduce the redundancy between variables, based on the idea of the feature selection of maximum correlation and minimum redundancy, before feature selection, the variables with high redundancy are removed according to the mutual information value between the variables. Figure 3 shows the mutual information thermogram of 14 sensor signals in the C-MAPSS dataset. The diagonal value of the thermogram is 1, and it is symmetrical with the diagonal.

Figure 3. Mutual information matrix among variables (figure drawn by seaborn [46]).

From Figure 3, it can be seen that some sensor signals are highly correlated. For example, the value of mutual information between s9 and s14 is 1, indicating that there is a high degree of redundancy between them. From Table 1, it can be seen that s9 represents the physical speed of the core engine, and s14 represents the corrected speed of the core engine. The measured values of the two are highly correlated, and the calculated mutual information is consistent in a physical sense. Based on the mutual information matrix, the system features are further filtered. With 0.95 as the threshold (manually set), some redundant features are deleted, and a total of nine sensor signals in the system can be obtained, that is, from 21 dimensions of the original data space to 9 dimensions after mutual information filtering among variables.

3.2. Data Normalization

Due to the different physical quantities measured by different sensors, the data need to be normalized to make the variable range consistent, so as to achieve better classification and prediction effects. This paper uses Z-score standardization to normalize the input data, which can be expressed as:

$$X' = \frac{X - \overline{X}}{\sigma} \tag{15}$$

where X is the time series data, $\overline{X}$ and σ are the average and variance of X, respectively, and X' is the normalized monitoring data. Figure 4 shows the trend in nine variables after normalization.

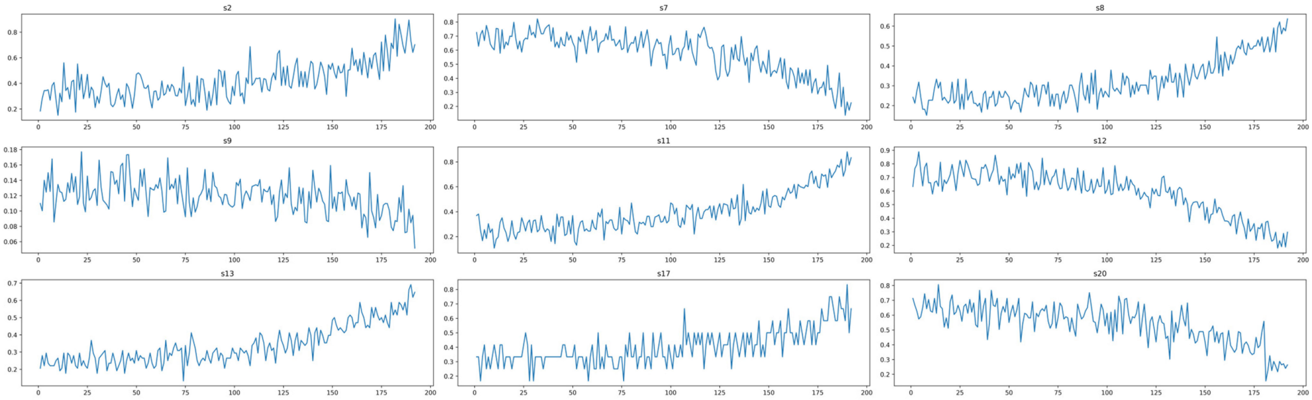


Figure 4. The trends in the 9 variables after normalization (figure drawn by matplotlib [47]).

3.3. Target Variable Construction

Two issues need to be considered in the construction of the target variable of the system. One is that the variable needs to be able to reflect the change in system operability, and the other is that the variable needs to ensure monotonicity and irreversibility in the whole life cycle. For monitoring and control systems, in most cases, the monotonicity, predictability, and irreversibility of a single sensor signal cannot be guaranteed, nor can it fully reflect the system operability. Therefore, this paper uses the sensor information fusion method based on PCA dimension reduction and kernel smoothing to extract the system target variables, thus ensuring the monotonicity and irreversibility of the degradation trajectory [48].

It should be noted that the degradation trajectory extracted using PCA and KR is the system operability mentioned above, which reflects the overall operational status of the system. The specific approach is to first fuse the multidimensional monitoring variables of the system using PCA to extract the principal component that contains the most system information. At the same time, in order to ensure the monotonicity and irreversibility of the system operability, KR is used to smooth the extracted principal components, so as to obtain the curve of system operability, as shown in Figure 5.

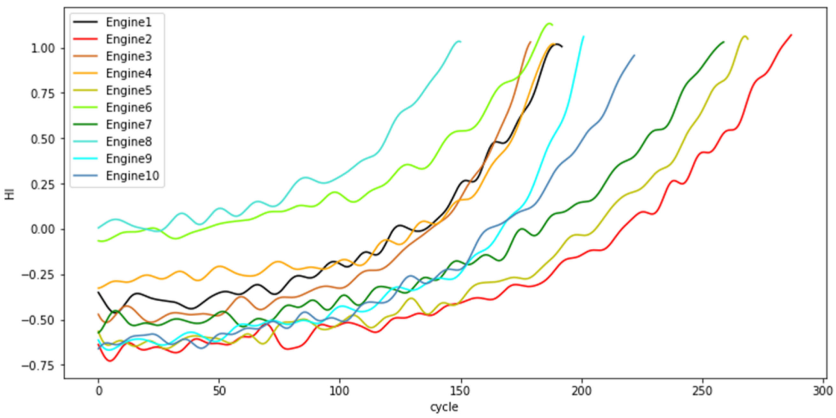


Figure 5. Target variables for engines 1 to 10 based on PCA dimensionality reduction (figure drawn by matplotlib).

Figure 5 shows the system operability of engines from numbers 1 to 10 in the dataset. It can be seen from the figure that by extracting the system degradation curve as the target variable, the monotonicity and irreversibility of the variable can be guaranteed, and the system operability can reflect the health status of the system to a certain extent.

3.4. Calculation of the Mutual Information of Sensor Signals

The first engine data in the CMAPSS dataset is taken as an example to analyze the mutual information values of different sensor signals with the target variable. It should be noted that among the 21 sensor signals in Table 1, some sensor signals are constant values, which do not vary with engine operation, such as s1, s5, s6, s10, s16, s18, and s19. After removing these data, a total of 14 sensor signals are obtained in the C-MAPSS dataset. Table 2 shows the mutual information values of the nine different sensor signals with the target variable, and Figure 6 shows the histogram of the mutual information values of sensor signals with the target variable.

Table 2. Value of mutual information between the sensor signal and the target variable.

Sensor Number	Value of Mutual Information
2	0.570
7	0.697
8	0.597
9	1.000
11	0.951
12	0.705
13	0.682
17	0.485
20	0.506

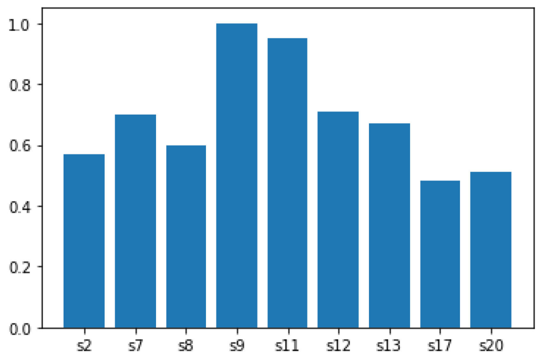


Figure 6. Mutual information histogram of sensor signals with the target variable (figure drawn by matplotlib).

It can be seen from Table 1 and Figure 6 that the five sensor signals with high mutual information with the target variable are s7, s9, s11, s12, and s13, corresponding to the total pressure at the high-pressure booster outlet, the physical speed of the core engine, the static pressure at the high-pressure booster outlet, the ratio of fuel flow and P30, and the corrected speed of the fan.

4. Result Analysis

Using the calculation in Section 3, we obtained the values of MI for each sensor signal based on the target variable. To verify the effectiveness of the proposed method, we compared and analyzed it with the scores obtained from the F-test score to clarify that the proposed idea is better than the existing approach. On the other hand, in order to measure the performance of the extracted features, we use the extracted features to identify six different operating conditions of the engine in the CMAPSS dataset and verify them with visual data clusters.

4.1. Comparative Analysis of the Selected Features

In order to verify the effect of the selected features, the correlation between the nine sensor signals with the target variable is visualized, and the F-test scores between sensor signals and target variables are given, as shown in Figure 7. The F-test score [49] is used to extract the linear relationship between the nine sensor signals with the target variable, which can be mutually verified with feature selection based on mutual information to a certain extent.

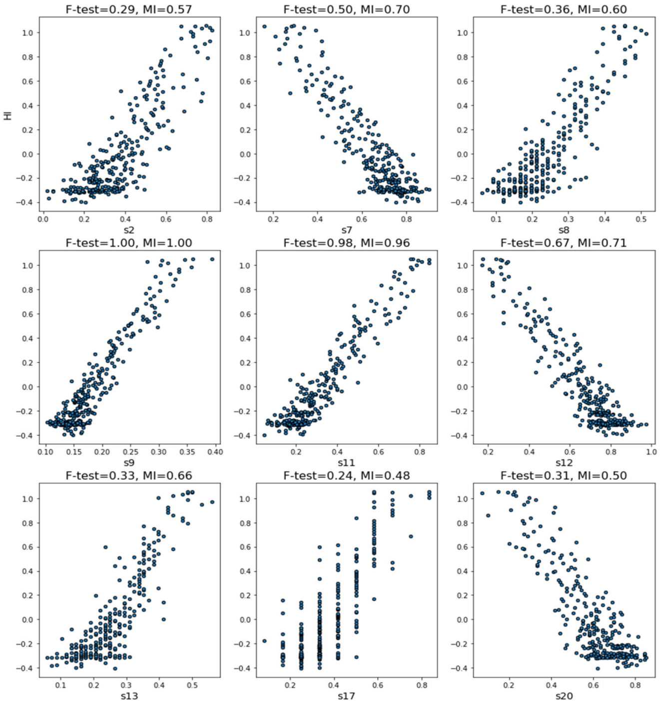


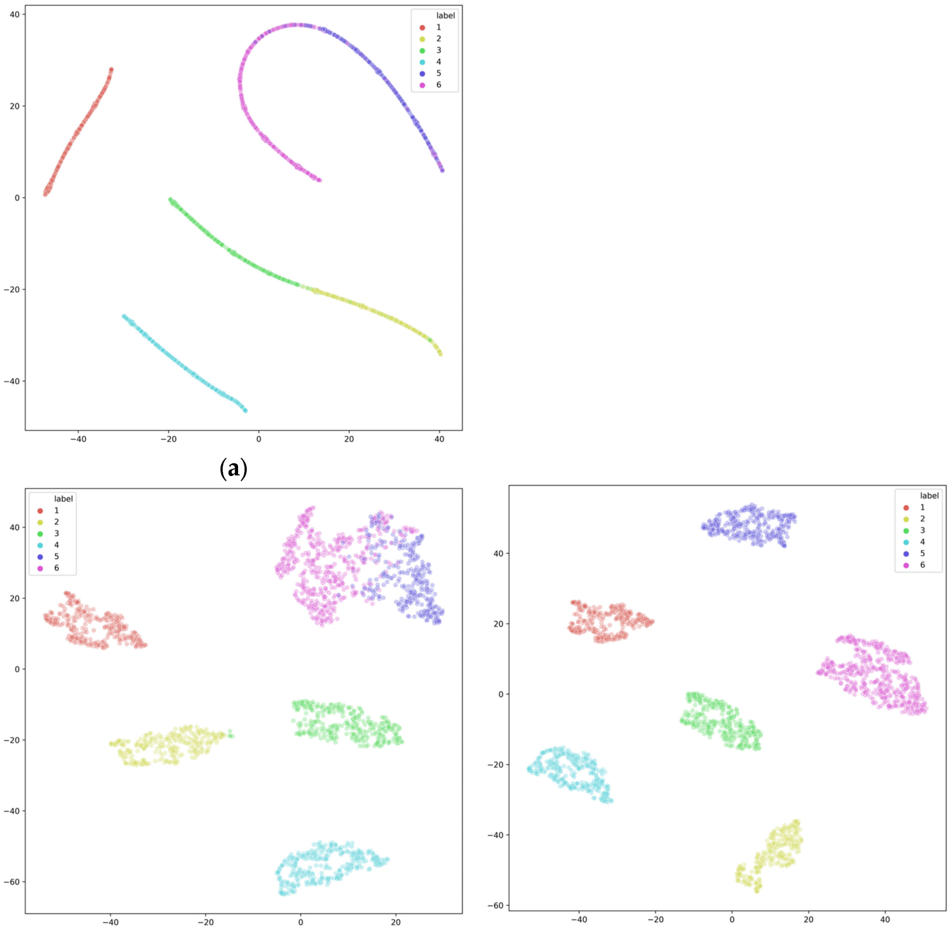
Figure 7. F-test and mutual information comparison of the nine sensor signals (figure drawn by matplotlib).

As can be seen from Figure 6, the features selected based on the F-test are basically consistent with those selected based on mutual information. For example, for sensor signal s9, both give scores of 1 and for sensor signal s11, the scores given by both are 0.98 and 0.96, respectively. Obviously, this indicates a strong linear relationship between s9 and s11 with the target variable. However, the difference is that feature extraction based on mutual information can better analyze and extract the nonlinear relationship between features and target variables. For example, when s13, s17, and s20 do not show an obvious linear relationship with the target variables, the value of mutual information is higher than the F-test score. Based on the above analysis, the five time series characteristics that can best represent the state of the system are obtained, and their value of mutual information is s9, s11, s12, s7 and s13 from high to low.

4.2. Effect Analysis of Working Condition Recognition Based on the Selected Features

Figure 7 shows the effect of working condition recognition based on the selected system time series features. In the CMAPSS dataset, there are a total of six different operating conditions for the engines, and they continuously alternate between the six different operating conditions throughout their entire lifetime. Accurately identifying the system working conditions based on the selected sensor signals is an important foundation for subsequent research such as system operability analysis, etc.

Figure 8a shows the effect of recognition based solely on sensor signal s9, Figure 8b shows the effect of working condition recognition based on s9 and s11, and Figure 8c shows working condition recognition based on s9, s11, and s12. From Figure 8, it can be seen that Figure 8a,b can achieve working condition recognition to a certain extent. However, there are some data points that overlap, which means that the system's working condition identification cannot be effectively and completely achieved. By selecting the first three features, the recognition of complex working conditions of the system can be effectively realized, which shows the effectiveness of selecting time series features of the system based on mutual information.



(**a**) (**b**) (**c**)

Figure 8. Recognition of working conditions based on the selected features (figure drawn by seaborn): (**a**) working condition recognition based on s9; (**b**) working condition recognition based on s9 and s11; and (**c**) working condition recognition based on s9, s11, and s12.

5. Conclusions

Extracting effective features of a system based on mutual information values is a commonly used feature extraction method in the field of machine learning. However, one of the most challenging issues with this method is finding suitable target variables. Theoretically, these target variables must include the main performance indicators of system operation to ensure that the extracted features can effectively reflect the multiple characteristics of the system. Especially for time series, the historical data of system operation contains a large amount of information, but at the same time, there is significant redundancy and noise within this information. To solve this problem, this paper proposes a method for feature selection based on MI, in which system operability is used as a target variable. Firstly, the system operability is extracted based on PCA dimension reduction and kernel smoothing, which is monotonic and irreversible. Using it as the target variable for MI analysis is more credible compared to using individual sensor signals as the target variable. On this basis, features are filtered based on the value of mutual information between features and the target variable. In order to verify the effectiveness of the method, comparisons and analyses were conducted with F-test scores. The results show that feature extraction based on mutual information can better analyze and extract the non-linear relationship between features and target variables. Finally, to further validate the effectiveness of the method, we conducted tests on condition recognition based on the CMPASS dataset. The CMPASS dataset contains continuous operational data of the engine throughout its entire life cycle under six different operating conditions. Effectively partitioning these data between the six different operating conditions is crucial for subsequent analysis. Therefore, based on the selected feature variables, the system condition recognition was carried out. The results indicate that by using the first three extracted feature variables, the complex condition recognition of the system can be achieved. From the above analysis, we identified that the proposed method can effectively achieve system feature extraction and is suitable for feature extraction problems of time series data.

However, it should be noted that one of the most critical steps of the method proposed in this paper is constructing the system operability based on sufficient historical operational data. Therefore, the method is mainly applicable to devices with a large amount of historical operating data. If the data are insufficient, the constructed system operability may not reflect the overall characteristics of the system, leading to inaccurate estimation of MI values. For example, the CMAPSS dataset FD001 used in this paper contains monitoring parameter information for the full life cycle status of 200 engines.

Regarding future work, we plan to study the construction of system operability based on online data and the analysis of MI values, in order to reduce the dependence on historical operating data and improve the applicability of this method.

Author Contributions: Conceptualization, L.H.; methodology, X.Z.; software, L.H.; validation, L.S.; formal analysis, X.Z.; investigation, L.G.; resources, L.G.; data curation, L.S.; writing—original draft preparation, L.H.; writing—review and editing, L.G.; supervision, L.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in the 'Turbofan Engine Degradation Simulation Data Set' at https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/ (accessed on 17 February 2024).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Aremu, O.O.; Hyland-Wood, D.; McAree, P.R. A machine learning approach to circumventing the curse of dimensionality in discontinuous time series machine data. *Reliab. Eng. Syst. Saf.* **2020**, *195*, 106706. [CrossRef]
2. Khaire, U.M.; Dhanalakshmi, R. Stability of feature selection algorithm: A review. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 1060–1073. [CrossRef]
3. Zebari, R.; Abdulazeez, A.; Zeebaree, D.; Zebari, D.; Saeed, J. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *J. Appl. Sci. Technol. Trends* **2020**, *1*, 56–70. [CrossRef]
4. Jiao, W.; Cheng, X.; Hu, Y.; Hao, Q.; Bi, H. Image Recognition Based on Compressive Imaging and Optimal Feature Selection. *IEEE Photonics J.* **2022**, *14*, 1–12. [CrossRef]
5. Afza, F.; Khan, M.A.; Sharif, M.; Kadry, S.; Manogaran, G.; Saba, T.; Ashraf, I.; Damaševičius, R. A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection. *Image Vis. Comput.* **2021**, *106*, 104090. [CrossRef]
6. Zhao, H.; Liu, Z.; Yao, X.; Yang, Q. A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach. *Inf. Process. Manag.* **2021**, *58*, 102656. [CrossRef]
7. Sharma, M.; Kaur, P. A Comprehensive Analysis of Nature-Inspired Meta-Heuristic Techniques for Feature Selection Problem. *Arch. Comput. Methods Eng.* **2021**, *28*, 1103–1127. [CrossRef]
8. Abualigah, L.; Dulaimi, A.J. A novel feature selection method for data mining tasks using hybrid Sine Cosine Algorithm and Genetic Algorithm. *Clust. Comput.* **2021**, *24*, 2161–2176. [CrossRef]
9. Wang, L.; Jiang, S.; Jiang, S. A feature selection method via analysis of relevance, redundancy, and interaction. *Expert Syst. Appl.* **2021**, *183*, 115365. [CrossRef]

10. Zheng, J.; Pan, H.; Tong, J.; Liu, Q. Generalized refined composite multiscale fuzzy entropy and multi-cluster feature selection based intelligent fault diagnosis of rolling bearing. *ISA Trans.* **2022**, *123*, 136–151. [CrossRef]
11. Cao, Y.; Sun, Y.; Xie, G.; Li, P. A Sound-Based Fault Diagnosis Method for Railway Point Machines Based on Two-Stage Feature Selection Strategy and Ensemble Classifier. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 12074–12083. [CrossRef]
12. Buchaiah, S.; Shakya, P. Bearing fault diagnosis and prognosis using data fusion based feature extraction and feature selection. *Measurement* **2022**, *188*, 110506. [CrossRef]
13. Wang, Y.; Zhao, Y. Three-stage feature selection approach for deep learning-based RUL prediction methods. *Qual. Reliab. Eng. Int.* **2023**, *39*, 1223–1247. [CrossRef]
14. Abushark, Y.B. An intelligent feature selection approach with systolic tree structures for efficient association rules in big data environment. *Comput. Electr. Eng.* **2022**, *101*, 108080. [CrossRef]
15. Dhindsa, A.; Bhatia, S.; Agrawal, S.; Sohi, B.S. An Improvised Machine Learning Model Based on Mutual Information Feature Selection Approach for Microbes Classification. *Entropy* **2021**, *23*, 257. [CrossRef] [PubMed]
16. Halomoan, J.; Ramli, K.; Sudiana, D.; Gunawan, T.S.; Salman, M. ECG-Based Driving Fatigue Detection Using Heart Rate Variability Analysis with Mutual Information. *Information* **2023**, *14*, 539. [CrossRef]
17. Islam, M.R.; Ahmed, B.; Hossain, M.A.; Uddin, M.P. Mutual Information-Driven Feature Reduction for Hyperspectral Image Classification. *Sensors* **2023**, *23*, 657. [CrossRef] [PubMed]
18. Thakkar, A.; Lohiya, R. A survey on intrusion detection system: Feature selection, model, performance measures, application perspective, challenges, and future research directions. *Artif. Intell. Rev.* **2022**, *55*, 453–563. [CrossRef]
19. Halim, Z.; Yousaf, M.N.; Waqas, M.; Sulaiman, M.; Abbas, G.; Hussain, M.; Ahmad, I.; Hanif, M. An effective genetic algorithm-based feature selection method for intrusion detection systems. *Comput. Secur.* **2021**, *110*, 102448. [CrossRef]
20. Nimbalkar, P.; Kshirsagar, D. Feature selection for intrusion detection system in Internet-of-Things (IoT). *ICT Express* **2021**, *7*, 177–181. [CrossRef]
21. Alalhareth, M.; Hong, S. An Improved Mutual Information Feature Selection Technique for Intrusion Detection Systems in the Internet of Medical Things. *Sensors* **2023**, *23*, 4971. [CrossRef]
22. Maldonado, J.; Riff, M.C.; Neveu, B. A review of recent approaches on wrapper feature selection for intrusion detection. *Expert Syst. Appl.* **2022**, *198*, 116822. [CrossRef]
23. Song, X.; Zhang, Y.; Gong, D.; Sun, X. Feature selection using bare-bones particle swarm optimization with mutual information. *Pattern Recognit.* **2021**, *112*, 107804. [CrossRef]
24. Sánchez-Maroño, N.; Alonso-Betanzos, A.; Tombilla-Sanromán, M. Filter Methods for Feature Selection—A Comparative Study. In *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning, Birmingham, UK, 16–19 December 2007*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 178–187.
25. Di Mauro, M.; Galatro, G.; Fortino, G.; Liotta, A. Supervised feature selection techniques in network intrusion detection: A critical review. *Eng. Appl. Artif. Intell.* **2021**, *101*, 104216. [CrossRef]
26. Cai, J.; Luo, J.; Wang, S.; Yang, S. Feature selection in machine learning: A new perspective. *Neurocomputing* **2018**, *300*, 70–79. [CrossRef]
27. González-Vidal, A.; Jiménez, F.; Gómez-Skarmeta, A.F. A methodology for energy multivariate time series forecasting in smart buildings based on feature selection. *Energy Build.* **2019**, *196*, 71–82. [CrossRef]
28. Kraskov, A.; Stögbauer, H.; Grassberger, P. Estimating mutual information. *Phys. Rev. E* **2004**, *69*, 066138. [CrossRef] [PubMed]
29. Pascoal, C.; Oliveira, M.R.; Pacheco, A.; Valadas, R. Theoretical evaluation of feature selection methods based on mutual information. *Neurocomputing* **2017**, *226*, 168–181. [CrossRef]
30. Zhou, H.; Wang, X.; Zhu, R. Feature selection based on mutual information with correlation coefficient. *Appl. Intell.* **2022**, *52*, 5457–5474. [CrossRef]
31. Cheng, J.; Sun, J.; Yao, K.; Xu, M.; Cao, Y. A variable selection method based on mutual information and variance inflation factor. *Spectrochim. Acta Part A Mol. Biomol. Spectrosc.* **2022**, *268*, 120652. [CrossRef] [PubMed]
32. Sun, L.; Wang, T.; Ding, W.; Xu, J.; Lin, Y. Feature selection using Fisher score and multilabel neighborhood rough sets for multilabel classification. *Inf. Sci.* **2021**, *578*, 887–912. [CrossRef]
33. Sadeghian, Z.; Akbari, E.; Nematzadeh, H. A hybrid feature selection method based on information theory and binary butterfly optimization algorithm. *Eng. Appl. Artif. Intell.* **2021**, *97*, 104079. [CrossRef]
34. Liang, J.; Hou, L.; Luan, Z.; Huang, W. Feature Selection with Conditional Mutual Information Considering Feature Interaction. *Symmetry* **2019**, *11*, 858. [CrossRef]
35. Hammad, M.; Chelloug, S.A.; Alayed, W.; El-Latif, A.A.A. Optimizing Multimodal Scene Recognition through Mutual Information-Based Feature Selection in Deep Learning Models. *Appl. Sci.* **2023**, *13*, 11829. [CrossRef]
36. Li, K.; Fard, N. A Novel Nonparametric Feature Selection Approach Based on Mutual Information Transfer Network. *Entropy* **2022**, *24*, 1255. [CrossRef]
37. Li, J.; Ren, W.; Han, M. Mutual Information Variational Autoencoders and Its Application to Feature Extraction of Multivariate Time Series. *Int. J. Pattern Recognit. Artif. Intell.* **2022**, *36*, 2255005. [CrossRef]
38. Liu, J.; Lin, Y.; Ding, W.; Zhang, H.; Du, J. Fuzzy Mutual Information-Based Multilabel Feature Selection with Label Dependency and Streaming Labels. *IEEE Trans. Fuzzy Syst.* **2023**, *31*, 77–91. [CrossRef]

39. Gu, X.; Guo, J.; Xiao, L.; Li, C. Conditional mutual information-based feature selection algorithm for maximal relevance minimal redundancy. *Appl. Intell.* **2022**, *52*, 1436–1447. [CrossRef]
40. Ringnér, M. What is principal component analysis? *Nat. Biotechnol.* **2008**, *26*, 303–304. [CrossRef] [PubMed]
41. Hasan, B.M.S.; Abdulazeez, A.M. A Review of Principal Component Analysis Algorithm for Dimensionality Reduction. *J. Soft Comput. Data Min.* **2021**, *2*, 20–30.
42. Schreiber, J.B. Issues and recommendations for exploratory factor analysis and principal component analysis. *Res. Soc. Adm. Pharm.* **2021**, *17*, 1004–1011. [CrossRef] [PubMed]
43. Verleysen, M.; François, D. The Curse of Dimensionality in Data Mining and Time Series Prediction. In *Proceedings of the International Work-Conference on Artificial Neural Networks, Barcelona, Spain, 8–10 June 2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 758–770.
44. Peng, H.; Long, F.; Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1226–1238. [CrossRef] [PubMed]
45. Saxena, A.; Goebel, K.; Simon, D.; Eklund, N. Damage propagation modeling for aircraft engine run-to-failure simulation. In Proceedings of the 2008 International Conference on Prognostics and Health Management, Denver, CO, USA, 6–9 October 2008; pp. 1–9.
46. Seaborn: Statistical Data Visualization. Available online: https://seaborn.pydata.org/ (accessed on 17 February 2024).
47. Matplotlib: Visualization with Python. Available online: https://matplotlib.org/ (accessed on 17 February 2024).
48. Huang, L.; Gong, L.; Chen, Y.; Li, D.; Zhu, G.; Ma, J. Trajectory Similarity Matching and Remaining Useful Life Prediction Based on Dynamic Time Warping. *Math. Probl. Eng.* **2022**, *2022*, 5344461. [CrossRef]
49. Hui, F.; Müller, S.; Welsh, A.H. Testing random effects in linear mixed models: Another look at the F-test (with discussion). *Aust. New Zealand J. Stat.* **2019**, *61*, 61–84. [CrossRef]

applied sciences

MDPI

Article

Machinery Fault Signal Detection with Deep One-Class Classification

Dosik Yoon and Jaehong Yu *

Department of Industrial and Management Engineering, Incheon National University, Incheon 22012, Republic of Korea; 202221075@inu.ac.kr
* Correspondence: jhyu@inu.ac.kr; Tel.: +82-32-835-8485

Abstract: Fault detection of machinery systems is a fundamental prerequisite to implementing condition-based maintenance, which is the most eminent manufacturing equipment system management strategy. To build the fault detection model, one-class classification algorithms have been used, which construct the decision boundary only using normal class. For more accurate one-class classification, signal data have been used recently because the signal data directly reflect the condition of the machinery system. To analyze the machinery condition effectively with the signal data, features of signals should be extracted, and then, the one-class classifier is constructed with the features. However, features separately extracted from one-class classification might not be optimized for the fault detection tasks, and thus, it leads to unsatisfactory performance. To address this problem, deep one-class classification methods can be used because the neural network structures can generate the features specialized to fault detection tasks through the end-to-end learning manner. In this study, we conducted a comprehensive experimental study with various fault signal datasets. The experimental results demonstrated that the deep support vector data description model, which is one of the most prominent deep one-class classification methods, outperforms its competitors and traditional methods.

Keywords: condition-based maintenance; deep one-class classification; deep support vector data description; fault signal detection; time series signal

1. Introduction

Modern industrial fields have furnished more complex and sophisticated machinery systems. However, unexpected faults in these machinery systems bring about significant losses in productivity and efficiency. To avoid the abrupt faults of machinery systems, preventive maintenance (PM) has been widely adopted as a maintenance strategy in various industrial fields [1,2]. These PM strategies attempt to conduct the maintenance (e.g., overhaul, refurbishment, and repair tasks) before the faults occur. In general, PM strategies can be divided into time-based maintenance (TBM) and condition-based maintenance (CBM) [3,4]. The TBM strategy periodically executes the maintenance activities in a predefined schedule. However, the TBM strategy often causes expensive maintenance costs because unnecessary maintenance activities are often performed [5]. Unlike, the CBM strategy performs maintenance tasks only when machinery fault symptoms are detected. Thus, it should entail a real-time diagnosis of the machinery status. Owing to its efficiency, CBM has received considerable attention in various industrial fields in recent years [5,6].

The CBM procedure comprises three main tasks: data acquisition, data processing, and fault diagnosis. (1) In the first step, sensor data that involve the health status of the machinery system are collected in real-time. (2) Then, to analyze the sensor data more easily, the collected raw data are preprocessed through feature engineering techniques. (3) Finally, using the preprocessed data, the machinery operating status is diagnosed, and maintenance is performed when the fault symptoms of the machinery systems are detected [1,7,8]. It is obvious that appropriately preprocessing the raw sensor data and accurately detecting the fault symptoms are fundamental prerequisites for a successful CBM strategy [9,10]. Therefore, the main purpose of this study is to propose a fault detection framework with deep neural network structures in order to implement a more efficient CBM strategy. The deep neural network structures are specialized to the feature engineering for given learning tasks [11,12]. Hence, the deep neural network-based approaches might show superior fault detection performance.

Most of the previous studies on machinery fault detection have used the sensor data obtained from both normal operating and fault statuses. However, the data from the fault status are not available in general because the fault rarely occurs in many real situations [13,14]. In addition, the operating status label information is not easy to obtain because it requires expensive analytical costs on machinery operating status. Therefore, in this study, we focus on the unsupervised deep neural network-based fault detection methods, which only use the sensor data obtained from normal operating status.

In recent years, various time-series signal data, including vibration, acoustic, and thermometric signals, have been widely used for fault detection [15,16]. These time-series signal data explicitly reflect machinery operating status, and it helps to more accurately detect the fault symptoms. Besides, owing to the rapid development of sensing and data storage techniques, a large amount of time-series signal data can be easily collected in real-time [8,17]. Therefore, utilizing these time-series signal data for fault detection has been spotlighted by a number of machinery operators. Then, a fault detection model can be constructed using one-class classification (OCC) methods. The OCC methods construct a decision boundary solely using time-series signal data obtained from only normal status (generally referred to as target signal), and the decision boundary finally determines whether a newly collected time-series signal was generated from normal status or not [18–20]. If the time-series signal is located outside the decision boundary, it is rejected as a fault signal, which is considered that the machinery fault might occur, and appropriate actions should be rapidly taken. Although the OCC methods have shown satisfactory performance in fault signal detection, they require an additional procedure that derives meaningful features from raw time-series signal data. This additional procedure is termed feature extraction.

The feature extraction attempts to generate several latent variables summarizing intrinsic properties of raw time-series signals. In most previous studies, manually created features (e.g., root mean square (RMS), kurtosis, and crest factor (CF)) have been widely utilized [21–23]. On the other hand, traditional feature learning techniques, including principal component analysis (PCA) [24] and kernel principal component analysis (KPCA) [25], have also been successfully used as feature extraction methods. The raw time-series signals are recorded at high sampling resolution and can be treated as high-dimensional data because individual values of the signal recorded at each time point can be regarded as variables. Thus, these traditional feature learning techniques entail dimensionality reduction to summarize the raw time-series signals of high dimensionality into smaller useful features. In spite of its simplicity, the manually crafted features simply summarize information of raw time-series signal data into a small number of values, and thus, it is difficult to reflect various characteristics of complex and noisy signals having nonstationary and nonlinear patterns [26,27]. For this reason, the manual feature creation may not be suitable for fault signal detection tasks. In addition, several traditional feature-learning techniques cannot accommodate the nonlinearity of target time-series signals, and kernel-based methods tend to be sensitive to the kernel function settings [28]. Finally, the feature creation or feature learning procedure is separately performed before fault detection model construction, and hence, the generated features may not be specialized for fault detection tasks [29].

To address these limitations, in recent years, deep neural network structures have been incorporated into OCC methods. In the deep neural network-based OCC methods (referred to as deep OCC methods), the feature extraction procedure is simultaneously performed with fault signal detection tasks in an end-to-end manner [11,12]. Thus, the deep neural network is basically designed to generate features optimized to a specified loss function of its corresponding task. Owing to the end-to-end manner's advantage, deep neural network structures have been successfully used in OCC methods in recent years. Among various OCC methods, the boundary-based OCC methods (e.g., support vector data description (SVDD [30]) and one-class support vector machine (OCSVM [31])) have been the most widely combined with the deep neural network because their objective function can be easily reformulated for a loss function of the deep neural network. In the boundary-based methods, the objective function is formulated to construct a compact hypersphere enclosing the target class, and the hyperspheres can be used as a decision boundary to discriminate the fault signals from target signals. By doing so, optimized features through the deep neural network structure help build more sophisticated decision boundaries by accommodating the inherent patterns of target signals.

The following are the main contributions of this study:

- The deep OCC methods can achieve superior fault detection performance, although raw time-series signals are directly used as input data. In general, for more effectively analyzing the time-series signals, signal processing techniques (e.g., short-time Fourier transform (STFT [32,33]) or wavelet transform [34]) are used to transform the raw signals. However, these signal-processing techniques require additional user-specified hyperparameters, which should be carefully determined. In contrast, the deep OCC methods do not need any signal processing techniques. This implies the efficiency of the deep OCC methods in handling the raw time-series signal data.
- In the deep OCC methods, more useful features for fault signal detection can be simultaneously extracted along with minimizing loss function on anomaly detection tasks. By doing so, the fault signal detection performance can be improved.
- Finally, we applied the deep OCC methods to the widely used benchmark fault signal datasets and the signal dataset collected from our own rolling element experimental platform. By doing so, the effectiveness and applicability of the deep neural network-based methods to real fault signal detection problems can be confirmed.

The remainder of this paper is organized as follows. Section 2 presents the related works on the proposed study. In Section 3, we present a deep neural network-based fault signal detection framework. The experimental settings, baseline of the comparison method, and results are reported in Section 4. The concluding remarks are provided in Section 5.

2. Related Works

This section deals with the details of existing OCC methods. Most of them are closely related to this study in that they can be used to fault signal detection problems. Later, we will consider most of the OCC methods introduced in this section for the various fault signal detection problems.

2.1. One-Class Classification Methods

Up to now, a number of OCC methods have been proposed, and they can be categorized according to the way to utilize the information on target signals, including density-based, ensemble-based, and boundary-based methods [19,20]. Among them, we focus on the boundary-based methods because they can generate more flexible decision boundaries through a nonlinear feature mapping function, and an objective function to build the optimal decision boundary can be explicitly formulated.

The SVDD is one of the most well-known boundary-based OCC methods. The SVDD attempts to find the most compact hypersphere to enclose as many target signal data as possible [30]. Then, if a signal falls outside of the hypersphere, it is rejected as a fault signal, and vice versa for the signal located inside of the hypersphere. However, if the hypersphere is constructed to involve all training target signals, the radius of the hypersphere might be too large, and too many fault signals are accepted as normal ones. To avoid the hypersphere being excessively large, the SVDD adopts slack variables, which allow a few training target signals to be located on the outer side of the hypersphere [30]. On the SVDD algorithm, the optimal hypersphere can be obtained as follows:

$$\begin{aligned} &\underset{R,a,\xi_i}{minmize}\ R^2 + C\sum_{i=1}^{n}\xi_i \\ &s.t.\ \|\phi(x_i) - a\|_{F_k}^2 \le\ R^2 + \xi_i,\ \ i = 1,\ldots,n \\ &\xi_i \ge\ 0, \qquad\qquad\qquad\quad i = 1,\ldots,n, \end{aligned} \tag{1}$$

where R^2 is the radius of the hypersphere, a is the center of the hypersphere, and ξ_i denotes the slack variable of the i-th training target signal. In addition, C is a regularization parameter controlling the trade-off between the hypersphere's volume and false positive error that rejects the target signal as fault signals, and ϕ is a nonlinear feature mapping function. The dual problem of Equation (1) can be formulated as follows:

$$\begin{aligned} &\underset{\alpha}{maximize}\sum_{i=1}^{n}\alpha_i\langle x_i, x_i\rangle - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\langle x_i, x_j\rangle \\ &s.t.\ 0\ \le \alpha_i \le C,\ \ i = 1,\ldots,n \\ &\textstyle\sum_{i=1}^{N}\alpha_i = 1, \qquad i = 1,\ldots,n, \end{aligned} \tag{2}$$

where α_i represents a Lagrange multiplier of the object x_i and $K\langle\ ,\ \rangle$ denotes the kernel function corresponding to its own feature mapping function ϕ. After solving the dual problem presented in Equation (2), the new time-series signal (x_{new}) is classified as a normal or fault signal by calculating the distance between the new signal and the hypersphere center, $\|\phi(x_{new}) - a\ \|^2$ [30]. If this distance is larger than R (i.e., the new signal falls outside the hypersphere), it is finally rejected as a fault signal.

Although the support vector-based methods are the most popular OCC methods, they are sensitive to the noisy patterns around the target class, and thus, when there are several noises in training data, the support vector-based methods cannot capture target class structures [19,20,35]. Moreover, they are still quite vulnerable to the model settings, including the kernel function choice or regularization hyperparameters [36,37].

In addition to boundary-based methods, density-based and ensemble-based methods have also been widely used for fault signal detection. The density-based methods estimate a density that quantifies the representativeness of the target signal, and a signal having a small density value is rejected as a fault signal. Generally, the density can be defined with a probability function, and thus, the density-based methods involve probability function estimation procedures. The kernel-density estimation (KDE [38]) is widely used to estimate the probability function along with target class structures, and this method defines the probability function as a weighted sum of the kernel function values centered at each training target signal. The contribution of each signal to the probability function is denoted by its kernel function value. This kernel function is computed by dividing the distance between signals with the bandwidth. In the KDE method, the bandwidth determines the shape of the kernel function and affects the smoothness of the estimated probability function. For small bandwidth, the kernel function has a high peak and narrow width, and the estimated probability function leads to a complicated decision boundary. Conversely, if the bandwidth value is too large, the kernel function may have a broad width, and the estimated probability function tends to generate a simple decision boundary.

On the other hand, ensemble-based methods have recently been proposed to overcome the fundamental limitations of the single-one class classifier. These methods construct multiple weak single one-class classifiers and aggregate them. By doing so, they help reflect various characteristics of the target signal using smaller subsets. Isolation forest (IsoForest [39]) is one of the most well-known ensemble-based methods, which builds many random isolation trees. Isolation trees recursively partition the target signals using random split conditions until the target signals are isolated in the individual isolation trees.

IsoForest assumes that the fault signals require a small number of splits to be isolated because they have a different pattern than the target signal. By employing this approach, the anomaly score of IsoForest can be defined as the sum of the number of splits until it is isolated, and if the new signal has a small average number of splits, it is determined as a fault signal.

Although these traditional OCC methods have been well performed in the structured tabular data, for the unstructured data (e.g., time-series signal, image and video, and text-formatted data), appropriate features should be derived to apply them. However, these features obtained from independent feature extraction procedures to the OCC methods may not be specialized to detect anomalous patterns [40]. Hence, conventional OCC methods might not produce satisfactory fault signal detection performance in that the signal-typed data are the most representative unstructured data and require a careful feature extraction procedure.

2.2. Deep Neural Network-Based One-Class Classification Methods

As mentioned earlier, for more successful fault signal detection, features specialized for the fault signal detection tasks should be extracted. To this end, deep neural network structures, which are designed to generate optimized features for given learning tasks, have been successfully used for fault signal detection in recent years [11,12]. The deep neural network structure is composed of multiple intermediate layers (referred to as hidden layers) between input and output layers. These hidden layers generate features to minimize a loss function corresponding to the learning task without additional preprocessing or feature extraction procedures. Owing to this end-to-end learning manner, the deep neural network structure used for fault signal detection helps to achieve superior performance in that it can produce a specialized feature for anomalous signal detection tasks.

The deep neural network has been usually used for supervised learning tasks with numerous labeled data. However, in many real situations, fault signals obtained from machinery system faults or malfunctions rarely exist compared to the signals obtained from normal operating status, and thus, label information on the machinery operating condition might not be available [13,14]. Thus, these supervised deep neural networks cannot be used for fault signal detection tasks. For the unsupervised fault detection problem settings, unsupervised deep neural network-based OCC methods can be used. The unsupervised deep neural network-based OCC methods can be categorized as reconstruction-based [19,20], generative adversarial network (GAN)-based [41], and boundary-based methods, according to the way to generate the features of input signals.

Among them, the reconstruction-based OCC methods are based on autoencoder (AE)-based neural network structures. The AE-based neural network structures (e.g., stacked AE [11] and convolutional autoencoder (CAE [42,43])) attempt to generate meaningful features by reconstruction of the input signal data in the output layer. In the AE neural network, the input signals are embedded into smaller dimensions of latent features (referred to as encoding), and the latent features reconstruct the input signals (referred to as decoding). For a more accurate reconstruction of input signals into the output layer, the latent features should retain the intrinsic properties of input signals as much as possible. The stacked AE neural network [11] and CAE neural network [42,43] are the most widely used reconstruction neural network structures. These neural network structures can be used as OCC models by learning the neural network structures with only target signals (i.e., signals obtained from normal machinery status). Then, the errors between input signals and reconstructed outcomes can be used to calculate anomaly scores, quantifying the chances that the input signals are fault signals. That is to say; the target signals tend to have small reconstruction errors because the AE structures are constructed only using the training target signals. Conversely, the fault signals have larger reconstruction errors because the encoding structures are trained with no information on the fault signals, and the latent features cannot properly reconstruct the fault signals. In addition, the GAN-based OCC methods, which utilize neural network structures composed of both generator and discriminator, have recently been proposed. The generator attempts to create artificial signals as similar to input signals as possible, whereas the discriminator attempts to distinguish the artificial signals produced by the generator from the input signal as accurately as possible. Through an adversarial training scheme between the generator and discriminator, the GAN can produce artificial signals that have characteristics similar to target signals. Anomaly generative adversarial network (AnoGAN [44]), which is the most well-known GAN-based OCC method, is trained only using target signals, and thus, the generator produces a signal having similar patterns of normal operating status. Therefore, the generator can only generate an artificial signal similar to the training target signals, and the anomaly score of the newly collected signal is defined as the difference between the new signal and the generated signal from the generator of the AnoGAN. Finally, a signal having a large anomaly score is rejected as a fault signal. In addition to these methods, more recently, deep neural network structures have been successfully used for unsupervised anomaly detection problems in various unstructured datasets. For instance, Luo et al. proposed a sparse recurrent neural network (sRNN [45]) method to detect anomalous patterns in video datasets. Furthermore, self-supervised learning-based OCC methods have also been used for anomaly detection problems in image-formatted datasets [46,47].

Although these existing deep neural network-based OCC methods render reasonable results within the situations for which they were designed, no consensus exists regarding the best all-around performer in real situations. Firstly, reconstruction-based methods are designed to reproduce the original signal data simply by minimizing reconstruction errors. However, the reconstruction of the input signal might not be directly associated with the classification between target and fault signals [20,48]. Therefore, the reconstruction errors cannot be properly used as anomaly scores. Moreover, the GAN-based methods often generate inappropriate artificial signals because they often suffer from mode collapse problems [49]. The mode collapse problem is that the generator part tends to be trained to create only artificial signals with patterns that are almost similar to the most representative training target signal in order to minimize the discriminator's loss. The mode collapse problem causes a lack of diversity of generated signals, and it eventually yields poor fault signal detection performance of the GAN-based OCC methods. Finally, the most recently proposed deep neural network-based OCC methods are designed for specific domains, such as anomaly video detection (sRNN) or unusual image detection (self-supervised approach-based methods). Therefore, these methods might not be properly utilized for fault signal detection problems.

To overcome the above limitations, we propose to use deep support vector data description (deep SVDD), the most well-known boundary-based method, for fault signal detection problems. The deep SVDD method has shown superior anomaly detection performance in various cases, only using the target class. Please note that Ruff et al. [50] have extended the deep SVDD methods as deep semi-supervised anomaly detection (deep SAD) methods, which use a limited number of anomaly samples. This deep neural network-based OCC method encourages the anomaly samples to be located outside the hypersphere as much as possible. By doing so, the decision boundary can be improved to discriminate between target class and anomaly. However, this method cannot be applied to the problem setting considered in this study, in which only the target class (i.e., time series signals collected from normal status) is available in the training phase. Therefore, we propose to use the deep SVDD as a fault signal detection method.

3. Fault Signal Detection with Deep Support Vector Data Description

In this study, we propose to utilize the deep SVDD method for the fault signal detection framework. Hence, this section presents a more detailed description of the deep SVDD method.

The deep SVDD [12] builds a hypersphere enclosing as many target signals as possible in the latent feature space generated by deep neural network structures. In the deep SVDD, the input time-series signal $\mathcal{X} \subseteq \mathbb{R}^d$ (d denotes the dimension of time-series signal (i.e., the number of time points of the raw signal)) is mapped into the latent feature space $\mathcal{F} \subseteq \mathbb{R}^p$ (p is the length of latent feature vector) as $\phi(\cdot; \mathcal{W}) : \mathcal{X} \rightarrow \mathcal{F}$, the neural network structures whose set of weights are $\mathcal{W} = \{\mathcal{W}^1, \ldots, \mathcal{W}^L\}$. The $\mathcal{W}^\ell$ denotes the weights of ℓ-th hidden layer $\ell \in \{1, \ldots, L\}$ (L represents the number of total hidden layers). In the deep SVDD, the latent feature space is analogous to the feature mapping function used in the traditional SVDD model, and the features obtained from training the deep SVDD model comprise the latent feature space. Thus, the deep SVDD attempts to generate better latent feature vectors where the optimal hypersphere can be constructed. The overall training procedure on the deep SVDD is illustrated in Figure 1.

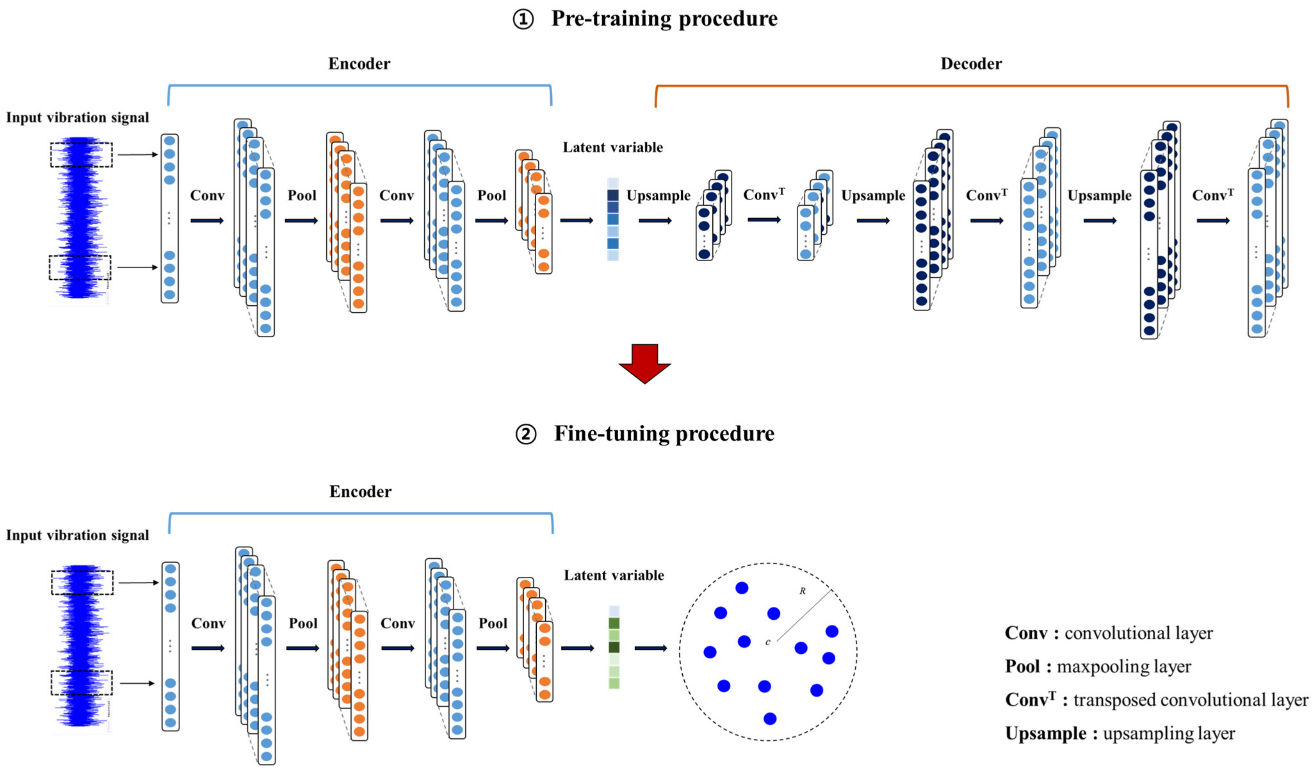


Figure 1. Graphical illustration of fault signal detection process using the deep SVDD.

As shown in Figure 1, the deep SVDD is trained by both pre-training and fine-tuning procedures. In order to more appropriately find the optimal latent feature space, deep SVDD first learns a stacked AE network as a pre-training procedure. The stacked AE network is composed of both encoding and decoding parts. The encoding part attempts to summarize the input time-series signal into smaller latent feature vectors, whereas the decoding part is used to reconstruct the input signals from the encoded latent feature vectors. The encoding part of the stacked AE is used as the initial network structure of the deep SVDD, and it helps the latent feature space can accommodate the intrinsic properties of the input time-series signal. By doing so, the initial weights of the neural network structure for the deep SVDD can be used to construct the hypersphere with a more accurate fault detection performance.

However, the stacked AE might not be localized characteristics of the time-series signal in that the hidden layers of the stacked AE are fully connected to each other. Hence, in this study, we propose to employ the one-dimensional CAE network instead of stacked AE because the one-dimensional CAE (1D-CAE) structures can successfully handle the time-series signal data owing to the convolutional layer and pooling layers [51,52]. In the 1D-CAE, the fully connected layers of the stacked AE are substituted as one-dimensional convolutional layers and pooling layers. The convolutional layer helps to consider localized properties within adjacent time points of input signals by using a convolutional kernel filter, and these convolutional kernel filters share weights across all regions of the input time-series signals. The output feature map of the j-th channel in the ℓ-th convolutional layer, f_j^{ℓ}, is calculated as follows:

$$f_j^{\ell} = \sigma\left(\sum_{m=1}^{C^{\ell-1}} f_m^{\ell-1} * \mathcal{W}_{m,\,c}^{\ell} + b_c^{\ell}\right),\ c = 1,\ 2,\ \ldots,\ C^{\ell},\ \ \ell = 1, 2\ldots, L, \tag{3}$$

where $f_m^{\ell-1}$ is the output feature map of the m-th channel in the (ℓ-1)-th layer transformed by the kernel filter $\mathcal{W}_{m,c}^{l}$ of the c-th channel in the ℓ-th layer. C^{ℓ} is the number of convolutional channels in ℓ layer and b_c^{ℓ} is the bias of c-th channel in the ℓ-th layer. In addition, * is the convolutional operator, and $\sigma(\cdot)$ denotes the nonlinear activation function. Hence, the output feature map having C^{ℓ} number of convolutional channels is computed. In general, each of the convolutional layers is followed by a pooling layer, which integrates adjacent values within the feature map into one value. Among various pooling manners, max-pooling, which picks the maximum value within a local region of the input feature map, is the most widely used. By employing the pooling layer, the size of the feature map and computational burden can be reduced. Both convolutional and max-pooling layer pairs (referred to as a convolutional block) can be repeated multiple times in the encoding part. At the end of the encoding part, the feature maps obtained by multiple convolutional blocks are flattened into a one-dimensional vector, and the flattened vector is finally mapped to a smaller latent features vector (dimension of latent feature vector (i.e., the number of latent features) is denoted as q) by the fully connected layer.

Then, in the decoding part, the input time-series signals are reconstructed from the latent variables of input time-series signals. The decoding part of the 1D-CAE comprises symmetrically arranged layers to the encoding part. In other words, the q-dimensional latent features vector is connected to a fully connected layer and reshaped into a one-dimensional feature map, and they are inversely reconstructed by transposed convolutional layers and up-sampling layers, which substitute the convolutional layers and max-pooling layers, respectively. The transposed convolutional layer expands the input feature map, conversely to the convolutional layer in the encoding part. The input feature map of the decoding part (i.e., the latent features vector) is transformed by a transposed convolutional layer as follows:

$$\overset{\sim\ell}{f}_m = \sigma\left(\sum_{m=1}^{C^{\ell-1}} \overset{\sim\ell-1}{f}_m \circledast \overset{\sim\ell}{\mathcal{W}}_{m,c} + \overset{\sim\ell}{b}_c\right),\ c = 1,\ 2,\ 3,\ \ldots,\ C^{l},\ \ \ell = 1, 2\ldots, L, \tag{4}$$

where $\circledast$ denotes the transposed convolutional operator which enlarges the input feature map, and $\overset{\sim\ell-1}{f}_m$ and $\overset{\sim\ell}{f}_m$ represent the output feature map of the k-th channel in the (ℓ-1)-th layer and the c-th channel in the ℓ-th layer, respectively. Further, $\overset{\sim\ell}{\mathcal{W}}_{m,c}$ denotes the transpose convolutional kernel filter of the c-th channel in the ℓ-th layer and $\overset{\sim\ell}{b}_c$ is the bias of the c-th channel in the ℓ-th layer. All the weights of the 1D-CAE are obtained by minimizing the reconstruction errors between the input time-series signal and its reconstructed ones. The reconstruction error between input time-series signal, $\mathcal{X}$, and reconstructed signals, $\tilde{\mathcal{X}}$, $L\left(\mathcal{X},\ \tilde{\mathcal{X}}\right)$, as follows:

$$L\left(\mathcal{X},\ \tilde{\mathcal{X}}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(\mathcal{X}_i - \tilde{\mathcal{X}}_i\right)^2, \tag{5}$$

where $\mathcal{X}_i$ and $\widetilde{\mathcal{X}}_i$ are the *i*-th input time-series signal and its reconstructed outcomes, respectively, and n is the number of total input time-series signals. By the encoding part of the 1D-CAE, the initial latent feature vectors can retain the intrinsic properties of input time-series signals. Hence, the weight set in the encoding part, denoted as $\mathcal{W}$, obtained from the 1D-CAE training procedure are used as initial weights of the deep support vector data description model $\phi(\cdot;\mathcal{W})$.

Although the latent feature vector of the encoding part of the 1D-CAE neural network structure can accommodate intrinsic properties of training target time-series signals, it might not be optimized to detect anomalous signals. Thus, the deep neural network structure should be improved for fault signal detection tasks. To this end, the weight set W obtained from pre-training is updated to be optimized for the fault signal detection task. The deep SVDD performs subsequent fine-tuning procedures to minimize a loss function formulated for the fault signal detection task. The deep SVDD models can be divided into soft-boundary deep SVDD and one-class deep SVDD, according to how the loss function is defined.

(1) Soft-boundary deep SVDD

In the Soft-boundary deep SVDD model, the loss function is defined as follows:

$$\underset{R,\mathcal{W}}{minimize} R^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\max\{0, \|\phi(x_i;\mathcal{W}) - c\|^2 - R^2\} + \frac{\lambda}{2}\sum_{\ell=1}^{L}\left\|\mathcal{W}^{\ell}\right\|_F^2, \tag{6}$$

R represents the radius of hypersphere in the latent feature space, and $\phi(x_i;\mathcal{W})$ denotes the mapped signal to the q-dimensional latent feature space by neural network structures. In addition, c is the center of the hypersphere, and $\mathcal{W}$ represents the weight set of the neural network structures, $\{\mathcal{W}^1,\ldots,\mathcal{W}^L\}$. In this loss function, the first term minimizes R^2 to find the smallest volume of the hypersphere that can enclose most of the target signal data. The second term is a slack term that allows for a few false positive errors that some mapped signals can be located outside the hypersphere. By adding the second term, the deep SVDD encourages the exclusion of a couple of anomalous target signals and prevents too many fault signals from being accepted as target signals. In this term, the slack hyperparameter $\nu \in (0,1]$ controls the trade-off between the size of the hypersphere and errors of the deep SVDD model. Finally, the third term is a weight decay regularizer for neural network weights $\mathcal{W}$ with the hyperparameter $\lambda > 0$, where $\|\cdot\|_F^2$ indicates the Frobenius norm. By adopting the third term, the overfitting problem of the deep neural network structures can be alleviated.

(2) One-class deep SVDD

Under the assumption that the training data comprise only target signals, minimizing the size of the hypersphere enclosing most of the target signals in the latent feature space can be regarded as minimizing the average distances from the center of the hypersphere to the target signals. Thus, the first and second terms in the soft-boundary deep SVDD can be integrated as the sum of the average distance between the center and all training target signals in the latent feature space. Then, the one-class deep SVDD model can be formulated as follows:

$$\underset{R,\mathcal{W}}{minimize}\frac{1}{n}\sum_{i=1}^{n}\|\phi(x_i;\mathcal{W}) - c\|^2 + \frac{\lambda}{2}\sum_{\ell=1}^{L}\left\|\mathcal{W}^{\ell}\right\|_F^2. \tag{7}$$

In the above equation, the first term attempts to minimize the average distance between the center and all training target signals in the q-dimensional latent feature space. Through this term, the neural network weights $\mathcal{W}$ can be updated to find latent feature space more specialized for fault signal detection in that it encourages the building of a compact decision boundary on the target signals. Compared with the soft-boundary deep SVDD, one-class deep SVDD can build ma more compact decision boundary by ignoring the slacks in Equation (5). Besides, the second term helps to prevent the overfitting risk, the same as the soft-boundary deep SVDD formulation. Finally, one-class deep SVDD does not require hyperparameter in that the first and second terms of the soft-boundary deep SVDD are integrated.

Please note that in the deep SVDD, the hypersphere center c should not be a free variable to prevent trivial solutions from having all zero values. That is to say, if the hypersphere center c is allowed to update during training of deep SVDD, c is estimated at zero values, and the hypersphere radius also nearly converges into zero. This problem is referred to as hypersphere collapse [12]. To avoid this problem, the hypersphere center c is calculated as the mean vector of the mapped signal $\phi(x_i; \mathcal{W})$ into the initial latent space obtained from the pre-training procedure for the encoding part of 1D-CAE. Once it is calculated, the hypersphere center c is then fixed during the fine-tuning procedure. In addition, we also do not employ bias terms and bounded activation functions to prevent the hypersphere collapse problem, as suggested by Ruff et al. [12].

After the deep SVDD structures are trained, they are finally used to determine whether newly collected signals are faulty or not. To this end, the anomaly score for the new signal x_{new}, denoted as $s(x_{new})$ and quantified as follows:

$$s(x_{new}) = \left\| \phi\left(x_{new}; \mathcal{W}^*\right) - c \right\|^2, \quad (8)$$

where $\mathcal{W}^*$ are the optimal parameter derived from training deep SVDD, $\phi(x_{new}; \mathcal{W}^*)$ is the mapped signal x_{new} into the q-dimensional latent feature space. If new signal x_{new} having large anomaly score is deemed as a fault signals, and an appropriate action should be taken to cope with abnormality on the machinery status. Conversely, for the new signal having a small score, it is classified as a target signal, and any actions are not taken.

4. Experimental Study

4.1. Experimental Settings

To demonstrate the superiority of the deep SVDD model for the fault signal detection problems, we performed an experimental study with signal datasets in three cases. First, we utilized the two datasets provided by Case Western Reserve University (CWRU) and Paderborn University (PU), which are the most widely used benchmark fault signals used in a number of previous studies. In addition to these two benchmark datasets, we collected both vibration and acoustic signal datasets collected from our own rolling element experiment platform. In this study, we used the time series signals collected from rolling element bearing because the it is the most representative rotating machinery. Hence, through this experimental study, we demonstrated the applicability of the deep SVDD in real world machinery fault detection problems. More detailed description of each signal datasets are presented as follows:

Case 1: Case Western Reserve University (CWRU) dataset—Vibration signal (The CWRU benchmark dataset is available at: https://engineering.case.edu/bearingdatacenter/download-data-file (accessed on: 20 December 2023)).

The CWRU bearing dataset was collected from the accelerometer sensor attached to the bearing installed in fan end side of the test rig, as depicted in Figure 2.

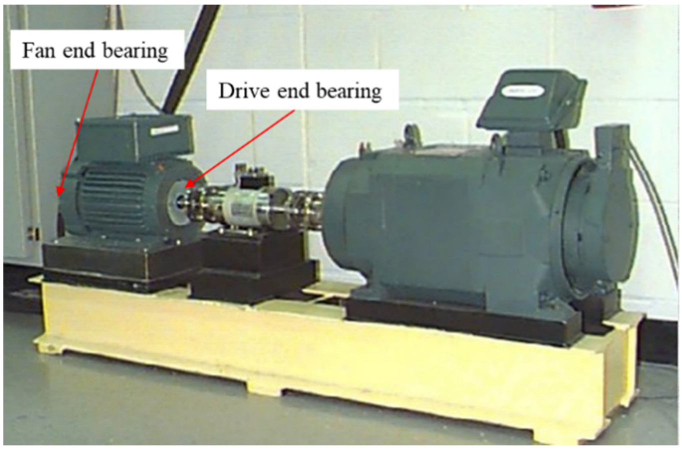


Figure 2. Test rig for the CWRU dataset (This figure is available at: https://engineering.case.edu/bearingdatacenter/download-data-file (accessed on 20 December 2023)).

In this study, we used the signal datasets collected by the accelerometer sensor attached in the fan end side. These benchmark signals are collected from various operating scenarios by changing the rotating speeds and vertical loads because vibration signals collected from different operating settings have different patterns. Table 1 shows the operating settings of four scenarios in the CWRU dataset.

Table 1. Operating scenarios in CWRU benchmark signal datasets.

	Rotating Speed	**Vertical Load**
Scenario 1	1797 RPM	0 HP (No pressure)
Scenario 2	1772 RPM	1 HP
Scenario 3	1750 RPM	2 HP
Scenario 4	1730 RPM	3 HP

The raw signals in this benchmark dataset were recorded at a sampling rate of 12,000 Hz per second (i.e., each raw signal was recorded as 12,000 time points per second). In addition, the dataset includes vibration signals from four types of bearing conditions: normal, inner race fault, outer race fault, and ball element fault. In these benchmark datasets, each fault type has four intensities corresponding fault diameters (e.g., 7 inch (weak fault), 14 inch (medium fault), 21 inch (strong fault)). Therefore, the CWRU dataset has ten bearing condition types, and signals from the normal condition are considered as target signals for OCC methods.

In this benchmark dataset, the ten vibration signals corresponding to each condition type were collected in approximately 10 to 20 s. In order to comprise the datasets having a sufficient number of signals to train the fault signal detection model, we segmented each raw signal into smaller ones. To this end, each raw signal is sliced in order for individual signals to have 1024 time points, and the sliced signals are overlapped each other with 768 time points. Then, the final fault signal detection model is constructed with the segmented signals. An example of these segmented signals to each bearing type is presented in Figure 3. As shown, the signals generated by the fault-bearing conditions have different patterns compared to the signals obtained from normal conditions. In addition, fault signal detection datasets of four scenarios in CWRU benchmark signal datasets are summarized in Table 2.

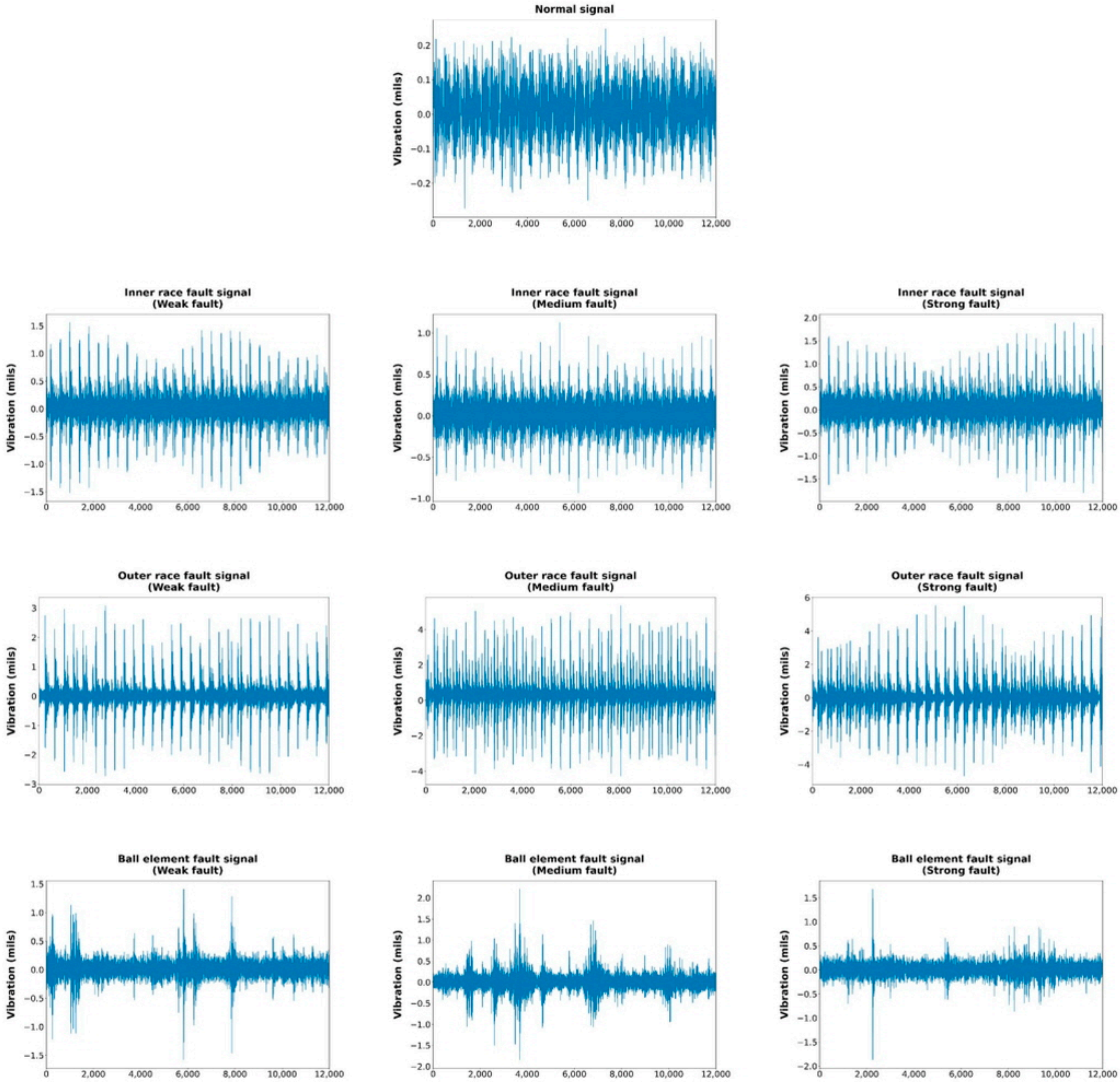


Figure 3. Example segmented signals of all bearing conditions in Scenario 1 of CWRU datasets.

Table 2. The number of target and fault signals in CWRU benchmark signal datasets.

	The Number of Target Signals (Normal Signals)	The Number of All Fault Signals
Scenario 1	952	6108
Scenario 2	1888	5160
Scenario 3	1892	5172
Scenario 4	1896	5180

Case 2: Paderborn University (PU) dataset—Vibration signal (The PD benchmark dataset is available at: https://mb.uni-paderborn.de/kat/forschung/kat-datacenter/bearing-datacenter/data-sets-and-download (accessed on 20 December 2023)).

The Paderborn University dataset was collected by Lessmeier et al. [53]. This dataset is comprised of both vibration and current time series signals collected from the rolling bearing test rig shown in Figure 4.

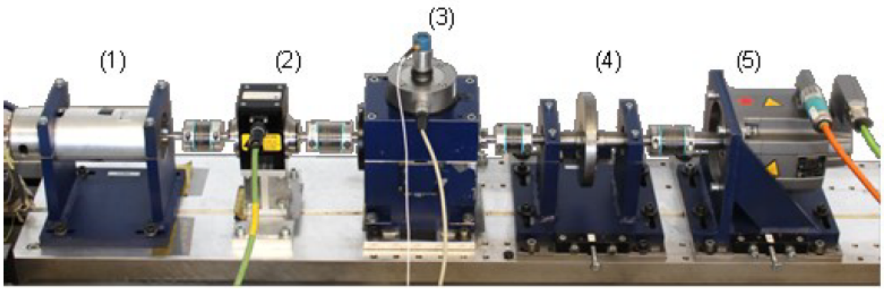


Figure 4. Test rig for the PU dataset. It consists of five modules: (1) electric motor, (2) torque-measurement shaft, (3) rolling bearing test module, (4) flywheel, and (5) load motor [53].

The vibration signals are collected by the accelerometer sensor installed at the top end of the rolling bearing module. These benchmark signals are collected from various operating scenarios by changing (1) rotating speed, (2) radial force onto the bearing, and (3) load torque in the drive train because vibration signals collected from different operating settings have different patterns. Table 3 shows the operating settings of four scenarios in the PU dataset.

Table 3. Operating scenarios in PU benchmark signal datasets.

	Rotating Speed	Radial Force	Load Torque
Scenario 1	1500 RPM	1000 N	0.7 Nm
Scenario 2	900 RPM	1000 N	0.7 Nm
Scenario 3	1500 RPM	1000 N	0.1 Nm
Scenario 4	1500 RPM	400 N	0.7 Nm

The dataset includes vibration signals from various types of bearing conditions: normal (Healthy), artificial inner race fault (AIR), artificial outer race fault (AOR), real inner race fault (RIR), and real outer race fault (ROR). In addition, the artificial fault types (AIR and AOR) are generated by electrical discharge machining (EDM), drilling, or electric engraver pitting, whereas real fault types (RIR and ROR) are made by accelerated life testing. Finally, among the six normal condition bearings having different operation times (bearing codes are K001, K002, K003, K004, K005, and K006), we employed the K003 bearing, which is operated for one hour. Hence, the normal signals obtained from K003 are used as target signals for OCC methods.

The raw signals in the PU dataset were collected for approximately 4 s at a sampling rate of 64,000 Hz per second (i.e., each raw signal was recorded 256,000 times). Thus, similar to the CWRU dataset, each raw signal is sliced in order for individual signals to have 2048 time points (no overlap between signals) to comprise the datasets having sufficient signals to train the fault signal detection model. Then, the final fault signal detection model is constructed with the segmented signals. An example of these segmented signals to each bearing condition type is presented in Figure 5.

As shown, the signals generated by the fault-bearing conditions have different patterns and scales compared to the signal obtained from normal conditions. In addition, fault signal detection datasets of four scenarios in PU benchmark signal datasets are summarized in Table 4.

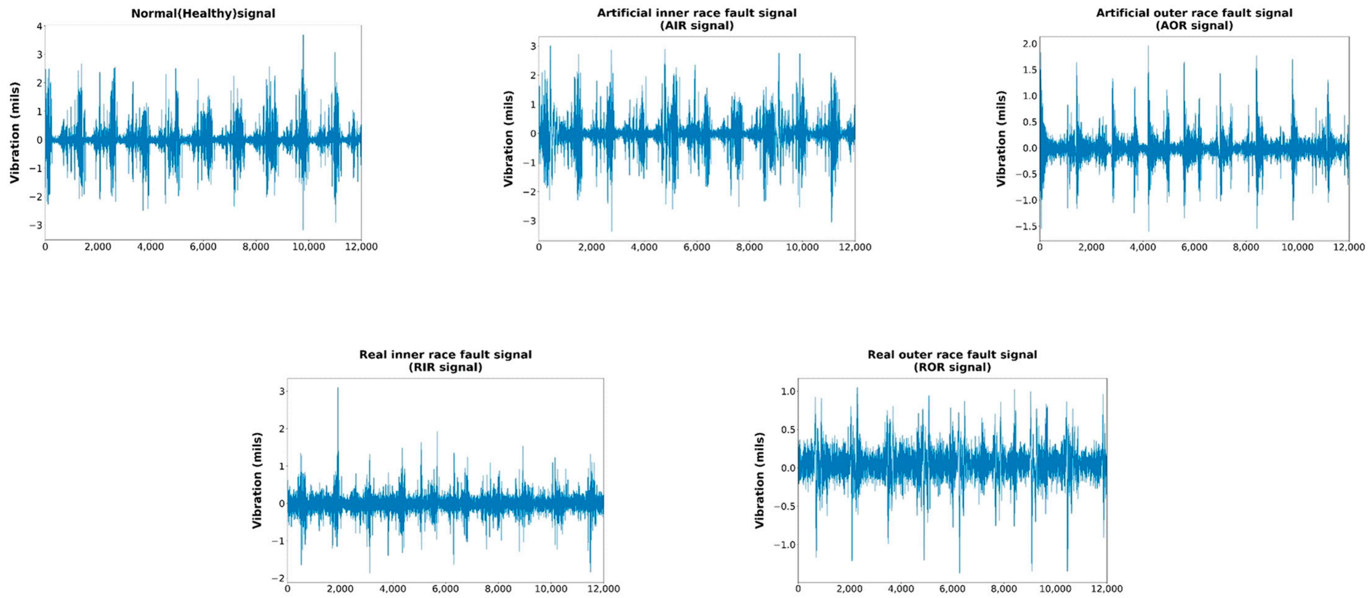


Figure 5. Example segmented signals of all bearing conditions in Scenario 1 of PU datasets.

Table 4. The number of target and fault signals in PD benchmark signal datasets.

	The Number of Target Signals (Normal Signals)	**The Number of All Fault Signals**
Scenario 1 Scenario 2 Scenario 3 Scenario 4	2500	51,660

Case 3: Rolling element experiment platform dataset—Vibration and acoustic signal

In addition to the benchmark datasets, we also collected both vibration and acoustic signals from our own rolling element-bearing experimental platform, presented in Figure 6.

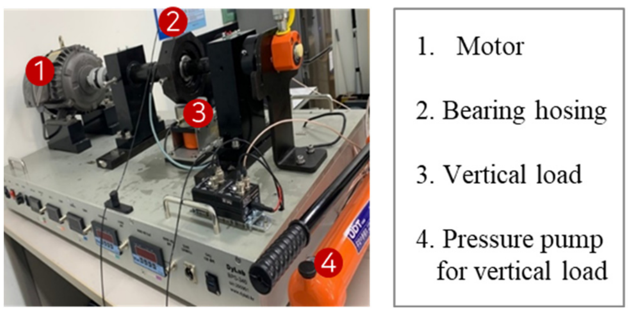


Figure 6. Rolling element experimental platform.

In this bearing experimental platform, both the vibration and acoustic signals were simultaneously collected from the accelerometer sensor and preamplifier sensor, respectively, with a sampling frequency of 10,240 Hz per unit second. In this case, individual vibration and acoustic signals are collected per unit second. Hence, individual signals have 10,240-time points. In this experimental platform, a deep groove ball bearing is installed in the bearing housing of the simulator, and its sizes of inner side diameter, outer side diameter, and width are 35 mm, 72 mm, and 17 mm, respectively. In addition, the rotating speed of the bearing is 1200 RPM (revolutions per minute), and the vertical loader presses a bearing housing 150 kg (kilogram-force).

Similar to the CWRU dataset, the signals are collected from normal, inner race fault, outer race fault, and ball element fault conditions because these three faults are the most representative fault types [54,55]. These three faults in the bearing were generated by an electrical discharge machine. Furthermore, we also considered two fault intensity levels (i.e., strong and weak) by different defect diameters for each fault type because the signal patterns and amplitude scales differ from fault intensities. The examples of the vibration and acoustic signals collected from various bearing conditions are presented in Figure 7 and Figure 8, respectively.

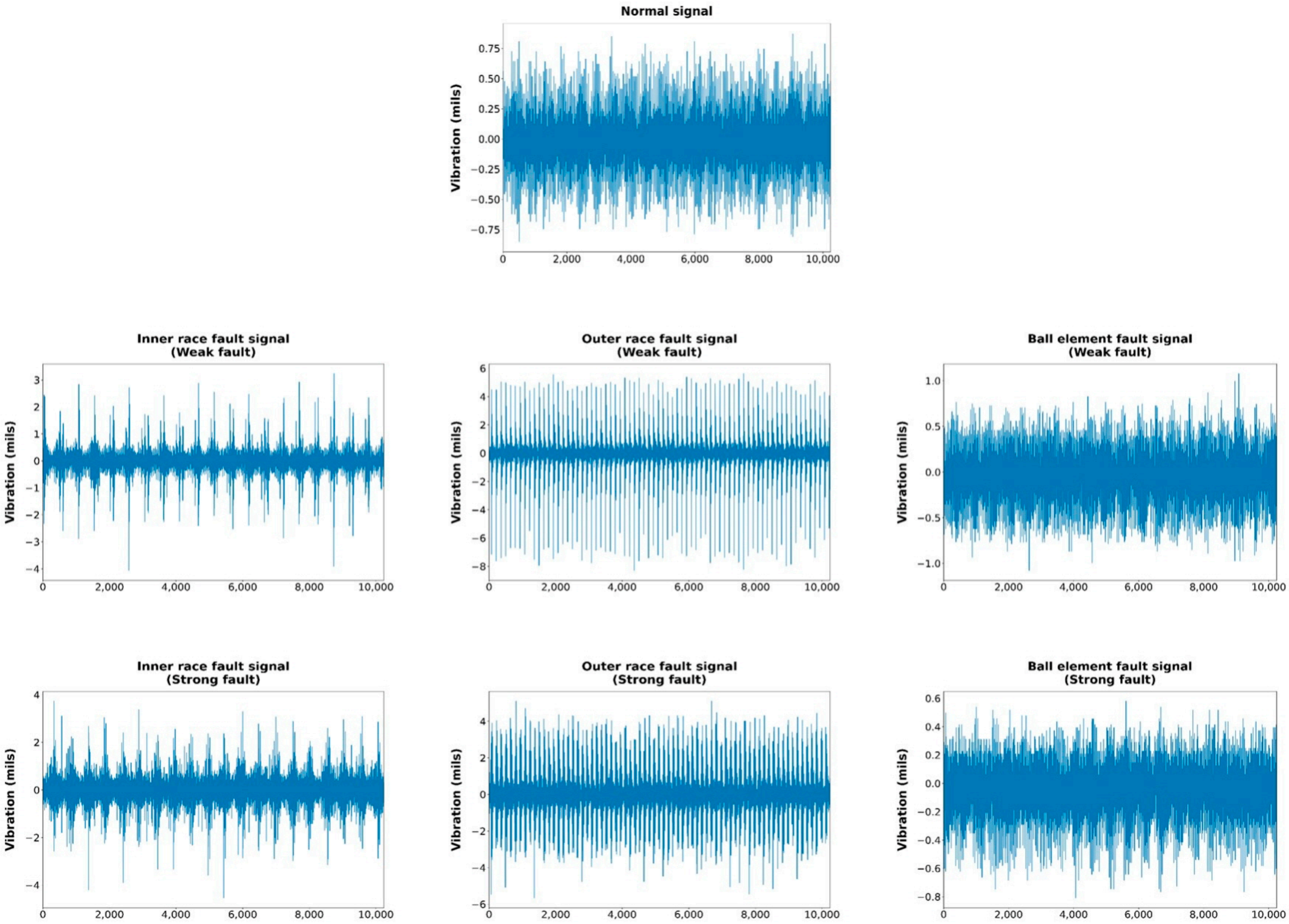


Figure 7. Example of vibration signals obtained from our own rolling element experiment platform.

The number of signals collected from both normal and all fault conditions is presented in Table 5.

Table 5. The number of target and fault signals obtained from our own rolling element experiment platform.

Signal Types	The Number of Target Signals (Normal Signals)	The Number of All Fault Signals
Vibration Acoustic	4850	21,050

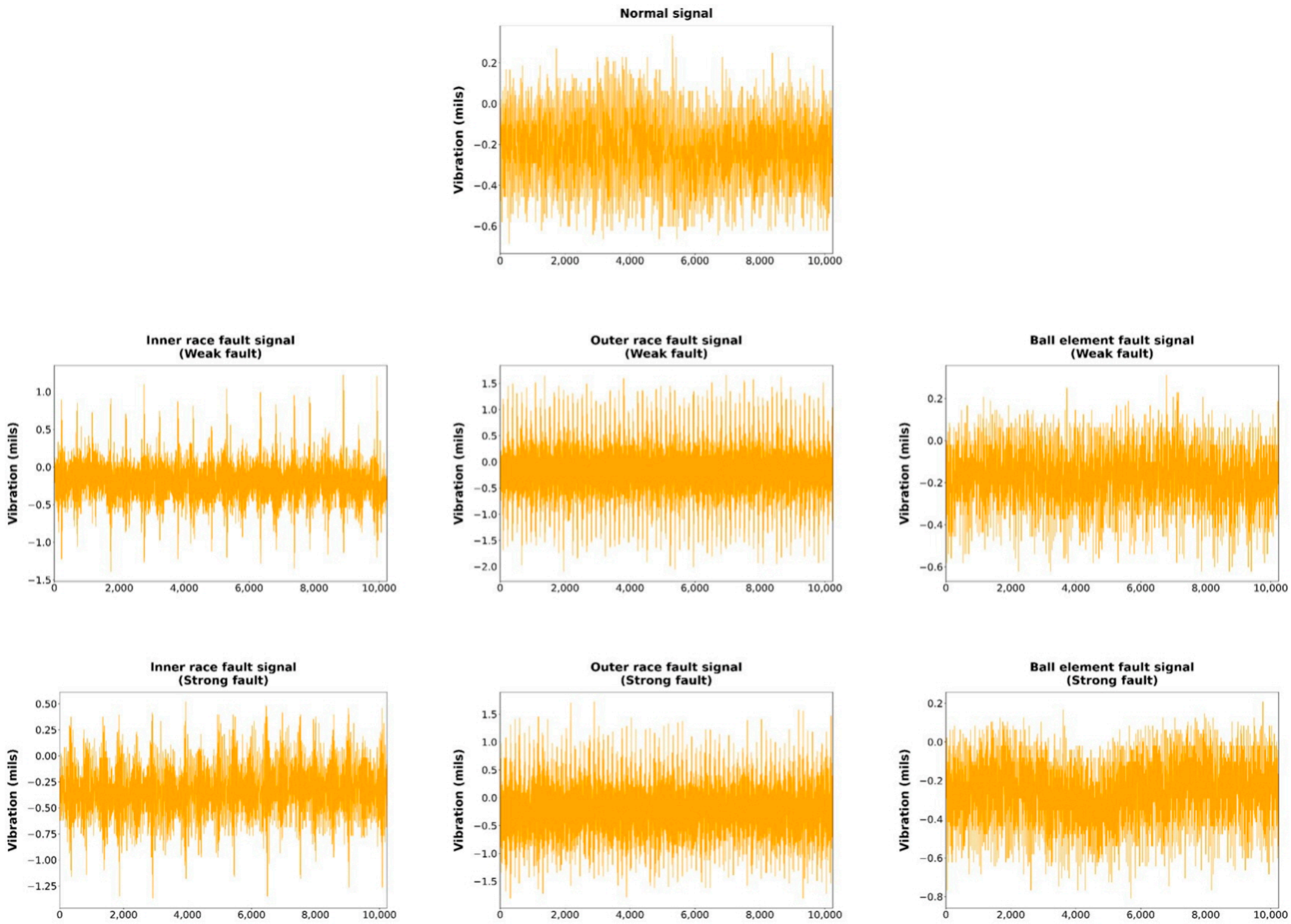


Figure 8. Example of acoustic signals obtained from our own rolling element experiment platform.

In this study, we considered the traditional OCC methods as follows: SVDD, KDE, and IsoForest, which are the most representative boundary-based, probability distribution-based, and ensemble-based methods. For these traditional OCC methods, additional feature extraction procedures should be performed beforehand. To this end, we used three feature learning methods, principal component analysis (PCA), stacked autoencoder (stacked AE), and convolutional autoencoder (CAE). Then, we applied the SVDD, KDE, and IsoForest to the latent features obtained from those three methods. For the SVDD, we used the radius basis function (RBF) kernel having a parameter γ. In addition to γ, the traditional SVDD model requires other regularization hyperparameter C. In this study, we explored the hyperparameter pair (γ, C) within the given range: $\gamma \in \{2^{-10}, 2^{-9}, \ldots, 2^{-1}\}$ and $C \in \{0.01, 0.02, 0.03, \ldots, 0.1\}$, respectively. For KDE, we also used a Gaussian kernel as the kernel function having bandwidth h, and the hyperparameter h was chosen within the range of $h \in \{2^{0.5}, 2^1, \ldots, 2^5\}$. Finally, for the IsoForest, we empirically selected the number of isolation trees, B, within the range of $B \in \{50, 100, 200, 500, 1000\}$. For these traditional OCC methods, we chose the optimal hyperparameter achieving the best performance within these candidate sets. The hyperparameter optimization for the OCC methods is not easy tasks because, as mentioned earlier, the fault detection is basically unsupervised problem settings [36,37]. In other words, in these problem settings, the label information is not available. To the best of our knowledge, there is lack of systematic ways to optimize the hyperparameters in fault detection problems. Hence, as in previous studies [35,56], we also picked the hyperparameters showing the highest AUROC values.

We also considered the following deep OCC methods for fault signal detection: reconstruction-based, GAN-based, and boundary-based methods. For the reconstruction-based methods, we trained stacked AE and CAE and computed the reconstruction error,

$\|x - \hat{x}\|^2$, between the input signal (x) and reconstructed outcome ($\hat{x}$) as an anomaly score. In this study, we refer to these reconstruction-based OCC methods as anomaly scores with reconstruction errors of the stacked AE (ReSAE) and the CAE (ReCAE), respectively. For the GAN-based method, we utilized an AnoGAN-based convolutional neural network structure to more effectively handle the signal-formatted data. We set the number of latent features of the AnoGAN model as 256, as suggested by Schlegl et al. [44]. Finally, we applied the two deep SVDD models, soft-boundary deep SVDD and one-class deep SVDD, as boundary-based methods. In this study, we tried both stacked AE or CAE in the pre-training procedure of the deep SVDD model, in order to indicate the efficacy of convolutional blocks in order to analyze the signal data. In addition, to avoid the hypersphere collapse problem, we removed the bias term and used the leaky rectified linear unit activation function (Leaky ReLU). For the soft-boundary deep SVDD, we varied the hyperparameter ν within the range of $\nu \in \{0.01, 0.02, 0.03, \ldots, 0.1\}$, and selected one showed the best performance. The architecture details of the neural network structure for the deep SVDD are summarized in Table 6.

Table 6. Details of deep SVDD architecture.

Case	Architecture	Batch Size	Optimizer (Learning Rate)
CWRU dataset (Case 1)	8 × (5 × 1)-filters + max pooling + Leaky ReLU 4 × (5 × 1)-filters + max pooling + Leaky ReLU Dense layer of 16 units (i.e., the number of latent features (q) is 16)	32	Adam optimizer (η = 0.005)
PU dataset (Case 2)	8 × (5 × 1)-filters + max pooling + Leaky ReLU 4 × (5 × 1)-filters + max pooling + Leaky ReLU Dense layer of 16 units (i.e., the number of latent features (q) is 16)	128	
Rolling element experiment platform dataset (Case 3)	8 × (5 × 1)-filters + max pooling + Leaky ReLU 4 × (5 × 1)-filters + max pooling + Leaky ReLU Dense layer of 16 units (i.e., the number of latent features (q) is 16)	128	

In the current study, the neural network hierarchy structures, such as the number of convolutional and pooling layers and the sizes of filters, are specified the same as in the original deep SVDD literature [12]. As suggested in [12], these network structures have shown reasonable results in various cases. Accordingly, we also adopted the same network hierarchy structures as [12]. Please note that the bounded activation functions, including the sigmoid and hyperbolic tangent functions, tend to cause the hypersphere collapse problem [12]. Therefore, we employed the Leaky ReLU, which is the most well-known unbounded activation function. As for the deep SVDD, this method is implemented by using open source code released to the public (The deep SVDD by using open source code available at: https://github.com/lukasruff/Deep-SVDD-PyTorch (accessed on 20 December 2023)). Finally, to conduct all experiments, we utilized an Intel® Core(TM) i5-9400F CPU @ 2.90 GHZ and NVIDIA GeForce RTX 2060 with 32 GB of RAM. All the DNN-based OCC methods were implemented with the GPU-accelerated Pytorch library (version 1.12.1) in Python (version 3.9.13).

As mentioned earlier, the training dataset consists of only normal signals, and the testing dataset is composed of the signals both in normal operating conditions and all fault types. To this end, we composed the training dataset as randomly selected 80% of the normal signals, and the testing dataset is composed of both all fault signals and the remaining 20% of the normal signals. Then, we evaluate the fault signal detection performance of each method through the area under the receiver operating characteristic curve (AUROC) on the testing dataset. The ROC (receiver operating characteristic) curve represents the trade-off between two types of errors, true positive and false positive, in fault signal detection problems. The true positive is the accuracy of classifying target signals as normal signals correctly, whereas the false positive is the error of classifying target signals as fault signals. It should be noted that a high true positive rate with a low false positive rate results under the large threshold values, while a low true positive rate with a high false positive rate results if the thresholds are set as small values. The ROC curve can be obtained by changing the threshold to make a false positive rate from zero to one. The AUROC is the area under the ROC curve, and the larger AUROC value indicates better fault signal detection performance.

4.2. Experimental Results

Table 7 shows the comparative results of the AUROC values of all OCC methods considered in the current study. We reported the averages and standard deviations of the AUROCs from 10 repetitions of random splitting of training/testing datasets. In this table, the highest average AUROC values are highlighted in bold.

Table 7. Average AUROC values of all OCC methods over ten time-series signal datasets (Scenario 1 to 4 of Case 1, Scenario 1 to 4 of Case 2, and vibration and acoustic signals of Case 3).

Methods	CWRU Dataset (Case 1)				PU Dataset (Case 2)				Rolling Element Experiment Platform Dataset (Case 3)	
	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Vibration Signals	Acoustic Signals
PCA + KDE	**1.000 (0.000)**	0.986 (0.005)	0.992 (0.005)	0.993 (0.001)	0.340 (0.006)	0.790 (0.005)	0.780 (0.005)	0.827 (0.004)	0.665 (0.005)	0.603 (0.009)
PCA + IF	0.995 (0.002)	0.994 (0.000)	0.994 (0.000)	0.994 (0.000)	0.349 (0.036)	0.782 (0.007)	0.778 (0.013)	0.822 (0.004)	0.722 (0.003)	0.618 (0.004)
PCA + SVDD	**1.000 (0.000)**	0.994 (0.000)	0.994 (0.000)	0.994 (0.000)	0.393 (0.101)	0.791 (0.005)	0.776 (0.017)	0.825 (0.004)	0.735 (0.003)	0.637 (0.005)
AE + KDE	0.883 (0.172)	0.548 (0.047)	0.524 (0.031)	0.391 (0.054)	0.681 (0.015)	0.730 (0.036)	0.733 (0.024)	0.748 (0.023)	0.519 (0.113)	0.534 (0.023)
AE + IF	0.870 (0.168)	0.084 (0.097)	0.092 (0.102)	0.122 (0.044)	0.707 (0.009)	0.748 (0.012)	0.749 (0.013)	0.746 (0.008)	0.264 (0.124)	0.551 (0.025)
AE + SVDD	0.858 (0.159)	0.286 (0.118)	0.530 (0.085)	0.535 (0.084)	0.717 (0.011)	0.684 (0.057)	0.650 (0.082)	0.700 (0.042)	0.412 (0.165)	0.530 (0.038)
CAE + KDE	0.978 (0.028)	0.993 (0.001)	0.993 (0.001)	0.993 (0.002)	0.987 (0.001)	0.847 (0.054)	0.869 (0.015)	0.867 (0.054)	0.921 (0.009)	0.801 (0.042)
CAE + IsoForest	0.962 (0.072)	0.998 (0.036)	0.988 (0.036)	0.974 (0.072)	0.980 (0.002)	0.860 (0.046)	0.883 (0.026)	0.870 (0.061)	0.924 (0.009)	0.819 (0.061)
CAE + SVDD	0.976 (0.047)	0.985 (0.042)	0.999 (0.001)	0.974 (0.072)	0.980 (0.004)	0.877 (0.038)	0.895 (0.024)	0.882 (0.056)	0.794 (0.048)	0.622 (0.070)
ReSAE	0.992 (0.001)	0.996 (0.002)	0.996 (0.003)	0.994 (0.000)	0.068 (0.011)	0.347 (0.022)	0.320 (0.017)	0.358 (0.024)	0.809 (0.038)	0.569 (0.021)
ReCAE	0.999 (0.002)	0.999 (0.001)	0.998 (0.002)	0.998 (0.001)	0.048 (0.001)	0.327 (0.003)	0.303 (0.002)	0.333 (0.003)	0.732 (0.002)	0.569 (0.007)
AnoGAN	0.765 (0.024)	0.567 (0.036)	0.570 (0.037)	0.604 (0.095)	0.216 (0.116)	0.452 (0.139)	0.399 (0.101)	0.417 (0.207)	0.722 (0.014)	0.691 (0.042)
Soft-boundary deep SVDD (AE pre-trained)	0.980 (0.005)	0.994 (0.003)	0.985 (0.002)	0.989 (0.003)	0.792 (0.007)	0.823 (0.007)	0.760 (0.004)	0.791 (0.007)	0.819 (0.079)	0.643 (0.091)
One-class deep SVDD (AE pre-trained)	0.980 (0.005)	0.993 (0.003)	0.985 (0.003)	0.988 (0.004)	0.775 (0.005)	0.793 (0.007)	0.762 (0.003)	0.759 (0.006)	0.859 (0.103)	0.683 (0.028)
Soft-boundary deep SVDD (CAE pre-trained; Proposed)	0.985 (0.026)	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**	**0.991 (0.001)**	0.936 (0.007)	0.941 (0.008)	**0.940 (0.007)**	**0.985 (0.002)**	0.912 (0.011)
One-class deep SVDD (CAE pre-trained; Proposed)	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**	**1.000 (0.000)**	**0.991 (0.001)**	**0.939 (0.006)**	**0.944 (0.002)**	0.939 (0.009)	**0.985 (0.004)**	**0.922 (0.014)**

Table 7 shows that the traditional OCC methods with independently performed feature extraction yielded lower AUROC values than deep neural network-based OCC methods. Those results indicate that separate feature extraction procedures might not generate specialized features for fault signal detection tasks. Conversely, the deep neural network-based OCC methods generally perform better than traditional OCC methods owing to their end-to-end learning manners for the fault signal detection tasks. In the deep neural network structures, latent features are optimized for fault signal detection tasks in that these features are simultaneously generated to minimize the loss functions for the fault signal detection tasks. Therefore, it confirms that deep neural network-based methods are more appropriate for detecting fault signals than traditional OCC methods with separate feature extraction procedures. However, in the AnoGAN model, the generator part may poorly generate artificial target signals due to the mode collapse problem, and it results in undesirable fault signal detection performance. Besides, the reconstruction-based methods, ReSAE and ReCAE, also cannot perform better than the deep SVDD models because the reconstruction error of the input signal might not directly quantify the anomalous level of the input signal. Hence, the weights of the stacked AE or CAE should be updated for more accurate fault signal detection tasks. Conversely, both one-class deep SVDD and soft boundary deep SVDD models outperform other deep neural network-based methods because the more specialized features of fault signal detection can be derived as minimizing loss functions for anomaly detection tasks. It should be noted that the pre-training with CAE performs better than those with stacked AE because convolutional layers of the CAE network help to draw temporal information of time-series signals because they combine the signal values within adjacent time intervals. Therefore, the initial neural network structures of the encoding part in CAE can accommodate the intrinsic properties of time-series signal data, and it eventually helps to build more accurate decision boundaries for detecting fault signals.

In addition, we conducted a post-analysis on performance differences through a nonparametric statistical method. We applied the Wilcoxon signed rank test [57] regarding the statistical difference in AUROC values among all the above 16 methods. Based on *p*-values over all methods, the null hypothesis of performance equivalence between CAE pre-trained one-class deep SVDD (the top-ranked method) and other ones is tested. The results of the Wilcoxon signed rank test are provided in Table 8.

Table 8. Wilcoxon signed-rank test results.

Methods	Average Rank	*p*-Value	Hypothesis (α = 0.01)
One-class deep SVDD (CAE pre-trained; Proposed)	1.1	-	-
Soft-boundary deep SVDD (CAE pre-trained; Proposed)	1.9	0.1056	Not reject
CAE + IsoForest	5.9	0.0059	Reject
CAE + KDE	6.0	0.0058	Reject
CAE + SVDD	6.7	0.0020	Reject
PCA + SVDD	6.8	0.0089	Reject
Soft-boundary deep SVDD (AE pre-trained)	7.8	0.0020	Reject
PCA + IF	7.9	0.0058	Reject
One-class deep SVDD (AE pre-trained)	8.2	0.0020	Reject
PCA + KDE	8.6	0.0092	Reject
ReSAE	9.9	0.0059	Reject
ReCAE	10.0	0.0059	Reject
AnoGAN	12.7	0.0020	Reject
AE + KDE	13.1	0.0020	Reject
AE + IF	13.5	0.0020	Reject
AE + SVDD	13.6	0.0020	Reject

As shown in Table 8, both the CAE pre-trained one-class deep SVDD and soft-boundary deep SVDD methods attained the best or second average rank. In addition, the null hypothesis of performance equivalence was rejected with a significance level of $\alpha = 0.01$, and it implies that there is a significant difference in performance between the CAE pre-trained deep SVDD (proposed) and other methods. Consequently, these results denote that the proposed deep SVDD with CAE pre-training significantly outperformed the others. Finally, there is no statistical difference between the two deep SVDD methods pre-trained by CAE, which are not significant. In spite of their equivalent performance, we suggest using the one-class deep SVDD because it alleviates the difficulty of selecting the hyperparameter ν in the soft-boundary deep SVDD method.

In this study, we also conducted additional experimental studies to examine the effect of a number of latent features on the fault signal detection performance by varying the number of latent features from 16 to 256. The AUROC values over different numbers of latent features in all cases are presented in Figure 9.

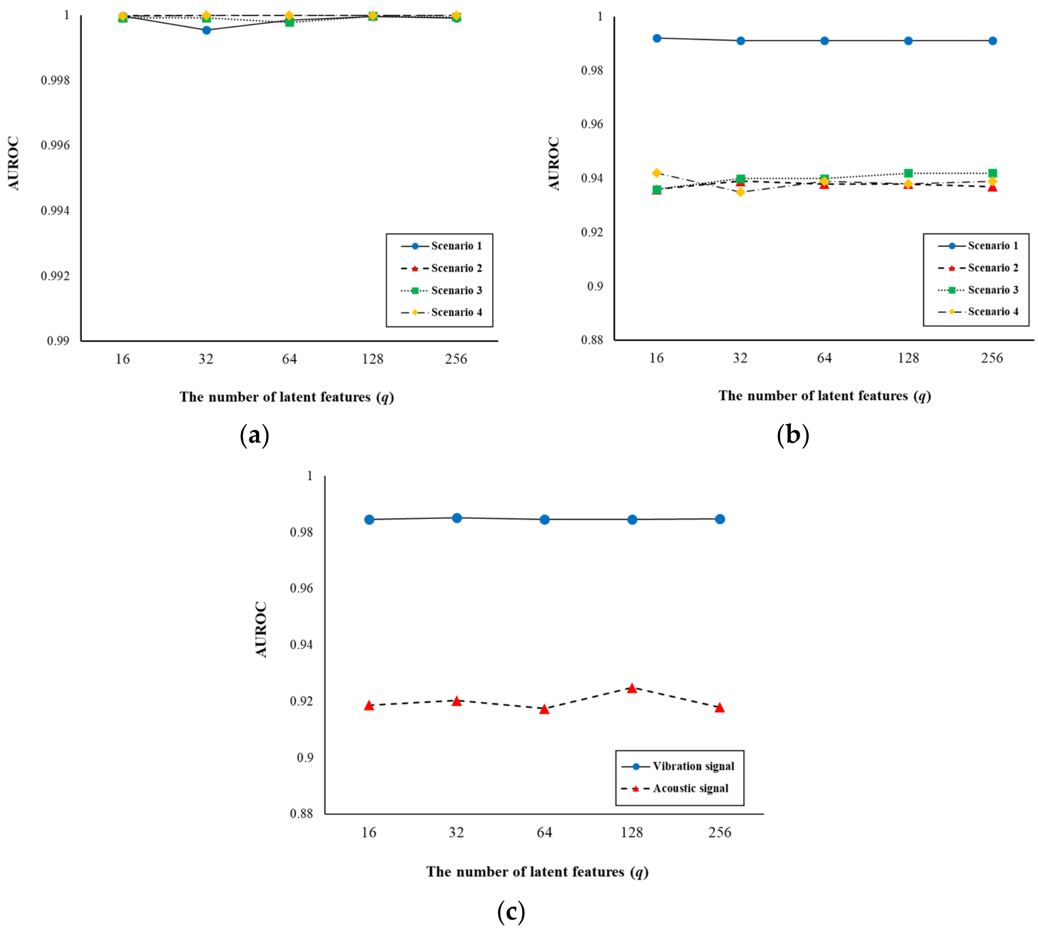
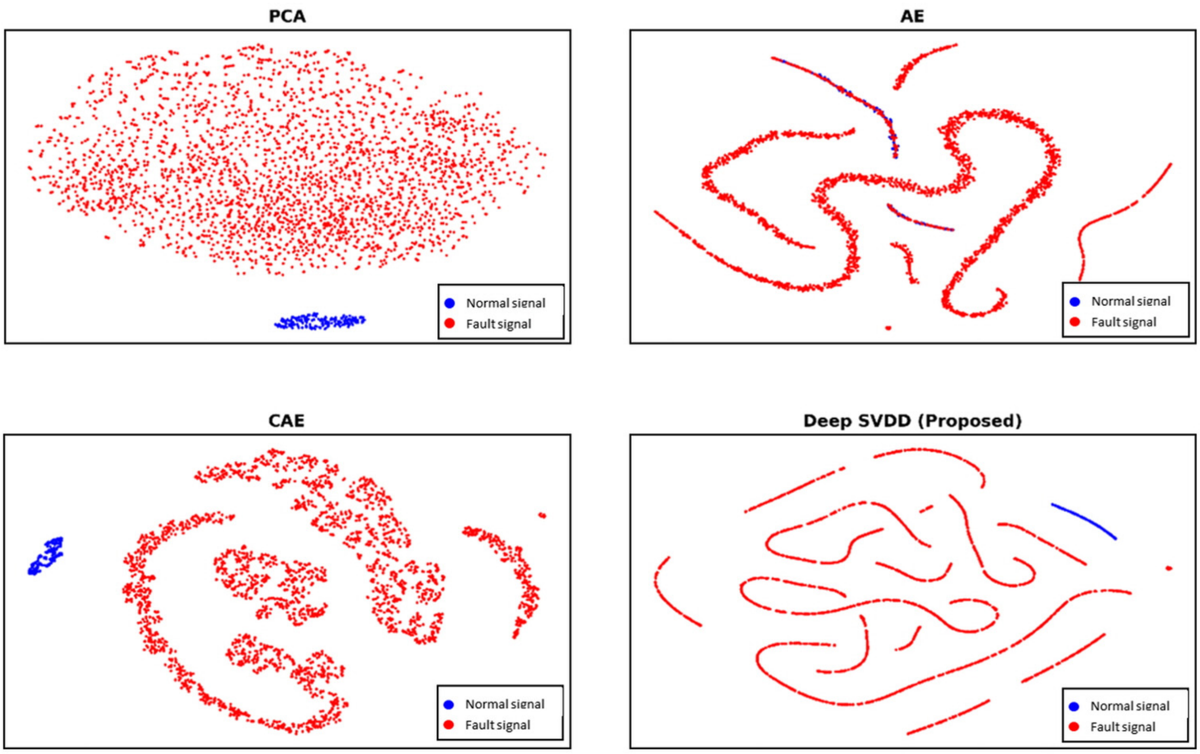


Figure 9. AUROC values over different numbers of latent features in (**a**) the CWRU dataset (Case 1), (**b**) the PU dataset (Case 2), and (**c**) the Rolling element experiment platform dataset (Case 3).

As shown in Figure 9, the fault signal detection performance of the deep SVDD model rarely differs by changing the number of latent features. Therefore, we specified the number of latent features as 16 in order to prevent the overfitting problem and mitigate the computational burden of the deep SVDD model.

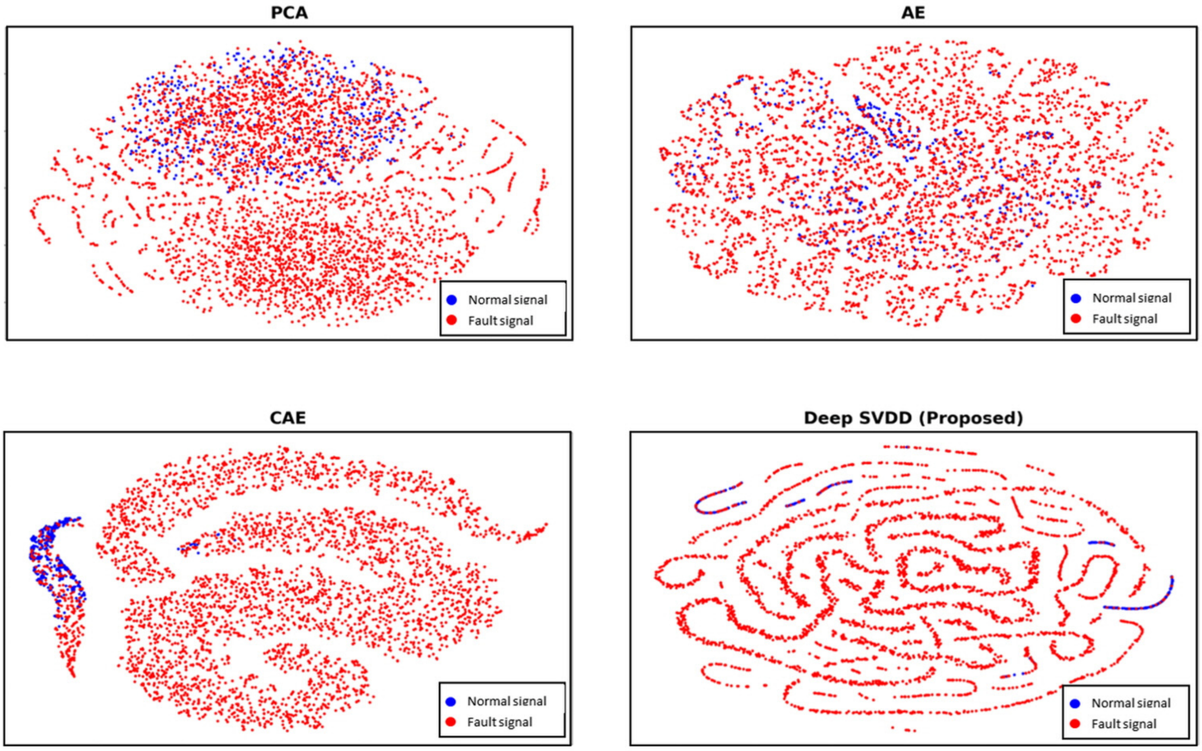
Finally, in order to demonstrate the advantage of the deep SVDD method regarding fault signal detection, we compared it with other feature extraction methods (PCA, AE, and one-dimensional CAE). To this end, we visualized these 16 latent features extracted from these four methods by facilitating the t-distributed stochastic neighborhood embedding (t-SNE [58]) technique. Figure 10 shows the two-dimensional t-SNE plots of latent features obtained from four feature extraction methods.

As shown in Figure 10, in the latent features obtained from PCA and AE, normal and fault signals are quite overlapped each other. The PCA cannot deal with the nonlinear patterns, and thus, it might not properly deal with the complex time series signal data. Moreover, in the AE, the layers are fully connected to each other, and these fully connected structures also cannot accommodate the temporal properties of the time series signals. For these reasons, these two feature extraction methods cannot accurately detect the fault signals. Conversely, the one-dimensional CAE and deep SVDD pre-trained by one-dimensional CAE more clearly discriminate the fault signals from normal ones than PCA and AE. These results indicate that one-dimensional CAE can more properly handle the time series signals because the convolutional blocks (i.e., convolutional-max pooling layer pair) in CAE help to consider the temporal properties of the time series signals [51,52]. On the other hand, latent features obtained from the deep SVDD more clearly separate the fault signals and normal signals than those from one-dimensional CAE. In the fine-tuning procedure of the deep SVDD, latent features are updated to be specialized for fault signal detection. Therefore, the deep SVDD method can generate more useful features for fault signal detection tasks than only using one-dimensional CAE.

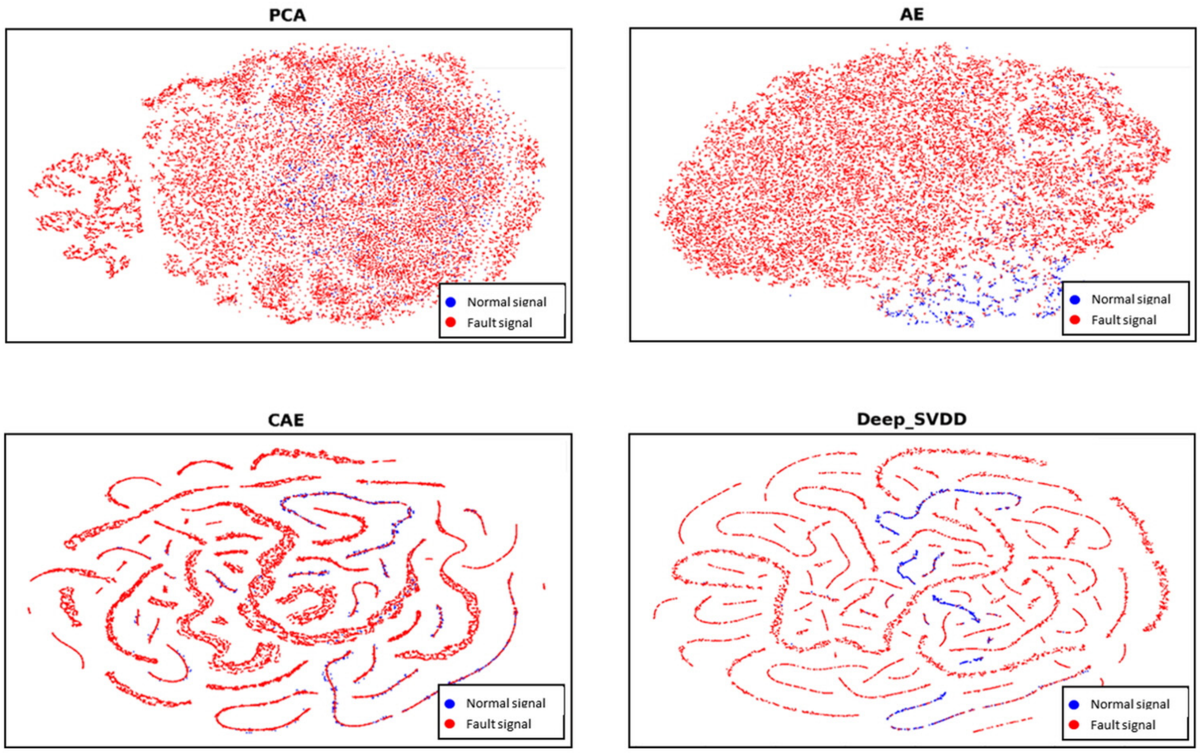


(**a**)

Figure 10. *Cont*.



(**b**)



(**c**)

Figure 10. Two-dimensional t-SNE plot of 16 latent features obtained from PCA, AE, CAE, and deep SVDD in (**a**) Scenario 1 of the CWRU dataset (Case 1), (**b**) Scenario 1 of the PU dataset (Case 2), and (**c**) vibration signals of Rolling element experiment platform dataset (Case 3).

5. Conclusions

In this study, we present a fault signal detection framework based on deep SVDD for the implementation of an efficient CBM strategy on machinery systems. To this end, we utilize the raw time-series signal because it directly reveals the health status of the machinery system. To handle the raw time-series signal data, the deep SVDD model is trained by one-dimensional CAE as a pre-training procedure, and the encoding part of the CAE structure is used as the initial network structure of the deep SVDD model.

The pre-training procedure helps to more accurately detect the decision boundary and the fault signals because intrinsic properties of time-series signal data can be accommodated to the neural network structures of the encoding part. Then, in the fine-tuning procedure, the neural network structures for the deep SVDD model are updated to minimize the loss function for the anomaly detection tasks. Through the fine-tuning procedure, the latent features specialized to the fault signal detection can be generated, and it eventually helps the deep SVDD model to achieve superior fault signal detection performance. To demonstrate the efficacy of the deep SVDD model in fault signal detection, we used a widely used benchmark signal dataset (CWRU dataset) and both the vibration and acoustic signal datasets collected from our own rolling element experiment platform. In this experimental study with these datasets, the deep SVDD model outperforms other OCC methods, and these results confirm the applicability of the deep SVDD model in real fault signal detection problems.

In spite of its superiority in fault signal detection problems, the decision boundary of the deep SVDD model might be corrupted by the noisy or outlying time series signals, which are generated by incomplete sampling or data transmission error [20,56]. Thus, in further study, we will address the deep SVDD model's vulnerability against noisy or outlying time series signals. To this end, we will improve the deep SVDD model by incorporating the relative density of individual signals because these noisy or outlying time series signals tend to be located in sparse regions in feature space.

Author Contributions: D.Y. is responsible for the whole part of the paper (Conceptualization, methodology, formal analysis, validation, and writing—original draft preparation), and J.Y. is responsible for formal analysis, review and editing, supervision, and fund acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: The corresponding author (J.Y.) of this research was supported by the Incheon National University (International Cooperative) Research Grant in 2021.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to ongoing several other studies with the datasets presented in this study. If all of these studies are published, we will release the datasets presented in this study in our research team's website.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kumar, S.; Goyal, D.; Dang, R.K.; Dhami, S.S.; Pabla, B.S. Condition based maintenance of bearings and gears for fault detection–A review. *Mater. Today Proc.* **2018**, *5*, 6128–6137. [CrossRef]
2. Kim, J.; Ahn, Y.; Yeo, H. A comparative study of time-based maintenance and condition-based maintenance for optimal choice of maintenance policy. *Struct. Infrastruct. Eng.* **2016**, *12*, 1525–1536. [CrossRef]
3. Wu, S.; Zuo, M.J. Linear and nonlinear preventive maintenance models. *IEEE Trans. Reliab.* **2010**, *59*, 242–249.
4. Yang, S.K. A condition-based failure-prediction and processing-scheme for preventive maintenance. *IEEE Trans. Reliab.* **2003**, *52*, 373–383. [CrossRef]
5. Yang, B.S. An intelligent condition-based maintenance platform for rotating machinery. *Expert Syst. Appl.* **2012**, *39*, 2977–2988.
6. Yang, S.K.; Liu, T.S. A Petri net approach to early failure detection and isolation for preventive maintenance. *Qual. Reliab. Eng. Int.* **1998**, *14*, 319–330. [CrossRef]

7. Jardine, A.K.; Lin, D.; Banjevic, D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mech. Syst. Signal Process.* **2006**, *20*, 1483–1510. [CrossRef]
8. Gao, Z.; Cecati, C.; Ding, S.X. A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3757–3767. [CrossRef]
9. Goyal, D.; Pabla, B.S. Condition based maintenance of machine tools—A review. *CIRP J. Manuf. Sci. Technol.* **2015**, *10*, 24–35. [CrossRef]
10. Lee, J.; Ardakani, H.D.; Yang, S.; Bagheri, B. Industrial big data analytics and cyber-physical systems for future maintenance & service innovation. *Procedia CIRP* **2015**, *38*, 3–7.
11. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef] [PubMed]
12. Ruff, L.; Vandermeulen, R.; Goernitz, N.; Deecke, L.; Siddiqui, S.A.; Binder, A.; Müller, E.; Kloft, M. Deep One-Class Classification. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018.
13. Li, W.; Shang, Z.; Gao, M.; Qian, S.; Zhang, B.; Zhang, J. A novel deep autoencoder and hyperparametric adaptive learning for imbalance intelligent fault diagnosis of rotating machinery. *Eng. Appl. Artif. Intell.* **2021**, *102*, 104279. [CrossRef]
14. Shao, S.; Wang, P.; Yan, R. Generative adversarial networks for data augmentation in machine fault diagnosis. *Comput. Ind.* **2019**, *106*, 85–93. [CrossRef]
15. Baydar, N.; Ball, A. Detection of gear failures via vibration and acoustic signals using wavelet transform. *Mech. Syst. Signal Process.* **2003**, *17*, 787–804. [CrossRef]
16. Zhu, Y.; Li, G.; Tang, S.; Wang, R.; Su, H.; Wang, C. Acoustic signal-based fault detection of hydraulic piston pump using a particle swarm optimization enhancement CNN. *Appl. Acoust.* **2022**, *192*, 108718. [CrossRef]
17. Zhang, J.; Sun, Y.; Guo, L.; Gao, H.; Hong, X.; Song, H. A new bearing fault diagnosis method based on modified convolutional neural networks. *Chin. J. Aeronaut.* **2020**, *33*, 439–447. [CrossRef]
18. Krawczyk, B.; Woźniak, M.; Cyganek, B. Clustering-based ensembles for one-class classification. *Inf. Sci.* **2014**, *264*, 182–195. [CrossRef]
19. Yu, J.; Kang, J. Clustering ensemble-based novelty score for outlier detection. *Eng. Appl. Artif. Intell.* **2023**, *121*, 106164. [CrossRef]
20. Yu, J.; Do, H. Proximity-based density description with regularized reconstruction algorithm for anomaly detection. *Inf. Sci.* **2024**, *654*, 119816. [CrossRef]
21. Lei, Y.; He, Z.; Zi, Y. A new approach to intelligent fault diagnosis of rotating machinery. *Expert Syst. Appl.* **2008**, *35*, 1593–1600. [CrossRef]
22. Lei, Y.; Zuo, M.J. Gear crack level identification based on weighted K nearest neighbor classification algorithm. *Mech. Syst. Signal Process.* **2009**, *23*, 1535–1547. [CrossRef]
23. Shen, Z.; Chen, X.; Zhang, X.; He, Z. A novel intelligent gear fault diagnosis model based on EMD and multi-class TSVM. *Measurement* **2012**, *45*, 30–40. [CrossRef]
24. Abdi, H.; Williams, L.J. Principal component analysis. *WIREs Comp. Stat.* **2010**, *2*, 433–459. [CrossRef]
25. Schölkopf, B.; Smola, A.; Müller, K.R. Kernel principal component analysis. In Proceedings of the International Conference on Artificial Neural Networks, Berlin, Germany, 8–10 October 1997.
26. Lu, W.; Wang, X.; Yang, C.; Zhang, T. A novel feature extraction method using deep neural network for rolling bearing fault diagnosis. In Proceedings of the 27th Chinese Control and Decision Conference, Qingdao, China, 23–25 May 2015.
27. Zhang, Y.; Zhou, T.; Huang, X.; Cao, L.; Zhou, Q. Fault diagnosis of rotating machinery based on recurrent neural networks. *Measurement* **2021**, *171*, 108774. [CrossRef]
28. Hu, Z.; Zhao, H.; Peng, J. Low-rank reconstruction-based autoencoder for robust fault detection. *Control Eng. Pract.* **2022**, *123*, 105156. [CrossRef]
29. Chalapathy, R.; Menon, A.K.; Chawla, S. Anomaly detection using one-class neural networks. *arXiv* **2019**, arXiv:1802.06360.
30. Tax, D.M.; Duin, R.P. Support vector data description. *Mach. Learn.* **2004**, *54*, 45–66. [CrossRef]
31. Schölkopf, B.; Platt, J.C.; Shawe-Taylor, J.; Smola, A.J.; Williamson, R.C. Estimating the support of a high-dimensional distribution. *Neural Comput.* **2001**, *13*, 1443–1471. [CrossRef]
32. Gröchenig, K. *Foundations of Time-Frequency Analysis*; Birkhäuser: Boston, MA, USA, 2001.
33. Sejdić, E.; Djurović, I.; Jiang, J. Time–frequency feature representation using energy concentration: An overview of recent advances. *Digital Signal Process.* **2009**, *19*, 153–183. [CrossRef]
34. Ogden, R.T. *Essential Wavelets for Statistical Applications and Data Analysis*; Birkhäuser: Boston, MA, USA, 1997.
35. Chen, G.; Zhang, X.; Wang, Z.J.; Li, F. Robust support vector data description for outlier detection with noise or uncertain data. *Knowl.-Based Syst.* **2015**, *90*, 129–137. [CrossRef]
36. Ghafoori, C.Z.; Erfani, S.M.; Rajasegarar, S.; Bezdek, J.C.; Karunasekera, S.; Leckie, C. Efficient unsupervised parameter estimation for one-class support vector machines. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 5057–5070. [CrossRef]
37. Yu, J.; Kang, S. Clustering-based proxy measure for optimizing one-class classifiers. *Pattern Recognit. Lett.* **2019**, *117*, 37–44. [CrossRef]
38. Parzen, E. On estimation of a probability density function and mode. *Ann. Math. Stat.* **1962**, *33*, 1065–1076. [CrossRef]
39. Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation forest. In Proceedings of the IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008.

40. Chen, L.; Xu, G.; Zhang, S.; Yan, W.; Wu, Q. Health indicator construction of machinery based on end-to-end trainable convolution recurrent neural networks. *J. Manuf. Syst.* **2020**, *54*, 1–11. [CrossRef]
41. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014.
42. Aggarwal, C.C. *Neural Networks and Deep Learning*; Springer: New York, NY, USA, 2018.
43. Masci, J.; Meier, U.; Cireşan, D.; Schmidhuber, J. Stacked convolutional auto-encoders for hierarchical feature extraction. In Proceedings of the International Conference on Artificial Neural Networks, Berlin, Germany, 14–17 June 2011.
44. Schlegl, T.; Seeböck, P.; Waldstein, S.M.; Schmidt-Erfurth, U.; Langs, G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In Proceedings of the International Conference on Information Processing in Medical Imaging, Boone, NC, USA, 25–30 June 2017.
45. Luo, W.; Liu, W.; Lian, D.; Tang, J.; Duan, L.; Peng, X.; Gao, S. Video anomaly detection with sparse coding inspired deep neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1070–1084. [CrossRef]
46. Bergman, L.; Hoshen, Y. Classification-based anomaly detection for general data. *arXiv* **2020**, arXiv:2005.02359.
47. Tack, J.; Mo, S.; Jeong, J.; Shin, J. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In Proceedings of the Advances in Neural Information Processing Systems, Virtual Conference, 6–12 December 2020.
48. Zong, B.; Song, Q.; Min, M.R.; Cheng, W.; Lumezanu, C.; Cho, D.; Chen, H. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
49. Metz, L.; Poole, B.; Pfau, D.; Sohl-Dickstein, J. Unrolled generative adversarial networks. *arXiv* **2016**, arXiv:1611.02163.
50. Ruff, L.; Vandermeulen, R.A.; Görnitz, N.; Binder, A.; Müller, E.; Müller, K.R.; Kloft, M. Deep semi-supervised anomaly detection. *arXiv* **2019**, arXiv:1906.02694.
51. Li, D.; Zhang, J.; Zhang, Q.; Wei, X. Classification of ECG signals based on 1D convolution neural network. In Proceedings of the IEEE International Conference on e-Health Networking, Applications and Services, Dalian, China, 12–15 October 2017.
52. Yu, J.; Zhou, X. One-dimensional residual convolutional autoencoder based feature learning for gearbox fault diagnosis. *IEEE Trans. Ind. Inf.* **2020**, *16*, 6347–6358. [CrossRef]
53. Lessmeier, C.; Kimotho, J.K.; Zimmer, D.; Sextro, W. Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In Proceedings of the PHM Society European Conference, Bilbao, Spain, 5–8 July 2016.
54. Liang, P.; Wang, W.; Yuan, X.; Liu, S.; Zhang, L.; Cheng, Y. Intelligent fault diagnosis of rolling bearing based on wavelet transform and improved ResNet under noisy labels and environment. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105269. [CrossRef]
55. Sun, J.; Liu, Z.; Wen, J.; Fu, R. Multiple hierarchical compression for deep neural network toward intelligent bearing fault diagnosis. *Eng. Appl. Artif. Intell.* **2022**, *116*, 105498. [CrossRef]
56. Liu, B.; Xiao, Y.; Philip, S.Y.; Hao, Z.; Cao, L. An efficient approach for outlier detection with imperfect data labels. *IEEE Trans. Knowl. Data Eng.* **2013**, *26*, 1602–1616. [CrossRef]
57. Conover, W.J. *Practical Nonparametric Statistics*, 3rd ed.; John Wiley & Sons: Hoboken, NJ, USA, 1999.
58. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn Res.* **2008**, *9*, 2579–2605.

applied sciences

MDPI

Article

Delamination Detection Framework for the Imbalanced Dataset in Laminated Composite Using Wasserstein Generative Adversarial Network-Based Data Augmentation

Sungjun Kim [1], Muhammad Muzammil Azad [2], Jinwoo Song [2,*] and Heungsoo Kim [2,*]

1 Smart Materials and Design Laboratory (SMD LAB), Department of Mechanical Engineering, Dongguk University-Seoul, 30 Pildong-ro 1-gil, Jung-gu, Seoul 04620, Republic of Korea; sjkim_400@dgu.ac.kr

2 Department of Mechanical, Robotics, and Energy Engineering, Dongguk University-Seoul, 30 Pildong-ro 1-gil, Jung-gu, Seoul 04620, Republic of Korea; muzammilazad@dgu.ac.kr

* Correspondence: jwsong0620@dgu.edu (J.S.); heungsoo@dgu.edu (H.K.); Tel.: +82-2-2260-8577 (H.K.); Fax: +82-2-2263-9379 (H.K.)

Abstract: As laminated composites are applied more commonly, Prognostics and Health Management (PHM) techniques for the maintenance of composite systems are also attracting attention. However, applying PHM techniques to a composite system is challenging due to the data imbalance problem from the lack of failure data and unpredictable failure cases. Despite numerous studies conducted to address this limitation, including techniques like data augmentation and transfer learning, significant challenges remain. In this study, the Wasserstein Generative Adversarial Network (WGAN) model using a time-series data augmentation technique is proposed as a solution to the data imbalance problem. To ensure the performance of the WGAN model, time-series data augmentation of experimental data is executed with a frequency analysis. After that, a One-Dimensional Convolutional Neural Network (1D CNN) is used for fault diagnosis in laminated composites, validating the performance improvement after data augmentation. The proposed data augmentation significantly elevated the performance of the 1D CNN classification model compared to its non-augmented counterpart. Specifically, the accuracy increased from 89.20% to 91.96%. The precision improved remarkably from 29.76% to 74.10%, and its sensitivity rose from 33.33% to 94.39%. Collectively, these enhancements highlight the vital role of data augmentation in improving fault diagnosis performance.

Keywords: PHM; fault diagnosis; data imbalance; laminated composite; WGAN

1. Introduction

With their high specific strength, stiffness, and resistance to both corrosion and heat, composite structures are increasingly replacing metallic structures in various engineering applications, including aerospace, marine, automobile, and infrastructure [1,2]. Composite structure has orthotropic, layered structure and complexity in the manufacturing process. Due to these natural characteristics, various fault modes, such as matrix crack, fiber breakage, and delamination, often occur, and this makes it difficult to apply composite structures in the actual industrial field [3,4]. Among these various failure modes, delamination or inter-ply separation is the most critical defect. These types of failure exist in the inner surface and are not observable without technical equipment, leading to sudden fracture of the structure and a huge loss of various resources [5,6]. To avoid structural failure and severe loss, it is necessary to quickly detect damage in composite structures to prolong their service life. PHM technology provides early damage detection and helps avoid the deterioration of various industrial systems [7–12]. Recently, techniques for PHM based on Machine Learning (ML) and Deep Learning (DL) using vibration signals have been continuously adopted for fault diagnosis in various structures. They have the ability to detect unseen defects in the interior of the structure, and using the vibration data demonstrates high fault diagnosis performance [13–16].

Initially, conventional Machine Learning (ML) model-based fault diagnosis was mainly used, in which engineers manually extracted features related to the faults of the system and diagnosed the condition of the system [17]. Features extracted from signals are generally from the time domain, frequency domain, and time–frequency domain [18]. Time domain features vary from conventional values, for example, the mean, standard deviation, peak, amplitude square, and Root Mean Square (RMS), to combinations of these features, like the waveform factor, peak factor, margin factor, and central moment [19,20]. For more detailed information, frequency, as well as time–frequency domain features, could also be derived from the Fourier transform, short-time Fourier transform, wavelet transform, and wavelet packet transform [21–23]. Because features related to frequency analysis contain the dynamic behavior of the system, they can offer more sensitive and thorough information about the system. The extracted features are then utilized to predict whether the system is in a normal or fault state. ML models, such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Decision Tree, are trained by part of the extracted features, while the rest of the features are exploited to test whether the trained model can properly decide the health state of the system [24–26].

Although fault diagnosis using ML showed remarkable performance, it was time consuming, and a massive theoretical background was needed to extract features manually from the data. To overcome this limitation, Deep Learning (DL) techniques started to be utilized for fault diagnosis. The DL model can extract features automatically by using its neural network and classify the health state of the system. The Convolutional Neural Network (CNN) model is generally used for fault diagnosis owing to its outstanding feature extraction performance from motor current signals, vibration signals, and multi-variate signals [27–30]. Raw signals are converted into images that can represent the health state of the system, like spectrogram, scalogram, or various sorts of grey-scale images. The CNN model extracts features from these images and then predicts whether the system is in a normal state. The Vision Transformer (ViT) model is also utilized for fault diagnosis using image data [31]. The ViT model can extract features from the local part of an image and classify the health state of the system. Transfer learning models that are pretrained on a large amount of image datasets are widely applied for fault diagnosis. Among the various models, Residual Network-50 (Resnet-50) is generally used [32]. Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) are other deep learning-based methods for fault diagnosis. They have a unique ability to extract features from time series data; they have remarkable performance in predicting Remaining Useful Life (RUL), as well as fault diagnosis. The models use extracted features from the raw time-series signal itself or preprocessed signals to perform fault diagnosis [33–36].

Although DL-based fault diagnosis has been explored by many researchers, numerous limitations still remain. For general applications of fault diagnosis using DL models, further research is required to address challenges, such as noise mitigation, data imbalance, and various uncertainties. In particular, data imbalance, where certain classes in the dataset have fewer instances, can significantly reduce the performance of the DL model. The model may struggle to learn features of the minority class, resulting in lower accuracy for that class [37]. In addition, since there is a myriad of unpredictable damage cases with a diverse range of operating and environmental conditions, it is impractical to comprehensively detect and anticipate all potential damages in advance. To cope with these challenges, it is essential to generate synthetic data through experimentation, simulation, or data-driven approaches. However, the process of data acquisition through experiments or simulations can be both time-consuming and financially burdensome. Additionally, these methods require an in-depth understanding of the physics of the composite structure [38]. To address these challenges, a focus has been placed on developing data-driven approaches to generate synthetic data. These methods require less in-depth understanding of the physics involved and are less labor-intensive by leveraging advanced algorithms and DL techniques.

Above all, there are some basic data augmentation techniques that use simple mathematical calculations. For example, images from impact response data can be manipulated by adding and multiplying by random numbers [39]. Similarly, adding Gaussian noise to the images obtained from acoustic emission and guided wave signals was also performed [40,41]. The advantage of this type of data augmentation method is the low level of difficulty. They are easy to implement, and the corresponding algorithms are not complex. Other simple methods—such as shifting, scaling, rotating, and random grid shuffling of the images—are also performed [42]. These are also basic techniques for data augmentation for images. However, in most engineering problems, images contain some dynamic characteristics of the system, including time, amplitude, and frequency of the signal. Therefore, this kind of augmentation technique should be applied with caution regarding the engineering problem. Recently, data augmentation using the GAN model for image data has been widely used [43–45]. By using the GAN model, it is possible to duplicate the dynamic characteristics of the original data and thus generate synthetic image data properly.

Although these approaches can generate sufficient data for robust fault diagnosis in composite structures, most of them rely on image-based data augmentation. Thus, this approach necessitates proper preprocessing of raw data for their transformation into 2D images, making the process tedious and time-consuming. Therefore, data augmentation of the raw time-series signal, not imagery, can benefit by eliminating the need for excessive preprocessing, making the PHM process for composite structures simple and computationally efficient.

To tackle the issues mentioned above, this paper proposes a time-series data augmentation technique using the Wasserstein Generative Adversarial Network (WGAN) model. WGAN is a type of DL model that can accurately capture the distribution of the training data. Thus, it can generate synthetic data that maintain identical dynamic characteristics with the training data. For this reason, an in-depth understanding of the physics of the composite structures is less necessary to use the WGAN model to generate the synthetic data. As a tool to validate the proposed data augmentation method, fault diagnosis in laminated composite structure using the One-Dimensional Convolutional Neural Network (1D CNN) model was implemented. The 1D CNN can extract features directly from time-series signals, and no preprocessing is needed for the use of input data. Therefore, the total time for both preprocessing and training is reduced.

To validate the proposed data augmentation methodology, an initial experiment was conducted to obtain vibration data from composite specimens. Three different health states of composite specimens—healthy, delamination 1, and delamination 2—were manufactured. Vibration signals from these states were then captured using a shaker and computer software. The amount of data collected for the healthy state was intentionally made much larger than the damaged ones to simulate the data imbalance problem. Then, data augmentation was performed across all health states. After data augmentation, a fault diagnosis was conducted using the 1D CNN model on two different datasets: a dataset consisting of only experimental data and a dataset composed of synthetic data. Then, the two results were compared to assess the data augmentation capabilities of the WGAN model. Afterward, a comparative analysis was performed to highlight the superior data augmentation ability of the WGAN model by comparing it with various oversampling and other data augmentation methods.

The rest of this paper is composed as follows: Section 2 presents the background of previous methods for data augmentation and the fault diagnosis model for this research. Section 3 illustrates the experimental setup to validate the suggested methodology. Section 4 describes the fault diagnosis results and provides a comparative analysis. Finally, Section 5 concludes this research, outlining its contributions and suggesting directions for future research.

2. Proposed Methodology

2.1. Overview

In this section, solutions for the data imbalance problem and fault diagnosis methodology are proposed. Section 2.2 briefly introduces the theoretical background of previous methods for oversampling and data augmentation, which is later utilized in Section 4. Then, Section 2.3 describes the mathematical formula of the WGAN model for time-series vibration data augmentation. Finally, Section 2.4 explains the fundamental contents of the 1D CNN model for fault diagnosis.

2.2. Previous Works

2.2.1. Adaptive Synthetic Sampling

Although several techniques have been developed to address data imbalance, methods that directly augment the minority class by generating synthetic data have gained significant attention. Adaptive Synthetic Sampling (ADASYN) is an oversampling technique based on the K-Nearest Neighbor (KNN) algorithm, considering the distribution of the minority and majority classes. ADASYN generates synthetic data by interpolating between existing data points within the minority class rather than merely duplicating the original data [46]. In the minority class, the basis data point is decided as a fixed point to apply the KNN algorithm, and the nearest neighbor value is selected. After the nearest neighbors are chosen, the linear interpolation method of Equation (1) is used to generate synthetic data between the basis data point and the nearest neighbors, where α is a random value from 0 to 1.

$$s_j = (1 - \alpha) \times p_i + p_{ij},\ i, j = 1, 2, 3 \ldots \quad (1)$$

Here, data points in the majority class, which exist in the inner area of the nearest neighbor boundary, are considered. The ratio of data points within the majority class is calculated according to Equation (2), where δ_i is the number of data points in the majority class in the nearest neighbor boundary for the i-th nearest neighbor, and K is the total number of nearest neighbors.

$$r_i = \frac{\delta_i}{K},\ i = 1, 2, 3 \ldots \quad (2)$$

ADASYN generates more synthetic data if r_i is larger than the others. Owing to this comparison process of data distribution, synthetic data could be distributed to maintain distinct characteristics compared to the data from other classes.

2.2.2. System Identification

While ADASYN focuses on generating synthetic data for the minority class based on the distribution of classes and neighboring data points, another approach to tackle data imbalance is through System Identification (SI). The SI process builds a mathematical dynamic model of the system by using the measured input and output signals. The SI technique is an important part of modern control theory [47] and has been used as a data augmentation to overcome data imbalance and the data scarcity problem [48]. The input and output signals of the system are used to reconstruct the discrete system matrix A_r, B_r, C_r, D_r, which is expressed as a state space model, where A_r is the realized system matrix, B_r is the realized input matrix, C_r is the realized output matrix, and D_r is the transmission matrix [49]. After these matrices are defined, unseen signals are used as input for the realized matrices. As a result, synthetic signals, which are augmented signals, could be obtained. The SI technique can restore the system dynamics by using only single input and output signals, which means that it does not need multiple pairs of signals. Because of this advantage, it could be used as a great solution for data imbalance and the scarcity problem, as mentioned above.

2.2.3. Generative Adversarial Network

The GAN model, originally proposed to produce real-like images from Gaussian noise input, has also been developed to generate synthetic time-series data [50–53]. The GAN model is composed of two deep learning models: generator and discriminator. The generator generates synthetic data by using Gaussian noise input. The discriminator gets real data and synthetic data as input data and discriminates whether the input data are real or synthetic. By using the training results of these models, parameters are updated to produce realistic synthetic data, which means that the synthetic data distribution p_z has a similar distribution to the real data $p_{data(x)}$. The loss function of the GAN model is written as Equation (3), which is the deformed shape of the Jensen–Shannon Divergence (JSD), where $D(x)$ is the discriminator output, and $G(z)$ is the generator output.

$$\max\min V(D,G) = E_{x \sim p_{data(x)}}[\log D(x)] + E_{z \sim p_{z(z)}}[\log(1 - D(G(z)))] \tag{3}$$

The generator and discriminator keep training to deceive each other until the models reach the Nash Equilibrium. Figure 1 illustrates the training process of the general GAN model.

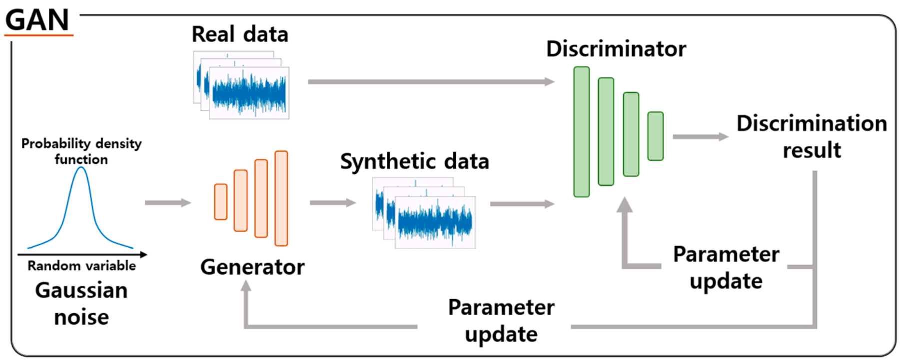


Figure 1. Training process of the GAN model.

Although the GAN model shows remarkable data augmentation performance, its mathematical nature raises several critical points. There are two main limitations to the application of the GAN model: gradient vanishing and mode collapse [54,55]. Both problems arise from the loss function of the GAN model. The fundamental concept of the GAN model is a minimax game in which the discriminator tries to maximize the loss function while the generator attempts to minimize the loss function. In this process, if the discriminator is trained better than the generator, the gradient of the generator becomes 0 according to Equation (3), and the generator cannot be trained properly. This is called gradient vanishing. Also, when the generator generates synthetic data in multiple classes, the generator only tries to deceive the discriminator and does not consider the information about each class. As a result of this characteristic, the training process of the generator could be biased to specific weights, which means the generator only generates data belonging to a specific class, not evenly. Simply, the generator falls into a local minimum problem and cannot imitate the distribution in any other classes except one class. This is called mode collapse.

Due to the drawbacks mentioned above, instability in training the GAN model often takes place. This makes it hard to converge the generator and discriminator, reducing the performance of the GAN model. Numerous unrevealed factors may exist in this instability, but one of the fundamental reasons is the loss function. The loss function that uses JSD could output 0 or too large a number, as mentioned above, leading to the submission of meaningless gradients to the generator and discriminator. To overcome these vulnerabilities, various GAN models that use different loss functions have been introduced.

2.3. Wasserstein Generative Adversarial Network

As outlined in Section 2.2.3, the limitations for GAN arise from the generator receiving meaningless gradients. To provide the generator with meaningful gradients, employing an alternative loss function is essential. WGAN uses Earth Mover's Distance (EMD) or Wasserstein distance as loss functions, which can be expressed as Equation (4).

$$\mathrm{W}(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|] \tag{4}$$

Here, $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ represents all sets of the joint distributions $\gamma(x, y)$, which have $\mathbb{P}_r$, $\mathbb{P}_g$ as marginal individually. The intuitive definition of the Wasserstein distance means the quantity of mass that must be transported from x to y for the distribution of the generated data $\mathbb{P}_g$ to align with the distribution of real data $\mathbb{P}_r$.

However, it is impossible to apply the Wasserstein distance in Equation (4) directly as a loss function because it is almost impossible to find all the joint distributions and minimum values. Due to this restriction, Kantorovich–Rubinstein duality is introduced, which constrains the objective function by using the 1–Lipshitz method [56]. In this context, since it is difficult to identify the exact function we want to constrain, the objective function is approximated using a neural network. After applying the Kantorovich–Rubinstein duality and 1–Lipshitz method, the loss function using Wasserstein distance turns into Equation (5).

$$\text{Loss function} = \max_{w\in W} \mathbb{E}_{x\sim\mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z\sim p(z)}[f_w(g_\theta(z))] \tag{5}$$

Here, f_w denotes a function approximated by the neural network that has variables of w, and similarly, g_θ is a function that involves variables of θ. For brevity of the study, detailed mathematical formulae of the Kantorovich–Rubinstein duality and Lipshitz method are omitted.

The gradients for updating WGAN must be restricted because it does not use the sigmoid function for the last layer of the discriminator. This could potentially lead to unbounded output values and destabilizing gradients during the training process. For this purpose, the 'weight clipping' technique is used to restrict the scale of the gradients between -0.01 and 0.01. Because of this mathematical difference between the discriminator in the conventional GAN and the discriminator in WGAN, the latter one is named the 'critic'. Through this overall process in WGAN, the generator can obtain proper gradients, and the training generator and critic can be balanced. Furthermore, through generating meaningful gradients, the stability of the training process can be improved.

2.4. One-Dimensional Convolutional Neural Network

Over recent years, there has been significant research interest in the Two-Dimensional Convolutional Neural Network (2D CNN) model-based fault diagnosis because it has a remarkable ability to automatically extract features from the images [57,58]. However, 2D CNN requires various data preprocessing techniques, especially when used with vibration signals. For this application, the signal has to be converted into spectral images, making the fault diagnosis procedure more complex and laborious. Also, 2D CNN takes a longer time for training and testing, which may not be suitable for real-time fault diagnosis. For these reasons, this research employs the One-Dimensional Convolutional Neural Network (1D CNN), using time-series vibration signals directly as input for fault diagnosis.

The 1D CNN can directly extract features from time-series signals without data preprocessing. Contrary to 2D CNN, convolution and pooling size can extract one-dimensional features from a time-series signal. Also, the training speed of the 1D CNN is much quicker than that of the 2D CNN because the input data size for the 1D CNN is smaller, and 1D CNN does not need additional preprocessing techniques since time-series data are directly used as input. This makes the 1D CNN more resource-efficient, leading to cost savings without decreasing the fault diagnosis performance. Owing to these advantages, 1D CNN is continuously applied for fault diagnosis in various systems [59,60]. Details of the structures of the 1D CNN used in this research are later discussed in Section 4.2.

3. Experimental Validation

Section 3 describes the experimental validation process to ensure the performance of the suggested data augmentation and fault diagnosis techniques. Section 3.1 explains the experimental setup to obtain the imbalanced vibration dataset, while Section 3.2 describes the data augmentation process using the WGAN model to resolve the data imbalance problem and the validation of the synthetic data involved.

3.1. Experimental Setup and Data Acquisition

Laminated composite plates are manufactured to obtain an imbalanced vibration dataset. Eight layers of carbon prepreg (T700SC–12k–60E), which have dimensions of 35 cm × 30 cm, are stacked in $[0/90/0/90]_s$ order, as shown in Figure 2. Table 1 shows the mechanical properties of the prepreg.

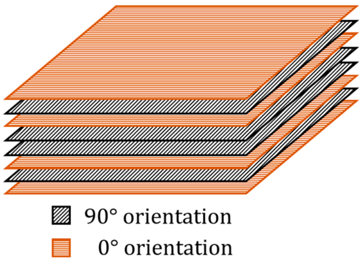


Figure 2. Laminated angles for each layer in the composite structure.

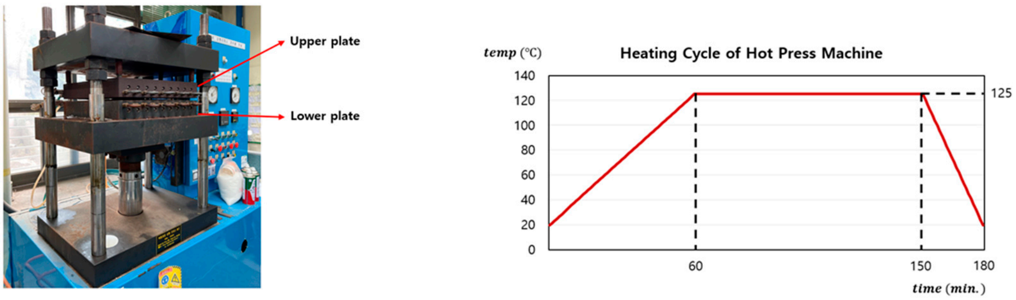
Table 1. Mechanical properties of the carbon prepreg.

Tensile Modulus	Tensile Strength	Elongation	Thermal Conductivity	Density	Filament Diameter
230 GPa	4900 MPa	2.1%	9.4 W/m·K	1.8 g/cm^3	7 µm

To imitate various health states of the laminated composites, Teflon film (Model KSC−V1000) with a thickness of 0.03 mm and a usable heating range of 280 °C was inserted between the fourth and fifth layers during the stacking. The film can act like damage inside the laminated composite, so the presence of delamination could be imitated. A total of three health states of composite plates are stacked: Healthy (H), which has no delamination, and Delamination 1 (D1) and Delamination 2 (D2), which have delamination in different locations inside of the specimens.

After stacking, the plates were cured using the hot press machine illustrated in Figure 3a. The pressure of the machine was 20 kg/cm^2, and the heating cycle of the plate is in Figure 3b. The cured plates that have dimensions of 35 cm × 30 cm are then cut into five pieces to receive a beam-shape of 35 cm × 5 cm, as shown in Figure 4. Thus, five specimens for each health state, and a total of fifteen specimens, were manufactured for all health states. The blue-colored area is used as the fixed part for the vibration experiment, while the red-colored area indicates the presence of delamination.

Figure 5 shows the Data Acquisition system. Random vibration signals were generated through MATLAB Simulink, and they were collected in the Data Acquisition (DAQ) unit (DAQ1, model dSPACE/CLP1104). The magnitude of signals was enlarged through the amplifier (model Labworks/PA-151). These amplified signals were then received by a shaker (model Labworks/ET-126-4), which induced vibration in the specimen.



(**a**) (**b**)

Figure 3. (**a**) Hot press machine used for manufacturing specimen. (**b**) Heating cycle for the hot press machine.

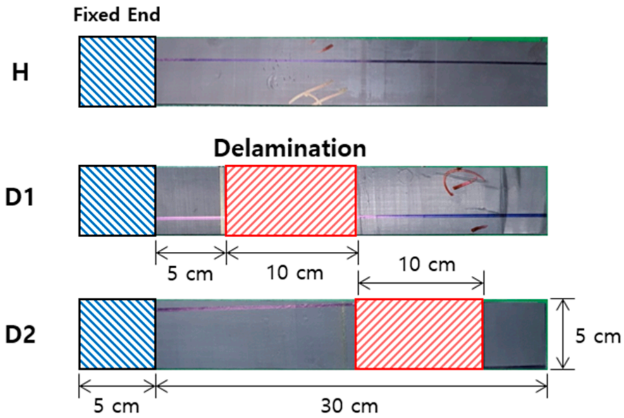


Figure 4. Three health states of the manufactured specimens.

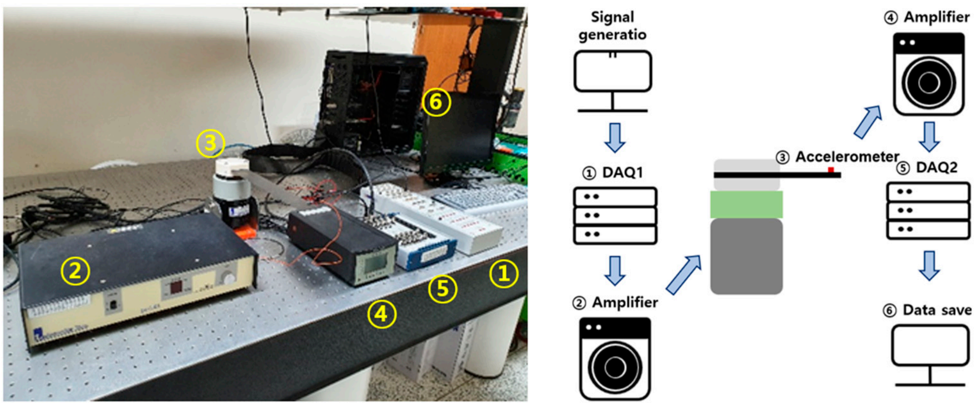


Figure 5. Data acquisition system for the experiment.

Response signals were obtained with accelerometer (model Bruel & Kjaer/Type 4517-C) sensors that were bonded on the top surface of the specimens. Herein, for the robustness of variations in signals, signals were acquired through the same ten sensors but in different locations. Figure 6 shows the locations of the accelerometers. The figure shows that the sensors were bonded near the edges to consider the effects of twisting and bending in the specimen.

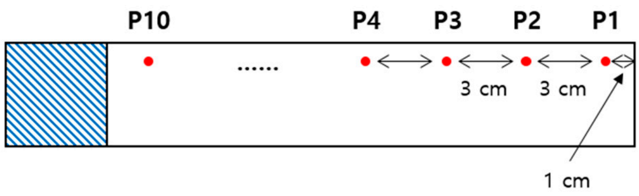


Figure 6. The ten different accelerometer locations for the research.

The response signals from the sensors were then put into the amplifier (model Bruel & Kjaer/Type 2692-0s2) again, and after passing the DAQ (DAQ2, model NI/USB-6341), the final response vibration signals were saved in the computer.

To simulate the imbalance problem in the experimental condition, a different number of sensors were selected for each health state. For state H, sensors from P1 to P10 were used, while for states D1 and D2, only P01 was used. For state H, 1000 signals were obtained for each sensor, and 60 signals for D1 and D2 were acquired to maximize the data imbalance problem. Because five specimens for each health state were utilized to obtain the dataset, 200 data instances for H and 12 data instances for D1 and D2 each were acquired from each specimen. Table 2 shows the number of the dataset and length of the signals for each health state.

Table 2. Description of dataset for each health state.

	Healthy	Delamination 1	Delamination 2
Number of sensors	P01–P10	P01	P01
Number of data	1000	60	60
Signal length	1875	1875	1875

3.2. Data Augmentation Using WGAN Model

To validate the proposed method, data augmentation was performed using the WGAN model illustrated in Figure 7. The structure of the WGAN model and model parameters listed in Table 3 are determined through a process of trial and error because there is no specific standard or technique.

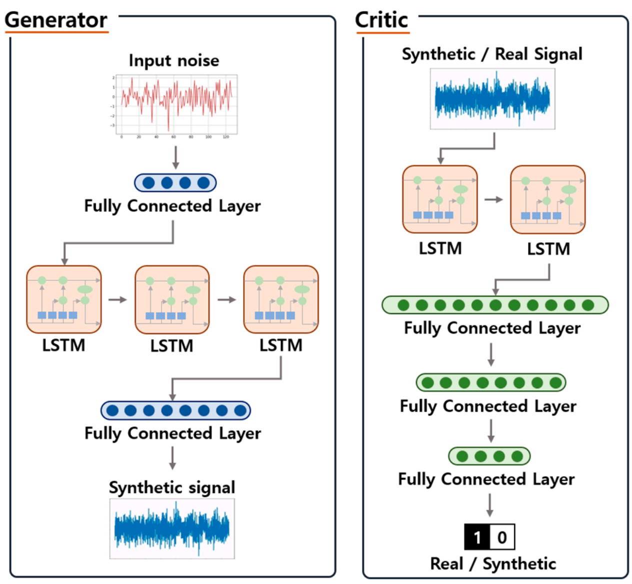


Figure 7. Structures of the generator and critic for the research.

Table 3. Model parameters of the WGAN model for each health state.

	Healthy	Delamination 1	Delamination 2
Beta 1	0.5	0.5	0.5
Beta 2	0.9	0.9	0.9
Training critic/Training generator	3	5	5
Learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4}
Epoch	2000	2000	2000
Batch size	8	8	8

For the Healthy state, since it takes a significant period to generate synthetic signals for all pairs, signals from sensors in the same location have similar dynamic characteristics. Thus, synthetic signals were generated by only considering sensor location. With data augmentation, with 300 signals per sensor, a total of 3000 signals were generated. For Delamination 1 state, three specimens, D1-1, D1-2, and D1-3, were utilized individually to obtain 1000 synthetic signals per specimen. Similarly, Delamination 2 state used three specimens and acquired 1000 signals per specimen. For delamination cases, delamination in the specimens was manufactured by manually inserting the Teflon films. Therefore, there are slight differences in dynamic characteristics for different specimens. In this perspective, data augmentation for delamination cases was conducted for each specimen. There were 1000 synthetic signals for each specimen; a total of 3000 signals were generated.

4. Fault Diagnosis Results

4.1. Evaluation Metrics

Evaluation metrics are employed to evaluate the classification performance of a DL model. These metrics are calculated by using the predicted labels from the DL model and comparing them with the actual labels. Figure 8 shows the relationships and denotations of these metrics, while Equations (6)–(9) provide the formulae for these evaluation metrics. Accuracy is an intuitive metric with which to evaluate the classification performance; it quantifies the proportion of predicted labels that match the actual labels, including both true and false predictions. Precision is the ratio of actual true labels to the labels that the model predicted as true. Sensitivity is the ratio of the labels that the model predicted as true to the actual true labels. The F1 score is the harmonic average of the precision and sensitivity. All metrics reinforce the weakness of others and thus can evaluate the classification performance of the model.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \tag{6}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \tag{9}$$

		Actual label	
		True	False
Predicted label	True	True Positive (TP)	False Positive (FP)
	False	False Negative (FN)	True Negative (TN)

Figure 8. Denotations of the predicted and actual labels.

4.2. Fault Diagnosis Results

In this section, fault diagnosis was performed on the experimental data and synthetic data to evaluate the effectiveness of data augmentation. Figure 9 shows the 1D CNN model structure that was used for fault diagnosis, and Table 4 indicates the detailed information of the model. To maintain the uniformity of the research, the same 1D CNN model structure was applied in all instances of fault diagnosis. Considering complex embedded information in the random signal, the model is constructed with several Convulotional–Maxpooling pairs. To activate the nonlinearities from convolution layers, the ReLU activation function was applied. After extracting features, the dropout layer was added after the dense layer to reduce overfitting due to its complex model structure.

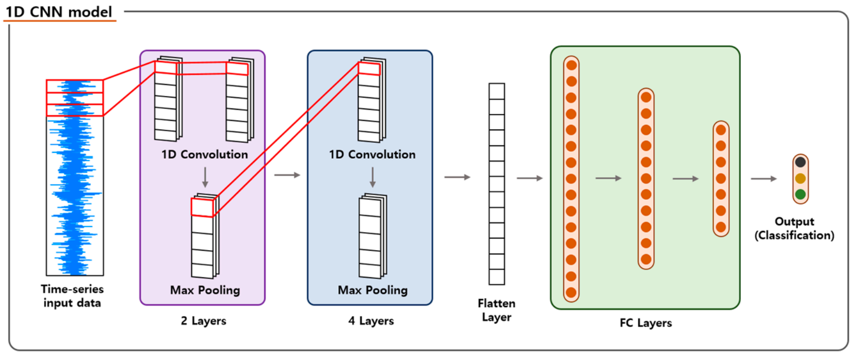


Figure 9. Structure of the 1D CNN model for the research.

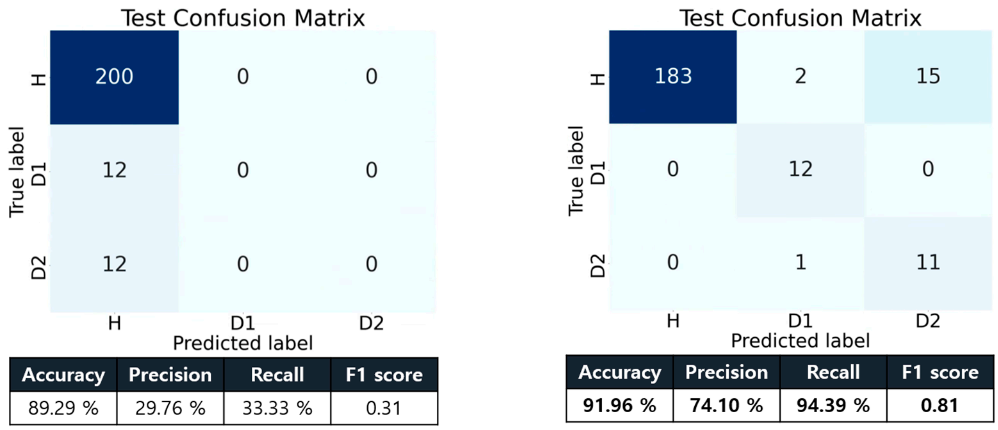
Table 4. Detailed information of 1D CNN model for the research.

Network Layer	Output Data Size	Parameters
Input Layer	1875 × 1	1875 length signal input
Conv1D	1875 × 32	32 @ 3 × 1, stride = 1, activation = ReLU
Conv1D	1875 × 32	32 @ 3 × 1, stride = 1, activation = ReLU
Max pooling 1D	937 × 32	2 × 1, stride = 2, activation = ReLU
Conv1D	937 × 64	64 @ 3 × 1, stride = 1, activation = ReLU
Conv1D	937 × 64	64 @ 3 × 1, stride = 1, activation = ReLU
Max pooling 1D	468 × 64	2 × 1, stride = 2, activation = ReLU
Conv1D	468 × 128	128 @ 2 × 1, stride = 1, activation = ReLU
Max pooling 1D	234 × 128	2 × 1, stride = 2, activation = ReLU
Conv1D	234 × 128	128 @ 2 × 1, stride = 1, activation = ReLU
Max pooling 1D	117 × 128	2 × 1, stride = 2, activation = ReLU
Conv1D	117 × 256	256 @ 2 × 1, stride = 1, activation = ReLU
Max pooling 1D	58 × 256	2 × 1, stride = 2, activation = ReLU
Conv1D	58 × 256	256 @ 2 × 1, stride = 1, activation = ReLU
Max pooling 1D	29 × 256	2 × 1, stride = 2, activation = ReLU
Flatten Layer	1 × 7424	7424 neurons
Input Layer	1 × 7424	7424 neurons
Dense	1 × 1024	1024 neurons
Dropout	1 × 1024	Dropout rate: 0.4
Dense	1 × 512	512 neurons
Dropout	1 × 512	Dropout rate: 0.4
Dense	1 × 128	128 neurons
Dropout	1 × 128	Dropout rate: 0.4
Dense	1 × 3	128 neurons
SoftMax	1 × 3	Classification Layer

For the 1120 signals in the experiment, they were randomly divided into 60%, 20%, and 20% for training, validation, and testing of the model while maintaining the proportion of the data for each class. After generating synthetic signals with the WGAN model, as written in Section 3.2, they were used for training and validation data with the same 1D

CNN model structure. For each health state, 3000 signals were generated, 9000 in total. Then, synthetic signals were randomly divided into a 60% allocation for training data and a 40% allocation for validation data while maintaining the class proportion. To demonstrate the enhanced fault diagnosis performance, the test dataset for the experimental data case was used for both cases. Because the data were divided randomly, test data may contain some identical data used for data augmentation. However, this fault diagnosis result could bring meaningful conclusions for the solution of the data imbalance problem.

The test results are shown in Figure 10. For the experimental data case, because of the severe data imbalance problem, the training process of the 1D CNN model is biased for a healthy state, dropping the fault diagnosis performance of fault states, as shown in Figure 10a. This bias caused the 1D CNN model to diagnose the fault state improperly, showing inferior fault diagnosis performance. The test result for synthetic data is depicted in Figure 10b. Compared to the results from the experimental data, the performance for the healthy state decreased slightly. However, the classification results for faulty states increased dramatically; 23 cases were predicted correctly out of 24 cases. From this, it could be concluded that the proposed data augmentation technique increased the overall fault diagnosis performance.



(**a**) (**b**)

Figure 10. (**a**) Fault diagnosis results for the experimental data only. (**b**) Fault diagnosis results for the synthetic data.

At first glance, the accuracy seems to show only a negligible improvement, rising from 89.29% to 91.96%. However, the other metrics, precision and sensitivity, showed much-improved outcomes. The precision surged from 29.76% to 74.10%, while the sensitivity soared from 33.33% to 94.39%. These differences in results across the metrics derive from the imbalanced nature of the dataset. Imbalanced datasets can often lead to misleading accuracy metrics, as the model might perform very well on the majority class but poorly on the minority class. In such cases, precision and sensitivity provide a more accurate understanding of the model's performance, especially in identifying and classifying the underrepresented class.

4.3. Comparative Analysis

To demonstrate that the WGAN model is the best solution for addressing the data imbalance problem in the vibration signal, a comparative analysis was conducted between the WGAN model and the other methods of ADASYN, SI, and GAN. Table 5 describes the model parameters of the GAN model. The same structure of the discriminator and generator as the WGAN model was used and the corresponding parameters were adjusted by trial and error. Through the parameter optimizing results, it was observed that even though the same discriminator and critic models were utilized, the parameters were completely different according to the type of loss function.

Table 5. Model parameters of the GAN model for each health state.

	Healthy	Delamination 1	Delamination 2
Beta 1	0.5	0.5	0.5
Beta 2	0.999	0.999	0.999
Learning rate	1×10^{-3}	1×10^{-3}	1×10^{-3}
Epoch	1200	1200	1200
Batch size	8	8	8

Table 6 shows the number of synthetic data for each method. Basically, the ADASYN algorithm is an oversampling method, which means that it can only generate minority class data, the D1 and D2 cases in this research. Therefore, the total amount of data is different from the other methods. For all the methods, synthetic data instances were randomly divided into 60% and 40% for training and validation while maintaining the proportion of each class.

Table 6. The number of synthetic data for training and validation using the different methods.

	Healthy	Delamination 1	Delamination 2
Experimental data only	800	48	48
ADASYN	800	793	798
SI	3000	3000	3000
GAN	3000	3000	3000
WGAN	3000	3000	3000

Table 7 shows the fault diagnosis results for the same test dataset as Section 4.2. The accuracy of the WGAN model was the highest result at 91.96%. Also, the experimental data-only case has relatively higher accuracy than the other methods. As mentioned above, this originates from the data imbalance. The majority of the class data was classified properly; thus, the accuracy itself turned out to be high. For precision, ADASYN was the highest, at 77.80%, with a difference of 3.70% compared to the WGAN model. For sensitivity, WGAN had the highest value at 94.39%, which is 3.11% higher than the ADASYN. By comparing the precision and sensitivity of the ADASYN and WGAN, WGAN seems to have a better ability to identify fault states properly. The health state data for ADASYN may be classified more properly than that for WGAN, and this point brought higher precision in ADASYN but lower performance in sensitivity. For the F1 score, ADASYN and WGAN were the highest at 0.81.

Table 7. Fault diagnosis result of each oversampling and augmentation method.

	Accuracy (%)	Precision (%)	Sensitivity (%)	F1 Score
Experimental data only	89.29	29.76	33.33	0.31
ADASYN	90.63	77.80	91.28	0.81
SI	5.80	35.13	35.00	0.04
GAN	86.61	53.83	55.83	0.55
WGAN	91.96	74.10	94.39	0.81

For overall analysis, the synthetic data generated with SI seems not to be proper for this research because all evaluation metrics were much lower than the other methods. In the case of ADASYN, because of the characteristic of oversampling, real data are contained in the training and validation datasets, while the rest of them are only composed of synthetic data. This could increase the fault diagnosis performance compared with the others. For the ADASYN, GAN, and WGAN cases, the evaluation metrics increased significantly compared to the experimental data case, but the WGAN model showed the best evaluation metrics, which means that the WGAN model can generate signals that contain dynamic characteristics that are the same as real signals.

5. Conclusions

This study addresses the time-series data augmentation technique using the WGAN model for the solution of the data imbalance problem and fault diagnosis using the 1D CNN model. For time-series data augmentation, vibration signals were directly used as input data for the WGAN model. To verify the performance increase in fault diagnosis by applying the WGAN model, experiments using laminated composite beam specimens to obtain vibration signals were conducted. For imitation of the data imbalance dataset, 1000 signals for the healthy state and 60 signals for faulty states were obtained through the experiments, and a total of 9000 synthetic signals were generated by using the WGAN model. To verify the enhanced fault diagnosis performance, the results from a 1D CNN model using both experimental and synthetic datasets were compared. The results demonstrated a significant improvement in fault diagnosis. Furthermore, a comparative study revealed that data augmentation with the WGAN model yielded the best diagnostic performance.

In contrast to previous works, this research has simplified the data augmentation and fault diagnosis process by directly using time-series data as input. The simplification of the process saves computational resources because both the WGAN and 1D CNN models use time-series data, which has a smaller data size than the image data. Thus, training time, as well as predicting time, are also much shorter, which means the enhanced probability of applications in real-time fault diagnosis. For the last contribution, fault diagnosis could be performed by inexpensive and simple experiments. The accelerometer is a cheap sensor among various sensors, which means that data can be obtained efficiently.

Notwithstanding these contributions to the solution of the data imbalance problem and the fault diagnosis process, some limitations still exist. After the advent of the GAN model, much research related to GAN has been conducted, but a generalization of the GAN model for various engineering problems is still one of the critical issues [61]. Thus, the trial and error process is generally employed to optimize the parameters of GAN models; however, it is a time-consuming and laborious method. A reasonable technique or specific standard for optimizing the parameters of the GAN model is essential. For the next limitation, the 1D CNN model can extract features and diagnose the health state of the composite system, as mentioned above. However, because the 1D CNN model is a black box model, it is unclear which part of the signal was considered, especially for fault diagnosis. To improve this hardship in explainability, the eXplainable Artificial Intelligence (XAI) algorithm will be used in future work. Through the XAI algorithm, it is anticipated that the features that are considered more important in the fault diagnosis process will be identified.

Also, this research focused on evaluating data augmentation performance. For future research, a generalization of the fault diagnosis model will be conducted. For this purpose, a large number of specimens with various conditions will be tested. Then, data augmentation will be implemented for the training dataset to prove that the fault diagnosis model can accurately classify the data that are not trained for the model.

Author Contributions: Conceptualization, S.K., M.M.A., J.S. and H.K.; methodology, S.K., M.M.A. and J.S.; software, S.K.; validation, S.K., M.M.A. and J.S.; formal analysis, S.K. and J.S.; investigation, S.K. and M.M.A.; resources, S.K. and J.S.; data curation, S.K. and M.M.A.; writing—original draft preparation, S.K. and M.M.A.; writing—review and editing, J.S. and H.K.; visualization, S.K.; supervision, J.S. and H.K.; project administration, H.K.; funding acquisition, H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2020R1A2C1006613) and also supported by the MOTIE (Ministry of Trade, Industry, and Energy) in Korea, under the Fostering Global Talents for Innovative Growth Program (P0017307) supervised by the Korea Institute for Advancement of Technology (KIAT).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, J.; Lin, G.; Vaidya, U.; Wang, H. Past, Present and Future Prospective of Global Carbon Fibre Composite Developments and Applications. *Compos. Part B Eng.* **2023**, *250*, 110463. [CrossRef]
2. Azad, M.M.; Ejaz, M.; Shah, A.R.; Kamran Afaq, S.; Song, J. Static Mechanical Properties of Bio-Fiber-Based Polymer Composites. In *Advances in Bio-Based Fiber*; Elsevier: Amsterdam, The Netherlands, 2022; pp. 97–139.
3. Bui, T.Q.; Hu, X. A Review of Phase-Field Models, Fundamentals and Their Applications to Composite Laminates. *Eng. Fract. Mech.* **2021**, *248*, 107705. [CrossRef]
4. Khalid, S.; Kim, H.S. Recent Studies on Stress Function-Based Approaches for the Free Edge Stress Analysis of Smart Composite Laminates: A Brief Review. *Multiscale Sci. Eng.* **2022**, *4*, 73–78. [CrossRef]
5. Banks-Sills, L. Interface Fracture and Delaminations in Composite Materials; SpringerBriefs. In *Applied Sciences and Technology*; Springer International Publishing: Cham, Switzerland, 2018; ISBN 978-3-319-60326-1.
6. Mortell, D.J.; Tanner, D.A.; McCarthy, C.T. In-Situ SEM Study of Transverse Cracking and Delamination in Laminated Composite Materials. *Compos. Sci. Technol.* **2014**, *105*, 118–126. [CrossRef]
7. Azad, M.M.; Kim, S.; Cheon, Y.B.; Kim, H.S. Intelligent Structural Health Monitoring of Composite Structures Using Machine Learning, Deep Learning, and Transfer Learning: A Review. *Adv. Compos. Mater.* **2023**, 1–27. [CrossRef]
8. Liu, G.; Li, L.; Zhang, L.; Li, Q.; Law, S.S. Sensor Faults Classification for SHM Systems Using Deep Learning-Based Method with Tsfresh Features. *Smart Mater. Struct.* **2020**, *29*, 075005. [CrossRef]
9. Deng, F.; Tao, X.; Wei, P.; Wei, S. A Robust Deep Learning-Based Damage Identification Approach for SHM Considering Missing Data. *Appl. Sci.* **2023**, *13*, 5421. [CrossRef]
10. Finotti, R.P.; Cury, A.A.; Barbosa, F.D.S. An SHM Approach Using Machine Learning and Statistical Indicators Extracted from Raw Dynamic Measurements. *Lat. Am. J. Solids Struct.* **2019**, *16*, e165. [CrossRef]
11. Song, G.; Wang, C.; Wang, B. Structural Health Monitoring (SHM) of Civil Structures. *Appl. Sci.* **2017**, *7*, 789. [CrossRef]
12. Galan-Uribe, E.; Morales-Velazquez, L.; Osornio-Rios, R.A. FPGA-Based Methodology for Detecting Positional Accuracy Degradation in Industrial Robots. *Appl. Sci.* **2023**, *13*, 8493. [CrossRef]
13. Zhang, C.; Mousavi, A.A.; Masri, S.F.; Gholipour, G.; Yan, K.; Li, X. Vibration Feature Extraction Using Signal Processing Techniques for Structural Health Monitoring: A Review. *Mech. Syst. Signal Process.* **2022**, *177*, 109175. [CrossRef]
14. Sohn, H.; Farrar, C.R. Damage Diagnosis Using Time Series Analysis of Vibration Signals. *Smart Mater. Struct.* **2001**, *10*, 446–451. [CrossRef]
15. Tcherniak, D.; Mølgaard, L.L. Vibration-Based SHM System: Application to Wind Turbine Blades. *J. Phys. Conf. Ser.* **2015**, *628*, 012072. [CrossRef]
16. Amezquita-Sanchez, J.P.; Adeli, H. Signal Processing Techniques for Vibration-Based Health Monitoring of Smart Structures. *Arch. Comput. Methods Eng.* **2016**, *23*, 1–15. [CrossRef]
17. Lei, Y.; Yang, B.; Jiang, X.; Jia, F.; Li, N.; Nandi, A.K. Applications of Machine Learning to Machine Fault Diagnosis: A Review and Roadmap. *Mech. Syst. Signal Process.* **2020**, *138*, 106587. [CrossRef]
18. Hoang, D.-T.; Kang, H.-J. A Survey on Deep Learning Based Bearing Fault Diagnosis. *Neurocomputing* **2019**, *335*, 327–335. [CrossRef]
19. Jiang, L.; Yin, H.; Li, X.; Tang, S. Fault Diagnosis of Rotating Machinery Based on Multisensor Information Fusion Using SVM and Time-Domain Features. *Shock Vib.* **2014**, *2014*, 418178. [CrossRef]
20. Samanta, B.; Al-Balushi, K.R. Artificial Neural Network Based Fault Diagnostics of Rolling Element Bearings Using Time-Domain Features. *Mech. Syst. Signal Process.* **2003**, *17*, 317–328. [CrossRef]
21. Ye, L.; Ma, X.; Wen, C. Rotating Machinery Fault Diagnosis Method by Combining Time-Frequency Domain Features and CNN Knowledge Transfer. *Sensors* **2021**, *21*, 8168. [CrossRef]
22. Li, C.; Sanchez, V.; Zurita, G.; Cerrada Lozada, M.; Cabrera, D. Rolling Element Bearing Defect Detection Using the Generalized Synchrosqueezing Transform Guided by Time–Frequency Ridge Enhancement. *ISA Trans.* **2016**, *60*, 274–284. [CrossRef]
23. Hemmati, F.; Orfali, W.; Gadala, M.S. Roller Bearing Acoustic Signature Extraction by Wavelet Packet Transform, Applications in Fault Detection and Size Estimation. *Appl. Acoust.* **2016**, *104*, 101–118. [CrossRef]
24. He, D.; Li, R.; Zhu, J. Plastic Bearing Fault Diagnosis Based on a Two-Step Data Mining Approach. *IEEE Trans. Ind. Electron.* **2012**, *60*, 3429–3440. [CrossRef]
25. Saravanan, N.; Kumar Siddabattuni, V.N.S.; Ramachandran, K.I. A Comparative Study on Classification of Features by SVM and PSVM Extracted Using Morlet Wavelet for Fault Diagnosis of Spur Bevel Gear Box. *Expert Syst. Appl.* **2008**, *35*, 1351–1366. [CrossRef]
26. Amarnath, M.; Sugumaran, V.; Kumar, H. Exploiting Sound Signals for Fault Diagnosis of Bearings Using Decision Tree. *Measurement* **2013**, *46*, 1250–1256. [CrossRef]
27. Zhong, S.; Fu, S.; Lin, L. A Novel Gas Turbine Fault Diagnosis Method Based on Transfer Learning with CNN. *Measurement* **2019**, *137*, 435–453. [CrossRef]

28. Janssens, O.; Slavkovikj, V.; Vervisch, B.; Stockman, K.; Loccufier, M.; Verstockt, S.; Van De Walle, R.; Van Hoecke, S. Convolutional Neural Network Based Fault Detection for Rotating Machinery. *J. Sound Vib.* **2016**, *377*, 331–345. [CrossRef]
29. Hsueh, Y.-M.; Ittangihal, V.R.; Wu, W.-B.; Chang, H.-C.; Kuo, C.-C. Fault Diagnosis System for Induction Motors by CNN Using Empirical Wavelet Transform. *Symmetry* **2019**, *11*, 1212. [CrossRef]
30. Khan, A.; Ko, D.-K.; Lim, S.C.; Kim, H.S. Structural Vibration-Based Classification and Prediction of Delamination in Smart Composite Laminates Using Deep Learning Neural Network. *Compos. Part B Eng.* **2019**, *161*, 586–594. [CrossRef]
31. Tang, X.; Xu, Z.; Wang, Z. A Novel Fault Diagnosis Method of Rolling Bearing Based on Integrated Vision Transformer Model. *Sensors* **2022**, *22*, 3878. [CrossRef]
32. Wen, L.; Li, X.; Gao, L. A Transfer Convolutional Neural Network for Fault Diagnosis Based on ResNet-50. *Neural. Comput. Appl.* **2020**, *32*, 6111–6124. [CrossRef]
33. Zhao, H.; Sun, S.; Jin, B. Sequential Fault Diagnosis Based on LSTM Neural Network. *IEEE Access* **2018**, *6*, 12929–12939. [CrossRef]
34. Yuan, M.; Wu, Y.; Lin, L. Fault Diagnosis and Remaining Useful Life Estimation of Aero Engine Using LSTM Neural Network. In Proceedings of the 2016 IEEE International Conference on Aircraft Utility Systems (AUS), Beijing, China, 8–14 October 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 135–140.
35. Sabir, R.; Rosato, D.; Hartmann, S.; Guehmann, C. LSTM Based Bearing Fault Diagnosis of Electrical Machines Using Motor Current Signal. In Proceedings of the 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 613–618.
36. Zhang, T.; Fei, Q.; Li, N.; Ma, D. Fault Diagnosis Based on Modified BiLSTM Neural Network. In Proceedings of the 2020 5th International Conference on Intelligent Information Technology, Hanoi, Vietnam, 19–22 February 2020; ACM Association for Computing Machinery: New York, NY, USA, 2020; pp. 21–26. [CrossRef]
37. Zhang, W.; Li, X.; Jia, X.-D.; Ma, H.; Luo, Z.; Li, X. Machinery Fault Diagnosis with Imbalanced Data Using Deep Generative Adversarial Networks. *Measurement* **2020**, *152*, 107377. [CrossRef]
38. Khan, A.; Azad, M.M.; Sohail, M.; Kim, H.S. A Review of Physics-Based Models in Prognostics and Health Management of Laminated Composite Structures. *Int. J. Precis. Eng. Manuf. Green Technol.* **2023**, 1–21. [CrossRef]
39. Jung, K.-C.; Chang, S.-H. Advanced Deep Learning Model-Based Impact Characterization Method for Composite Laminates. *Compos. Sci. Technol.* **2021**, *207*, 108713. [CrossRef]
40. Sikdar, S.; Liu, D.; Kundu, A. Acoustic Emission Data Based Deep Learning Approach for Classification and Detection of Damage-Sources in a Composite Panel. *Compos. Part B Eng.* **2022**, *228*, 109450. [CrossRef]
41. Liao, Y.; Qing, X.; Wang, Y.; Zhang, F. Damage Localization for Composite Structure Using Guided Wave Signals with Gramian Angular Field Image Coding and Convolutional Neural Networks. *Compos. Struct.* **2023**, *312*, 116871. [CrossRef]
42. Feng, B.; Cheng, S.; Deng, K.; Kang, Y. Localization of Low-Velocity Impact in CFRP Plate Using Time–Frequency Features of Guided Wave and Convolutional Neural Network. *Wave Motion* **2023**, *119*, 103127. [CrossRef]
43. Cheng, L.; Tong, Z.; Xie, S.; Kersemans, M. IRT-GAN: A Generative Adversarial Network with a Multi-Headed Fusion Strategy for Automated Defect Detection in Composites Using Infrared Thermography. *Compos. Struct.* **2022**, *290*, 115543. [CrossRef]
44. Meister, S.; Möller, N.; Stüve, J.; Groves, R.M. Synthetic Image Data Augmentation for Fibre Layup Inspection Processes: Techniques to Enhance the Data Set. *J. Intell. Manuf.* **2021**, *32*, 1767–1789. [CrossRef]
45. Cheng, L.; Kersemans, M. Dual-IRT-GAN: A Defect-Aware Deep Adversarial Network to Perform Super-Resolution Tasks in Infrared Thermographic Inspection. *Compos. Part B Eng.* **2022**, *247*, 110309. [CrossRef]
46. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–6 June 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1322–1328.
47. Marra, A.L.; Juliani, R.; Garcia, C. Data Augmentation for Vibration Signals Using System Identification Techniques. In Proceedings of the 2021 5th International Conference on System Reliability and Safety (ICSRS), Palermo, Italy, 24–26 November 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 281–285.
48. Khan, A.; Raouf, I.; Noh, Y.R.; Lee, D.; Sohn, J.W.; Kim, H.S. Autonomous Assessment of Delamination in Laminated Composites Using Deep Learning and Data Augmentation. *Compos. Struct.* **2022**, *290*, 115502. [CrossRef]
49. Phan, M.; Solbeck, J.; Ray, L. A Direct Method for State-Space Model and Observer/Kalman Filter Gain Identification. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, American Institute of Aeronautics and Astronautics, Providence, RI, USA, 16–19 August 2004.
50. Yang, Z.; Li, Y.; Zhou, G. TS-GAN: Time-Series GAN for Sensor-Based Health Data Augmentation. *ACM Trans. Comput. Healthc.* **2023**, *4*, 1–21. [CrossRef]
51. Lu, H.; Du, M.; Qian, K.; He, X.; Wang, K. GAN-Based Data Augmentation Strategy for Sensor Anomaly Detection in Industrial Robots. *IEEE Sens. J.* **2022**, *22*, 17464–17474. [CrossRef]
52. Smith, K.E.; Smith, A.O. Conditional GAN for Timeseries Generation. *arXiv* **2020**, arXiv:2006.16477v1.
53. Boicea, V.A.; Ulmeanu, A.P.; Vulpe-Grigoraşi, A. A Novel Approach for Power Load Forecast Based on GAN Data Augmentation. *IOP Conf. Ser. Mater. Sci. Eng.* **2022**, *1254*, 012030. [CrossRef]
54. Arjovsky, M.; Bottou, L. Towards Principled Methods for Training Generative Adversarial Networks. *arXiv* **2017**, arXiv:1701.04862v1.
55. Goodfellow, I. NIPS 2016 Tutorial: Generative Adversarial Networks. *arXiv* **2017**, arXiv:1701.00160v4.

56. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875v3.
57. Tang, S.; Yuan, S.; Zhu, Y. Data Preprocessing Techniques in Convolutional Neural Network Based on Fault Diagnosis towards Rotating Machinery. *IEEE Access* **2020**, *8*, 149487–149496. [CrossRef]
58. Neupane, D.; Kim, Y.; Seok, J.; Hong, J. CNN-Based Fault Detection for Smart Manufacturing. *Appl. Sci.* **2021**, *11*, 11732. [CrossRef]
59. Chen, C.-C.; Liu, Z.; Yang, G.; Wu, C.-C.; Ye, Q. An Improved Fault Diagnosis Using 1D-Convolutional Neural Network Model. *Electronics* **2020**, *10*, 59. [CrossRef]
60. Fu, Q.; Wang, H. A Novel Deep Learning System with Data Augmentation for Machine Fault Diagnosis from Vibration Signals. *Appl. Sci.* **2020**, *10*, 5765. [CrossRef]
61. Thanh-Tung, H.; Tran, T.; Venkatesh, S. Improving Generalization and Stability of Generative Adversarial Networks. *arXiv* **2019**, arXiv:1902.03984v1.

applied sciences

MDPI

Article

FPGA-Based Methodology for Detecting Positional Accuracy Degradation in Industrial Robots

Ervin Galan-Uribe, Luis Morales-Velazquez and Roque Alfredo Osornio-Rios *

Facultad de Ingeniería, Universidad Autónoma de Querétaro, Campus San Juan del Río, Río Moctezuma 249, Col. San Cayetano, San Juan del Río 76807, Mexico; egalan18@alumnos.uaq.mx (E.G.-U.); luis.moralesv@uaq.mx (L.M.-V.)
* Correspondence: raosornio@hspdigital.org

Abstract: Industrial processes involving manipulator robots require accurate positioning and orienting for high-quality results. Any decrease in positional accuracy can result in resource wastage. Machine learning methodologies have been proposed to analyze failures and wear in electronic and mechanical components, affecting positional accuracy. These methods are typically implemented in software for offline analysis. In this regard, this work proposes a methodology for detecting a positional deviation in the robot's joints and its implementation in a digital system of proprietary design based on a field-programmable gate array (FPGA) equipped with several developed intellectual property cores (IPcores). The method implemented in FPGA consists of the analysis of current signals from a UR5 robot using discrete wavelet transform (DWT), statistical indicators, and a neural network classifier. IPcores are developed and tested with synthetic current signals, and their effectiveness is validated using a real robot dataset. The results show that the system can classify the synthetic robot signals for joints two and three with 97% accuracy and the real robot signals for joints five and six with 100% accuracy. This system aims to be a high-speed reconfigurable tool to help detect robot precision degradation and implement timely maintenance strategies.

Keywords: FPGA; positional accuracy; discrete wavelet transform; industrial robot; degradation; neural networks

1. Introduction

Robotic systems have revolutionized the manufacturing industry by providing unprecedented accuracy, precision, and speed. Accuracy is critical to ensure the robot can position and orient itself to the required locations and repeat the task over a long time with minimal error. One way to enhance the positioning accuracy of the robot is by focusing on the design stage. For instance, Kelaiaia et al. [1], present a comprehensive review of optimal design aspects for parallel manipulators and propose a methodology for achieving the optimal design of such manipulators. Additionally, improving robot performance indices can contribute to better accuracy. Brahmia et al. [2], propose a new dimensionless sensitivity index for Parallel Kinematic Manipulators (PKM), based on the definition of the local sensitivity index (LSI). This index aids in identifying the contribution of error sources to the positioning error of the manipulator for a given task. By reducing geometric errors in PKMs, this index can significantly enhance the positioning accuracy of the robot.

Despite their advanced capabilities, robots can still experience accuracy degradation due to several factors such as environmental conditions, assembly errors, manufacturing defects in the structure, backlash, and component wear [3]. Therefore, it is essential to monitor and maintain the accuracy of robotic systems to ensure optimal performance over time. Different factors involved in the positional degradation of industrial robots and failures caused by wear in electrical and mechanical components are frequently studied. In this regard, methods based on machine learning techniques have been proposed to identify failures, such as in [4] where the wavelet packet transform (WPT) and the hidden Markov model (HMM) are used to analyze acoustic emissions (EA) and identify failures in a rotate vector reducer (RV). A method based on artificial failure data, random forest regression (RFR), support vector regression (SVR), and deep neural network regression (DNNR) is proposed in [5] to identify overload anomalies in the robot's end effector and their effect on the joints. Moreover, in [6], a simulation of the torque degradation of the actuator of an industrial robot is carried out, where it is determined that the robot controller cannot deal with failures of this type. Other works related to methodologies that deal with mechanical failures in robots can be consulted for gears [7–10] and RV [11].

Recent research on deep learning and fault detection techniques for robots includes several noteworthy works. In [12], the authors propose a method that combines a sparse auto-decoder with a support vector machine (SVM) to construct a fault detection model for the robot's reducer. They utilize signals obtained from an attitude sensor composed of an accelerometer, gyroscope, and magnetometer. Another study, presented in [13], introduces a method based on a one-dimensional convolutional neural network (1DCNN) with matrix kernels. This approach is designed to diagnose faults in harmonic reducers used in industrial robots by analyzing vibration signals. In addition, [14] describes a fault diagnosis method that employs Deep Convolutional Neural Networks (DCNN) to detect faults in the robot's joints. The model is trained using a database generated by modeling the actuators and sensors. Meanwhile, Ref. [15] presents a data-driven approach utilizing Deep Residual Neural Networks (DRNN) to detect faults in robot joints, also based on sensor and actuator modeling. Furthermore, Jiao and Zheng [16] propose a method for detecting joint bearing failures in industrial robots. They utilize deep belief networks (DBN) to analyze the robot joint vibration signals and identify potential issues.

On the other hand, there have been works that directly address the problem of positional degradation of industrial robots, e.g., in [3], the dependent errors present in the robot produced by non-ideal deflections of the structure and movement are characterized using polynomials of Chebyshev for an advanced error model. Taha et al. [17] use multivariable regression adjustment (MRA) and deep long short-term memory (DLSTM) to predict and model the displacement of an industrial robot and to estimate the residual error on the end-effector. An expert system for detecting a deviation in the joints of a robot is proposed in [18] and is based on DWT analysis, neural networks, and fractal and energy features. Additionally, methods based on artificial vision systems to carry out studies on the deviation of the robot have been proposed in [19–22], they have even been complemented with re-calibration methods through kinematic analysis of the robot as the work presented in [23].

It is important to note that some of the presented methods involve performing analysis on personal computers (PCs) after the necessary data acquisition, thus operating offline. Therefore, integrating robot failure or position degradation analysis methods into reconfigurable processing systems, such as Field Programmable Gate Arrays (FPGAs), provides the opportunity to perform non-invasive information analysis, online or offline, quickly and efficiently. In this sense, it is worth noting that numerous FPGA-based approaches have been developed tailored for robotics. For example, Sun et al. [24] propose an FPGA-based torque predictor control focused on legged robots. The torque estimation is performed considering the delay in the calculation and measurement, improving the prediction and period of the controller, and obtaining a better controller performance. The acceleration of the kinematic models of a 6-degree-of-freedom (DOF) robot arm implemented in FPGA is presented in [25], emphasizing the inverse kinematic model required in the robot motion. The speed and accuracy of the processing of the proposed method demonstrate its effectiveness in real-time robot motion. Besides, the implementation in FPGA of a control system for a 3DOF robot with pneumatic actuators is presented in [26]. Proportional integral derivative controllers (PID) are used for air valve management and position control of the robot. As well as a neural network is used to tune the position controller gains, and although it obtains some disturbances attributed to the ANN tuning process, the results are satisfactory.

Liu et al. [27] introduced a balance controller for a humanoid robot. The system, implemented on an FPGA, comprises several stages, including external force detection, recovery balance control, trajectory planning, and inverse kinematics. Their method enables the humanoid robot to effectively recover its balance after experiencing external force. In addition, in [28], a method based on artificial vision is proposed to detect the deviation in a conveyor belt in real time using FPGAs and the Line Segment Detection algorithm (LSD), obtaining a high value of images processed per second, which improves the performance in real-time of the fault detection method. There have also been developed works focused on modular, reconfigurable robots (MRR) as [29], where a distributed computing system for multiple object tracking based on FPGA and Kalman filters is presented. It uses specialized computational cores (SCCs) and a strategy to use the least amount of FPGA resources for real-time motion control tasks of MRR. They reconfigure the FPGA by replacing complex models with simpler SCCs to reduce power consumption. The results show better performance, resource usage, and computational accuracy than high-performance CPUs. Furthermore, Plancher et al. [30] propose implementing the dynamic model of robots on CPU, GPU, and FPGA through the Recursive Newton-Euler Algorithm (RNEA) computation. The results demonstrate the superior performance of the GPU and FPGA over the CPU as the number of computations increases. Furthermore, works have also been developed on manufacturing equipment such as computer numerical control machines (CNC); for example, Ref. [31] proposes an anomaly detection system in the milling process based on FPGA. The extraction of features in the frequency domain of machine vibration signals is performed using discrete Fourier transform (DFT), and this information is used to enter an auto-associative neural network (AANN) for anomaly detection, reporting satisfactory results offline. Other recent FPGA applications in robotics are agricultural robots [32], reactive robotics [33], and grasping recognition [34], among others. The works presented have demonstrated the usefulness and advantages of using FPGA processing systems in robotics problems.

Considering the importance of the robot to perform tasks without losing its accuracy, some of the mentioned works that address the issue of positional degradation and fault detection require the use of external sensors such as cameras, lasers, or microphones to acquire the signals for analysis. These external sensors can work correctly in controlled environments, but in real environments, they are affected by various conditions such as sensor size, noise, illumination, temperature, and humidity This makes it difficult to implement in an industrial environment; however, a portable device of a small size equipped with the necessary components to perform signal acquisition and real-time processing presents an alternative to traditional methods of signal acquisition where data processing is performed offline. In this sense, FPGA-based digital systems are an option to perform the task of data acquisition and processing in a single, non-invasive device and have the advantages of reconfiguration and high processing speed

In this context, the contribution of this work lies in proposing a methodology to detect positional degradation in industrial robot joints that is suitable for implementation in FPGA-based digital systems. The approach involves analyzing the current signals of their actuators to identify any anomalous behaviors that may result in end-effector deviation. For this purpose, the current signals from the robot actuators are generated synthetically by modeling the UR5 robot dynamics. The synthetic dataset contains signals with and without deviation. Then different noise levels (5 dB, 10 dB, 15 dB, and 20 dB) are added to the signals to enhance the method's robustness. Subsequently, the current signal is processed for a time-frequency analysis using the DWT. Then statistical indices are computed for the two last levels of decomposition. Finally, a neural network is trained using the indices data to classify each robot joint. The proposed method is implemented in a digital system board FPGA-based proprietary design and validated using a real UR5 robot dataset. For the synthetic case, the results show that the proposed method can classify the states of the robot's operation with and without deviation for the joints with more significant intervention in the task. The results obtained with the real-world dataset (joint five) are supported by results reported in the literature. Implementing the method in FPGA allows a quick classification of less than one second for all the joints of the robot, providing a functional system with the advantage of being a high-frequency operating system, reconfigurable, portable, and energy efficient.

2. Materials and Methods

The techniques used to carry out the proposed methodology and its implementation in a digital system based on FPGA are presented below. A general review of robot theory, the DWT time-frequency analysis technique, statistical indicators, and a quick look at neural networks are conducted. On the other hand, a general description of the FPGA device used for this work is made.

2.1. Robot Overview

The robot used for this paper is the UR5 robot by Universal Robotics; a graphic representation of the robot is presented in Figure 1a,b.

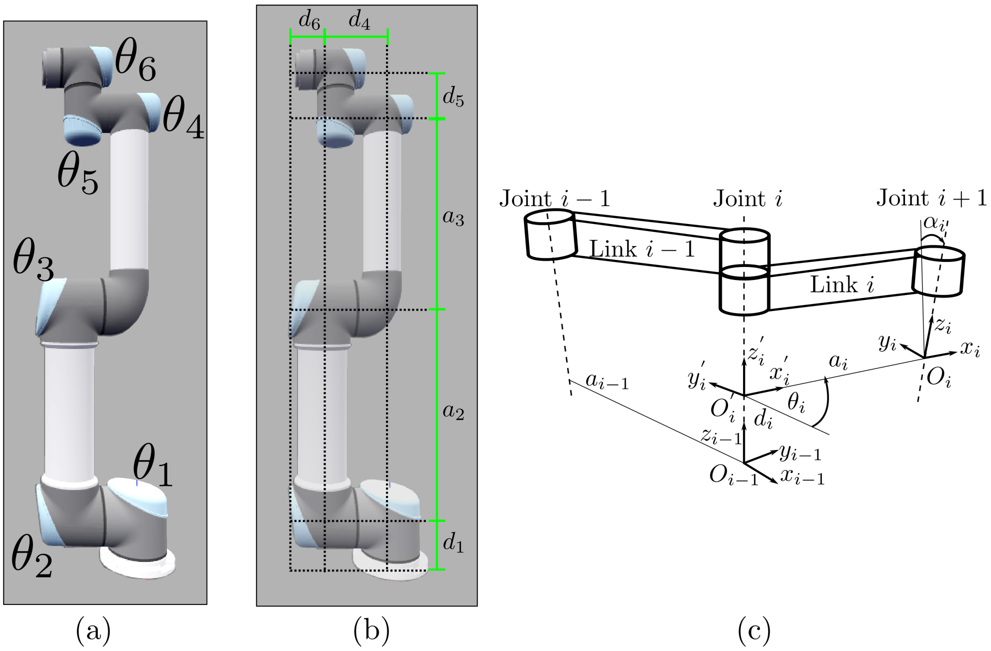


Figure 1. (**a**) UR5 robot arm (figure generated using the library Robotics Toolbox for Python [35]). (**b**) DH parameters for robot link lengths (**c**) Standard DH graphical representation.

It comprises six rotational joints providing six degrees of freedom. To approximate the robot motors' currents, the robot's dynamic model and the torque-current relationship are proposed to estimate the currents synthetically.

The Denavit-Hartenberg (DH) parameters presented in Table 1 allow the description of the robot through four characteristics that enable the calculation of the positions of the different joints of the robot using transformation matrices obtaining the kinematic model, for which it is recommended to review the following source for the forward and inverse models [36].

For Table 1 values, using DH notation Figure 1c:

- θ_i is the angle between the axes x_{i-1} and x_i around z_{i-1}, taken as positive in a counter-clockwise rotation. The y_i axis is located according to the right-hand rule.
- d_i is the coordinate of O_i' along the axis z_{i-1}.
- a_i is the distance from O_i to O_i'.
- α_i is the angle between z_{i-1} and z_i around x_i, taken positive in a counter-clockwise rotation.

Table 1. Denavint-Hatenberg parameters for a UR5 manipulator.

Joint	θ_i (rad)	d_i (m)	a_i (m)	α_i (rad)
1	θ_1	0.089159	0	$\pi/2$
2	θ_2	0	−0.425	0
3	θ_3	0	−0.039225	0
4	θ_4	0.10915	0	$\pi/2$
5	θ_5	0.09465	0	$-\pi/2$
6	θ_6	0.0823	0	0

On the other hand, the dynamic model of the robot allows for estimating the robot's behavior considering the forces present in the system. Equation (1) describes the motion of the robot

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau, \tag{1}$$

and it is composed by the inertia matrix $D(q)$, Coriolis and centrifugal matrix $C(q,\dot{q})$, gravity vector $g(q)$, torque vector τ,the articular position q, velocity $\dot{q}$, and acceleration $\ddot{q}$ vectors. To implement the model, it is necessary to know additional information about the physical properties of the joints and motors.

Table 2 shows each joint's mass parameters, the center of mass coordinates, and the corresponding inertia tensor.

Table 2. Dynamic parameters for the links of a UR5 robotic arm.

Link	Mass (kg)	Center of Mass $_{xyz}$ (m)	$\mathbf{I}_{xyz}$ (kg m^2)
1	3.7	$\begin{bmatrix} 0 & -0.02561 & 0.00193 \end{bmatrix}$	$\begin{bmatrix} 0.00375 & 0.00765 & 0.00765 \end{bmatrix}$
2	8.393	$\begin{bmatrix} 0.2125 & 0 & 0.11336 \end{bmatrix}$	$\begin{bmatrix} 0.0085 & 0.208 & 0.208 \end{bmatrix}$
3	2.33	$\begin{bmatrix} 0.15 & 0 & 0.0265 \end{bmatrix}$	$\begin{bmatrix} 2.46 \times 10^{-4} & 0.00719 & 0.00719 \end{bmatrix}$
4	1.219	$\begin{bmatrix} 0 & -0.0018 & 0.01634 \end{bmatrix}$	$\begin{bmatrix} 9.09 \times 10^{-4} & 0.00119 & 0.00119 \end{bmatrix}$
5	1.219	$\begin{bmatrix} 0 & 0.0018 & 0.01634 \end{bmatrix}$	$\begin{bmatrix} 9.09 \times 10^{-4} & 0.00119 & 0.00119 \end{bmatrix}$
6	0.1879	$\begin{bmatrix} 0 & 0 & -0.00159 \end{bmatrix}$	$\begin{bmatrix} 1.22 \times 10^{-4} & 8.21 \times 10^{-5} & 8.21 \times 10^{-5} \end{bmatrix}$

Table 3 shows the parameters corresponding to the motor inertia Jm, gear ratio G, drive viscous friction B, and Coulomb friction Tc.

Table 3. Dynamic parameters for the motors of a UR5 robotic arm.

Link	Jm (kg m^2)	G	B (Nms rad^{-1})	Tc $[-,+]$ (Nm)
1	1.87×10^{-8}	101	1×10^{-4}	$\begin{bmatrix} 0.076 & -0.076 \end{bmatrix}$
2	1.87×10^{-8}	101	1×10^{-4}	$\begin{bmatrix} 0.083 & -0.082 \end{bmatrix}$
3	1.87×10^{-8}	101	1×10^{-4}	$\begin{bmatrix} 0.078 & -0.077 \end{bmatrix}$
4	2.7×10^{-5}	101	1×10^{-4}	$\begin{bmatrix} 0.014 & -0.014 \end{bmatrix}$
5	2.7×10^{-5}	101	1×10^{-4}	$\begin{bmatrix} 0.020 & -0.019 \end{bmatrix}$
6	2.7×10^{-5}	101	1×10^{-4}	$\begin{bmatrix} 0.020 & -0.021 \end{bmatrix}$

Note that the manufacturer provides all the parameters, except the drive viscous friction B, in different media and can be consulted in summary form in [37]. Parameter B has been proposed in a heuristic way since the calculation and modeling of the friction for the joints of the robot are outside the main objective of this work.

From the Equation (1) τ also can be expressed in terms of the relationship between torque and current (2):

$$\hat{\tau} = K_{ti} ia_i, \tag{2}$$

where $\hat{\tau}$ is the estimated torque of the model considering the effects of friction and without the intervention of external forces, K_{ti} and ia_i are the torque constants, and the motor current for the joint i, respectively. To estimate the motor current from the torque, a linear model proposed in [38] is used, where the torque constants and an offset are obtained, with which they establish a relationship between the modeled torque and the measured current. These parameters are proposed to compute the inverse process and estimate the motor current from a modeled torque. Equation (2) is reformatted as Equation (3):

$$\hat{\tau} = K_{ti}^{T} ia_i + b_i, \tag{3}$$

where the term b_i represents the offset of the linear model for the respective joint i. The values of the parameters for $K_{ti}^{T} = [7.5, 6.8, 7.1, 2.8, 3.2, 3.4]$, and for $b_i = [-0.61, 16, 4.5, 0.84, 0.064, 0.099]$.

The UR5 robot is commonly utilized in cooperative tasks and robotics research projects. Additionally, there are toolboxes available for different platform software which facilitates programming and controlling the robot. Moreover, there exists a publicly accessible dataset specifically based on this robot, which has been utilized for exploring and studying robot accuracy degradation.

2.2. Discrete Wavelet Transform

The discrete wavelet transform (DWT) is a mathematical method that gives time-frequency information of a signal [39]. The signal is separated into different frequency bands by passing through a filtering process. The process involves convolving the signal of interest with a transform function known as the mother wavelet, which determines the parameters of the high-pass and low-pass filters. In simpler terms, DWT allows a signal to be separated into its frequency components, which makes it possible to identify frequency changes over time. The complete process consists of two stages; first, the calculation of the approximation and detail coefficients is made, and second, the signal at the required level of decomposition is reconstructed using the coefficients of the previous stage in a process known as inverse DWT.

Equation (4) defines the DWT of a discrete-time signal $x(k)$ with k samples, and ψ represents the discrete mother wavelet used for decomposition level i

$$DWT_{ik} = \sum x(k)\psi_{i,k}(t). \tag{4}$$

The selection of the wavelet mother family depends on the application; the most common families are Meyer, Haar, and Daubechies, among others. The Daubechies family is used in this work due to its characteristics of the higher number of vanishing moments [40], compact support in the frequency domain, and conserving the energy of the signal [41].

DWT algorithm consists of the following steps and is represented in Figure 2a,b:

- Define the wavelet mother function or wavelet basis and the decomposition level i to be used.
- The wavelet mother function provides the high-pass $g[n]$ and low-pass $h[m]$ filter coefficients.
- Perform the convolution operation between the signal $x[n]$ and the filters to separate the signal into high and low-frequency bands.
- Downsample by two filtered signals obtained in the previous step, keeping the even-indexed samples.
- The outputs of filtering and downsampling the signal provide the detail vector d_i and the approximation vector a_i of the high-pass and low-pass filtering, respectively, at each decomposition level i.
- Repeat the filtering and downsampling process using the previous approximation output as input for the next decomposition level until all levels are completed.
- To reconstruct the approximation and detail branches of the signal for a specific level i, upsample the corresponding coefficient vectors a_i and d_i by a factor of 2 (adding zeros in the positions of the removed samples in the downsampling process, if necessary).

- Apply the corresponding time-inverse high-pass $g'[n]$ or low-pass $h'[m]$ filter coefficients.
- The signal reconstruction $x'[n]$ is obtained by the inverse process using the last approximation vector a_i and all the detail vectors d_i, $i = 1, 2, \ldots, i$.

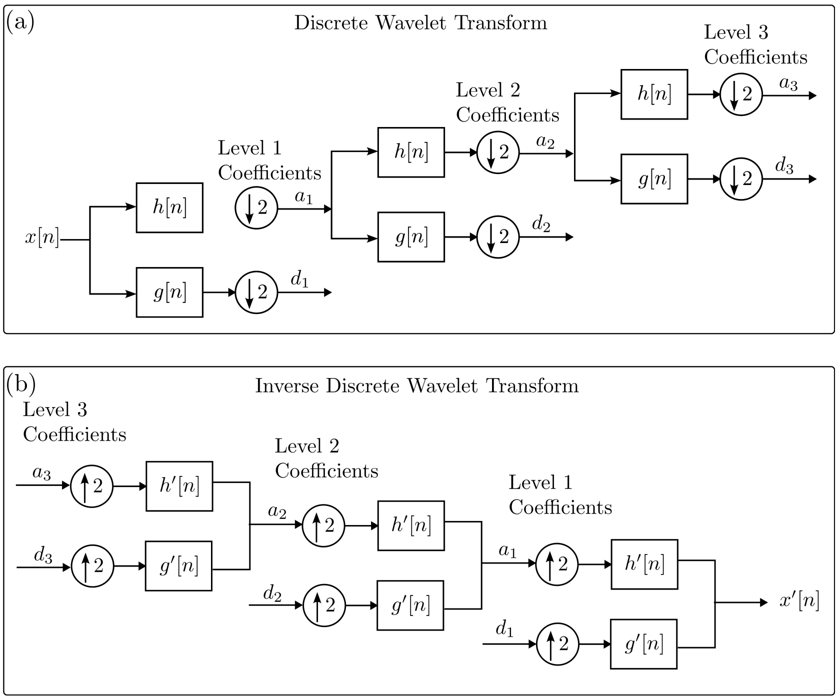


Figure 2. Block diagram of (**a**) DWT decomposition process of a signal $x[n]$. (**b**) Inverse DWT reconstruction diagram of a signal $x'[n]$.

DWT's recent applications include condition monitoring of slew bearings [42], evaluation of plant growth status [43], and wind power prediction [44].

Robot operating conditions in industrial environments involve different factors such as disturbances, noise, temperature, humidity, and others. The acquisition of robot signals in a noisy environment requires the use of signal processing techniques to handle the noise. In this sense, it is necessary to perform an analysis of the signal of interest to determine, depending on its characteristics, which type of techniques to use, such as high-pass or low-pass filtering. On the other hand, time-frequency analysis techniques such as DWT allow the separation of the original signal into different parts that are already filtered while preserving the original characteristics of the signal. DWT is used in this work because it allows for the decomposition of a signal into different frequency bands using a series of filters, which is useful when dealing with noisy signals. Each frequency band can be independently analyzed, compressing the signal information and reducing the computational burden for further analysis. Additionally, the DWT algorithm and its inverse can be converted to matrix operations, facilitating its implementation in FPGA-based digital systems.

2.3. Statistical Indicators

Statistical indicators are used to find characteristics that help to know the behavior of a dynamic system. For this, they are applied to the signals of interest, and through these, it is possible to determine if there are changes or variations in the system at different moments in time. In this sense, the statistical indicators proposed in this work are the root mean square (RMS) value and the variance.

The RMS and variance metrics are useful for identifying patterns or anomalies within a signal. On one hand, RMS provides information about the signal's amplitude or energy, while variance focuses on the signal's dispersion around the average amplitude.

Additionally, the calculation of both metrics is similar and can be easily implemented in an FPGA-based system. This allows for the optimization of the structure to perform the calculations of both metrics without the need for separate structures.

These indicators have been widely used in applications such as fault diagnosis in transformers [45] and bearings [46], among others.

The Root Mean Square (RMS) defined by Equation (5) value of a dataset or a discrete-time waveform is the square root of the arithmetic mean of the squares of the values

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}. \quad (5)$$

The variance measures the dispersion of a data set with respect to the mean and is defined by the Equation (6)

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2. \quad (6)$$

2.4. Artificial Neural Network

Artificial neural networks (ANN) emulate the computational processes performed by neurons in the human brain, using mathematical models. At the core of an ANN is the fundamental building block known as a simple perception and its structure is shown in Figure 3a.

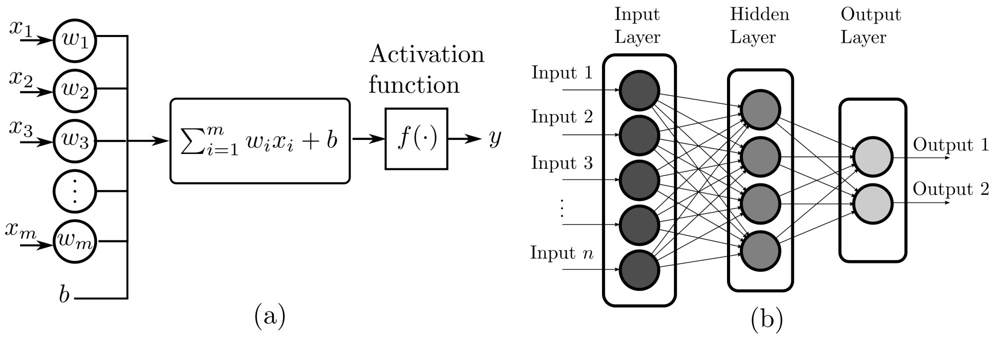


Figure 3. (**a**) General diagram of a simple perceptron. (**b**) Simplified diagram of an artificial neural network, the connections between layers represent a multi-layer perception network.

A simple perceptron is a mathematical unit that represents a neuron. It takes in a vector input x of size m and performs a series of operations. First, the input vector is multiplied by a weight vector w, and the products are then summed together with a bias coefficient b. The resulting value is then passed through an activation function $f(\cdot)$, such as binary step, linear, sigmoid, or hyperbolic tangent, to produce the output y. An Artificial Neural Network (ANN) consists of a collection of interconnected basic perceptions. The most common is the multi-layer perceptron (MLP), which is a network where all the neurons in the input, hidden, and output layers are interconnected. This configuration is shown in Figure 3b.

ANNs find utility in classification, learning, and pattern recognition tasks and are extensively employed across various scientific domains. For instance, some of their recent uses are the early diagnosis of breast cancer [47] and the combination with bio-inspired algorithms to improve the ANN training process [48].

Specifically, MLP possesses favorable attributes such as straightforward implementation, efficient training time, and the ability to generate high-quality models [49]. Additionally, the hardware implementation of MLP is relatively simple compared to other neural network architectures. Consequently, this work proposes the utilization of MLP based on these advantageous characteristics.

2.5. Proposed Methodology

The proposed methodology to determine if there is a deviation in the joints of a manipulator robot through the analysis of its motor currents is presented in a general way in Figure 4.

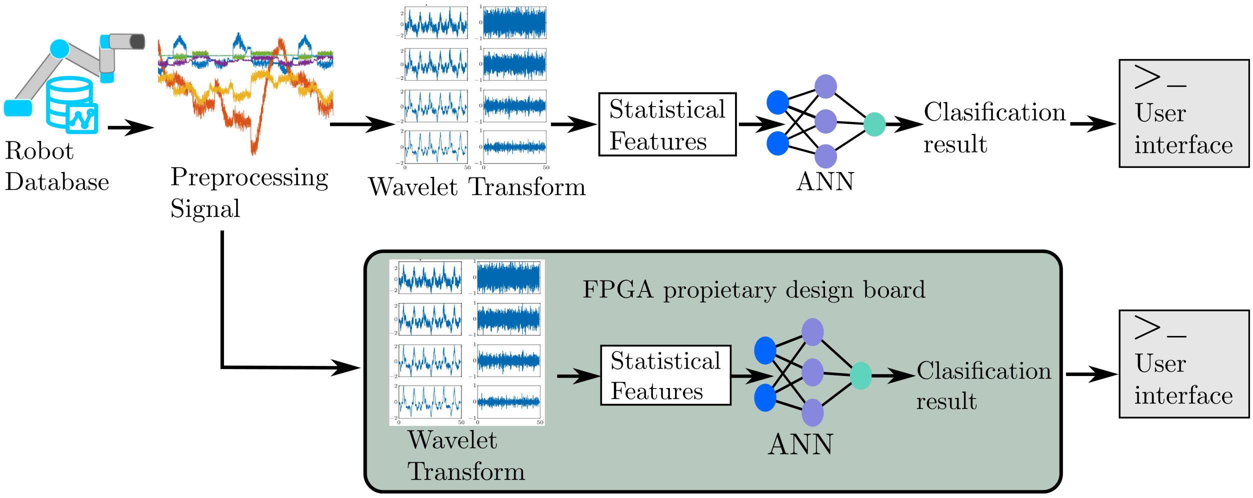


Figure 4. General diagram of the proposed methodology.

In this work, it is considered that the calibration conditions of the robot, as well as its electromechanical components, are in optimal operating conditions. However, robot calibration is a procedure that is performed with external sensors or with the tools provided by the manufacturers. In this sense, the implementation of robot calibration algorithms in FPGA involves not only the implementation of these but also the development and implementation of control or communication systems to operate the robot along with all subsystems that this entails, this aspect is subject to the limitation of the logical resources of the FPGA device and the resolution of the processing of fixed point operations and it is beyond the scope of this work.

The proposed method starts with the database of the robot to be analyzed; for this, a synthetic and a real-world dataset are used. The current signals of the robot motors are required when performing a repetitive trajectory or task for a prolonged time. Signals are acquired at two different moments: at the beginning of the trajectory or task and hours after the task's start. In this way, the dataset contains signals with different operating conditions, named cold operation (CO) for the initial state and hot operation (HO) for the second state. Figure 5a–f shows the real-world dataset motor current signals of joints one to six during the CO state.

In the pre-processing of the signals to provide robustness to the method, different noise levels are added to the signals in the dataset. These noisy signals are then employed in subsequent process stages as a dataset. The noise levels used are 5 dB, 10 dB, 15 dB, and 20 dB. Figure 6a,b show the current signal for CO and HO states from the real-world dataset, respectively, both with 5 dB of noise.

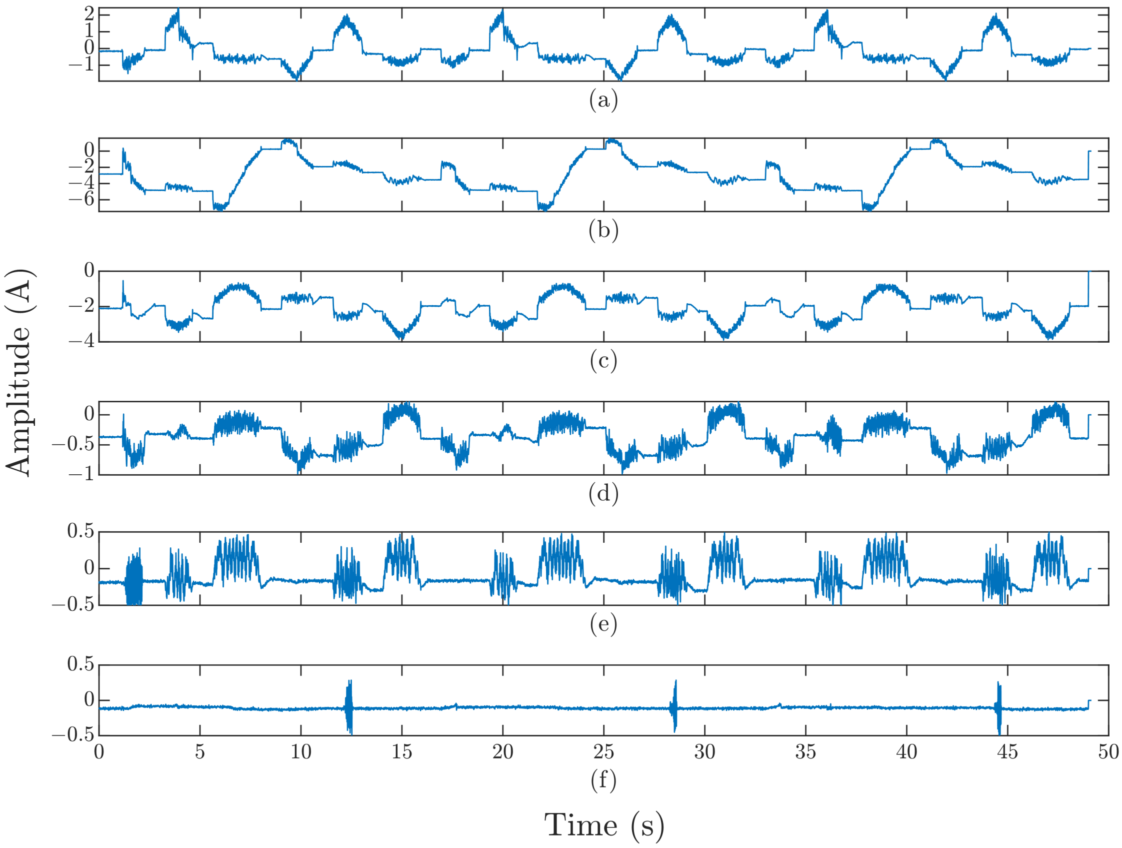


Figure 5. Real-world dataset current joints signal. Plots (**a**–**f**) show the signal for joints one to six.

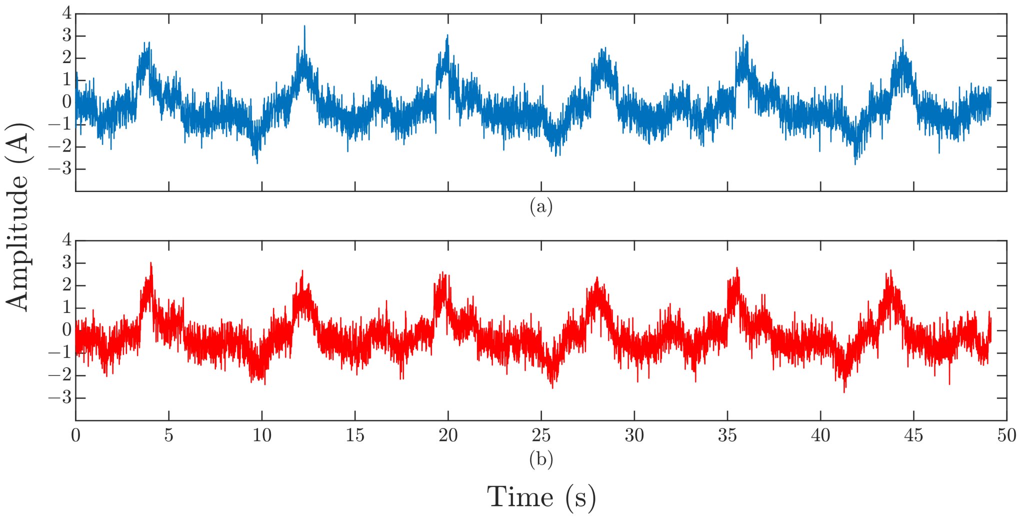


Figure 6. Real-world dataset joint one signal with 5 dB of noise. (**a**) CO state signal. (**b**) HO state signal.

Subsequently, the signals are divided into windows of 256 samples. In this point according to Figure 4, there are two branches; the first branch involves the implementation of the proposed method through software simulation, and its description is provided below. The second branch focuses on the hardware implementation of the method using an FPGA-based system, and this aspect is covered in the subsequent section. Since the DWT maintains the characteristics of the original signal through the different frequency bands, the windowed signals are processed with the DWT to find differences associated with the joint deviation of the robot. The wavelet mother Daubechies order six is used to perform the analysis. However, as shown in [18], only the first four levels are considered relevant for the analysis due to the amplitude and narrow frequency bands of the last two decomposition levels. Additionally, due to the FPGA system memory resources, and the DWT soft-core functionality, only levels three and four are used for the analysis. Figure 7 presents the DWT decomposition at level four for the joint one, with 5 dB noise added from the real-world dataset. Figure 7a,b represent the approximation and details of the CO state, respectively. On the other hand, Figure 7c,d display the results for the HO state.

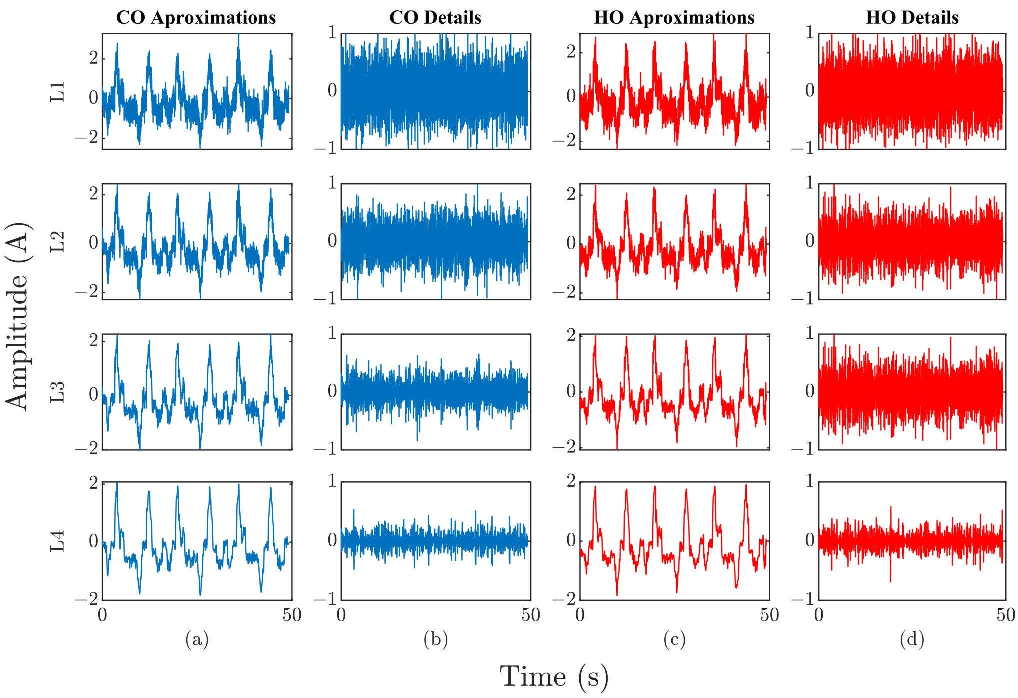


Figure 7. Real-world dataset joint one with 5 dB of noise DWT decomposition level 4. (**a**) CO state signal approximation components. (**b**) CO state signal details components. (**c**) HO state signal approximation components. (**d**) HO state signal details components.

Afterward, for the approximations and details of levels 3 and 4 of each signal, the RMS and variance indices are calculated; this process is repeated with each window that makes up the original signal. Subsequently, the indicators corresponding to each signal are averaged over the number of windows. This process is applied to each signal in the dataset, and the obtained indices are used for training, validation, and testing of a classifier based on ANN. To perform the complete robot analysis, the above process is replicated for each robot joint exploiting the FPGA's concurrent processing capability. In this case, for a UR5 robot, there are six classifiers based on ANN. In this way, each classifier keeps a simple configuration and the resource consumption for the hardware implementation of the whole process is kept below the capacity of the proprietary FPGA board.

Finally, the soft-cores for DWT, statistical features calculation, and neural network classifier are developed to validate the method implementation in an FPGA system, and the classifier results are displayed to the user in a command line terminal. The proposed methodology is tested in two synthetic and one real scenario for this work. For the synthetic scenario, the robot's database is obtained through the dynamic modeling of the robot and its simulation, performing a parametric trajectory as shown in Figure 8a.

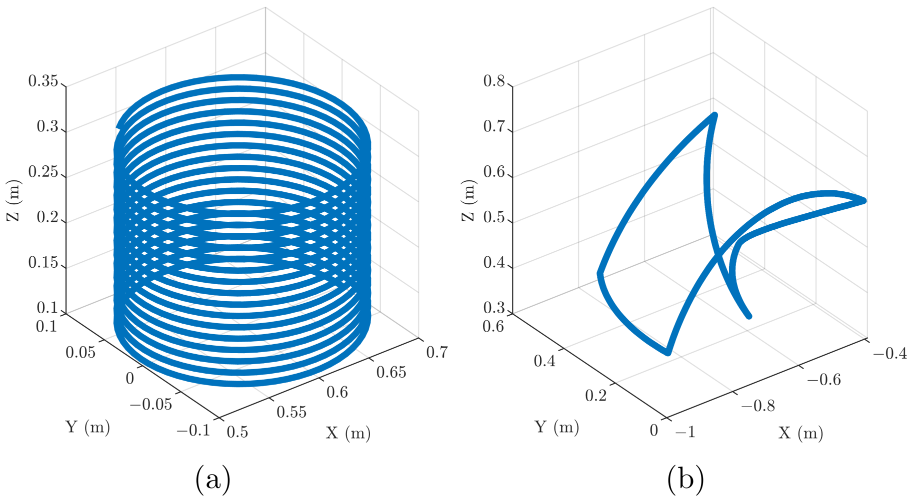


Figure 8. Trajectories followed by the robot. (**a**) Synthetic case. (**b**) Real case.

The trajectory parameters presented in Equation (7) are the radius of the helix $r = 0.1$ m, the final height $b = 0.025$ m, the frequency of the helix $f = 2$, and time $t = 8.1920$ s.

$$x = 0.6 + rcos(2\pi ft), y = rsin(2\pi ft), z = 0.1 + bt. \tag{7}$$

The sampling frequency of the UR5 robot is 125 Hz; therefore, with the above parameters, the synthetic signals of the simulated robot have a total of 1024 samples. In the synthetic scenario, one data set consists of the trajectory signals without deviation, which is the initial CO state. Six different data sets are generated for the HO state, and the deviation at each of the different joints of the robot is added. A sinusoidal signal is added to the robot's joint position signal obtained by following the helix trajectory to simulate the deviation. The sinusoidal signal has a randomly generated amplitude of 1 to 10% of the maximum amplitude value of the robot's joint position signal, a frequency of $f = 2$, and a time of $t = 8.1920$ s. This way, six HO datasets with 50 synthetic signals are obtained; each set contains random deviation levels in different joints. Each signal in the dataset generates ten additional signals for each noise level by adding white Gaussian noise (5 dB, 10 dB, 15 dB, and 20 dB), resulting in 2000 signals for each HO state dataset. For the CO dataset, one signal without deviation is processed similarly to obtain 500 signals per noise level. Combining the HO and CO datasets gives 4000 signals for the cases with joint deviation.

On the other hand, for the real case, a public-access dataset provided by the National Institute of Standards and Technology (NIST) is used [50]. Different speed levels test for the robot are provided in the dataset (50% and 100% speed). In this work, the dataset corresponding to the 100% speed of the robot is used. Like the synthetic case, the robot used is the UR5. The signals from the robot actuators were received at the controller level. The frequency was 125 Hz.

Figure 8b presents the trajectory used in this dataset. The robot continuously follows the trajectory with a weight of 4.5 lb on its end-effector for approximately 2 h. The CO status signals are captured when the robot has just started, and the HO status signals are obtained after 2 h of continuous operation. Three signals are provided for each robot's operation state: CO and HO, and the size of the signals was adjusted to obtain exact powers of 2 and to divide each signal into three parts of 2048 samples for a set of nine signals for each CO and HO state.

As was done with the synthetic dataset and to augment the dataset size, 50 new signals are generated from each of the nine original signals by applying white Gaussian noise at each noise level. This results in 3600 signals, with 1800 for each HO and CO data set. For each signal in the data set, the approximation and details of levels three and four are used to calculate the variance and RMS features, which means that each signal has eight features that are used as input to the neural network. So the datasets have a size of 4000×8

and 3600 × 8. For the neural network classifier, the datasets are split as follows: 70% for training, 15% for validation, and 15% for testing. The neural network comprises one input layer of size eight, a hidden layer of 16 neurons, and an output layer of size two. Besides, it is trained for 1000 epochs, using the Adam optimizer and the mean square error loss function. The activation function for the hidden and output layers is sigmoid.

With the dataset of the synthetic and real scenarios prepared, we proceed to describe the hardware implementation of the proposed method.

2.6. Positional Accuracy Degradation FPGA Implementation

Figure 9 shows a block diagram of the hardware implementation of the proposed methodology to determine if there is a deviation in the joints of the robot through the analysis of motor currents.

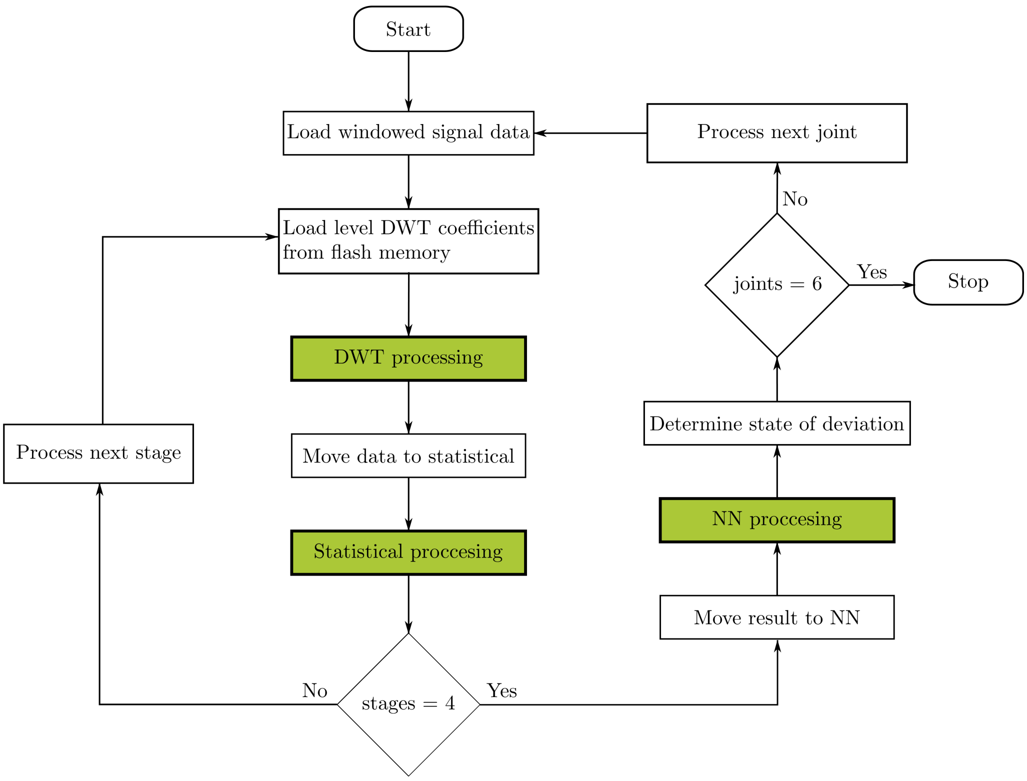


Figure 9. General scheme of the FPGA implementation of the proposed method.

Starting from the fact that there is a dataset of current signals from the robot's motors, as previously mentioned, the signal to be processed is divided into windows; these windowed signals are stored in the RAM of the processing card. The signal information corresponding to its first window is loaded into the FPGA. Then DWT coefficients corresponding to the first stage are loaded, and the DWT process is performed in the FPGA for that particular stage. Subsequently, information on the frequency band obtained in this process is moved to the calculation process of the RMS and variance statistical indicators. The indicators obtained are saved in the device's RAM at the end of the calculation. Later, the previously described procedure is repeated until the four stages, which correspond to the approximation and detail of decomposition levels three and four, are completed.

When the calculation and storage of the statistical indicators of the approximations and details of levels three and four have been completed, they are moved to the FPGA processing of the classifier based on neural networks, where the classification obtained with the network is stored in the memory of the device. Subsequently, the window signal for the next joint is loaded, and the entire process is repeated until all six articulations have been completed.

Figure 10 shows the general configuration of the system used to carry out the hardware implementation.

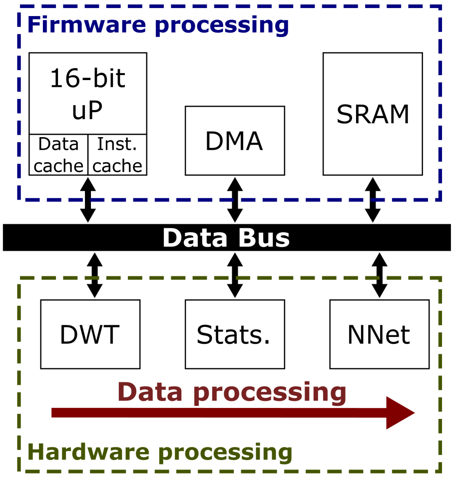


Figure 10. General diagram of the FPGA system.

The system comprises firmware processing containing 16-bit microprocessor modules, direct memory access (DMA), and Static Random Access Memory (SRAM). Whereas the hardware processing part comprises the DWT, statistical, and neural network models connected to the firmware processing through the data bus. Below are the modules corresponding to the main processes implemented in the FPGA, such as the DWT, the calculation of statistical indicators, and the neural network.

The proposed methodology was implemented in a digital system using a low-cost proprietary designed board that is based on an FPGA. This framework is specifically designed for in-situ online data processing, eliminating the need for external communication with other systems for data processing. The board is equipped with a low-cost Spartan 6 XC6SLX45 FPGA with a clock frequency of 48 MHz; it also features static random-access memory (RAM), communication ports such as the universal serial bus (USB), and universal asynchronous receiver transmitter (UART), power management, and flash memory. A 16-bit xQuP01v0 processor is embedded in the FPGA, besides an interconnection in-system bus (ISB), both proprietary designs; a broader description of the applications of the FPGA-based proprietary board can be found in [51,52]. The input data used in the system is scaled to an input range of $[-2, 2)$, and signal data are converted to a binary stream in 2.14 fixed-point format saved in onboard flash memory. As the process is executed, the microprocessor embedded in the FPGA uploads the data from the flash to the hardware calculation units and, when finished, reports the result via a USB to a user interface.

2.6.1. DWT Soft-Core FPGA Module

The DWT soft-core module calculates the decomposition of a signal in different approximation and detail frequency bands and performs the original signal reconstruction. This process is realized through a matrices multiplications. Figure 11 shows the module's architecture for the DWT process.

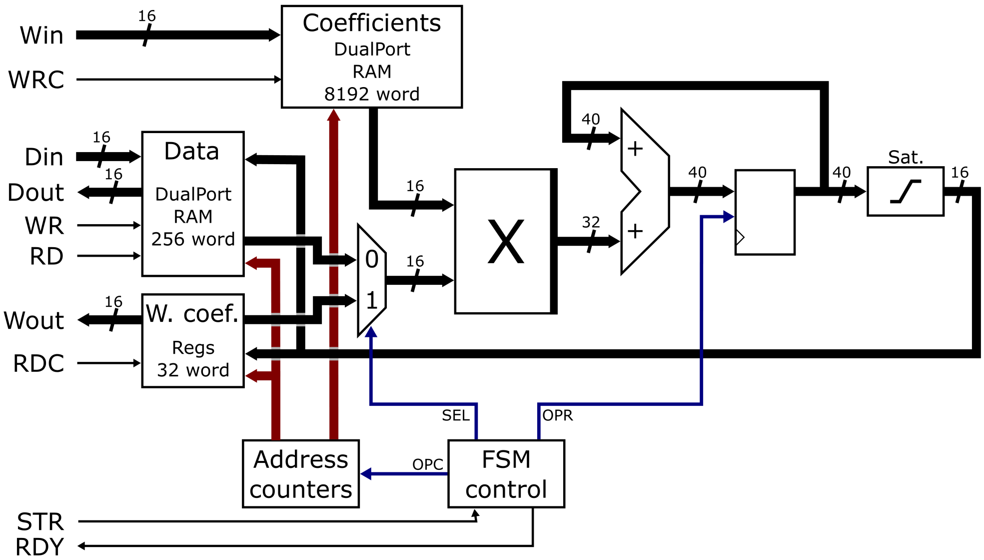


Figure 11. Architecture diagram of the DWT soft-core.

Within this module, the data corresponding to the windowed signal of 256 samples and the coefficient module are read from RAM. The coefficients module contains the wavelet matrices with wavelet filter values for approximations and details of levels three and four of the mother wavelet Daubechies order six. The "W. coef." module stores the wavelet coefficient vectors obtained by multiplying the wavelet matrix with the input signal. For reconstruction, the stored data in "W. coef." is read and multiplied by the transposed wavelet matrix corresponding to the respective level, approximation, or detail. These matrices are stored in "coefficients" in RAM. The result of the decomposition and reconstruction of the frequency bands is saturated at 16 bits and is stored in the "Data" and "W. coef." modules. Figure 11 depicts the bit size changes after each operation. A finite state machine (FSM) controls the signal decomposition and reconstruction process selection. It is done through the SEL signal and memory addresses of the coefficients, data, and wavelet coefficients modules. The address counters contain this information, and the FSM chooses the active module using the OPC signal. The STR signal indicates to FSM the beginning of the process, and once it is completed, the RDY signal is activated. To write and read memory, the modules use the WRC, WR, RD, and RDC signals. DWT module performs the process for the approximations and details of levels three and four of the signal stored in "Data". Once the process is finished, the reconstructed frequency bands are transferred to the statistical module for further processing.

2.6.2. Statistics Soft-Core FPGA Module

The Statistics soft-core module calculates a discrete input signal's RMS and variance features. The module comprises a single structure, and different indicators can be calculated by changing the data path. Figure 12a shows the general diagram of the architecture of the statistics module. It comprises three blocks: a first-input first-output memory storage block, the Datapath block where the index calculation is carried out, and an FSM block that conducts the operation of the process.

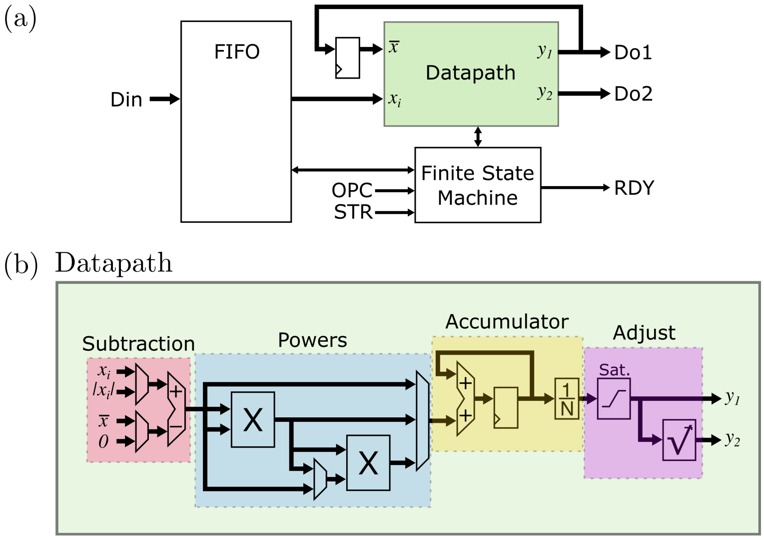


Figure 12. Architecture Statistics soft-core. (**a**) General core diagram. (**b**) Datapath module.

In the initial process, when the STR signal is activated in the FSM, the choice of the statistical indicator to be calculated is made with the OPC signal. At the end of the calculation, the RDY flag is activated. The input data to the module (Din) is stored in memory in the FIFO block to enter the Datapath block later, and the Do1 and Do2 outputs obtain the calculation of the index.

The process carried out within the datapath block is shown in a general way in Figure 12b; it consists of four sections: subtraction, powers, accumulation, and adjustment. A subtraction is performed between the sample x_i, or the absolute value of the sample $|x_i|$, and the mean of the input signal $\bar{x}$. Subsequently, in the powers section, the result obtained in the subtraction is raised to different powers, as the case may be, from the first to the fourth power. The accumulated value of the previous operation is stored in the accumulator and divided by the number of samples N. Finally, the result is saturated to adjust the required bit size and is output by y_1; if it is necessary to apply the square root, the output is made by y_2.

Figure 13a shows the module Datapath configured to calculate the variance for an input signal x; the connection is highlighted in red, and the operation's progress is observed in the different sections of the module.

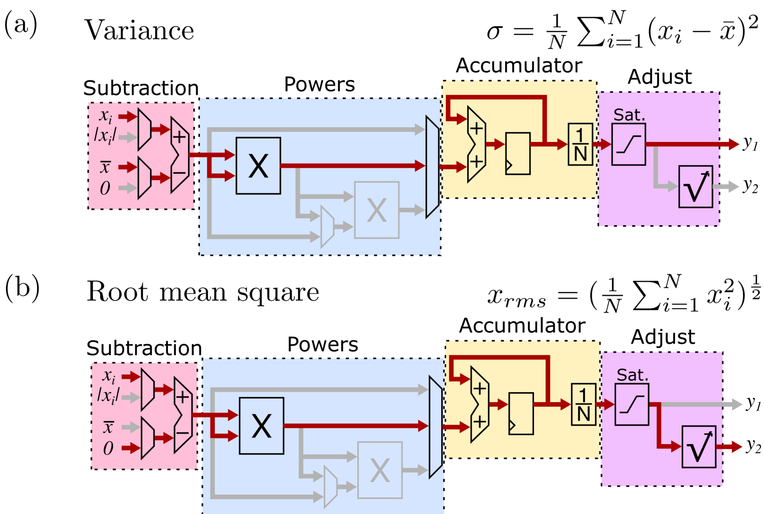


Figure 13. Statistical features modules. (**a**) Variance module operation path. (**b**) RMS module operation path.

First, the difference between the sample and the mean of the signal is calculated; second, it is squared in the powers section, the accumulation is divided by the number of samples, and after saturation, the indicator completed by the output y_1 is obtained. To calculate the RMS index, the process is presented in Figure 13b, wherein the subtraction section of the IPcore of the sample x_i is subtracted with a zero, which does not modify the value of the sample, then it is squared. In the powers section, the accumulated value is divided by the number of samples; at saturation output, the square root is calculated, and the result is delivered by y_2. When the indicators of the approximation and detail frequency bands of levels three and four have been calculated (eight indicators), they are moved to the neural network-based classifier module.

2.6.3. Neural Network Soft-Core FPGA Module

The neural network soft-core module performs the process of a trained network for classification; a description of its components is provided below. Figure 14a shows the general diagram of a simple perceptron.

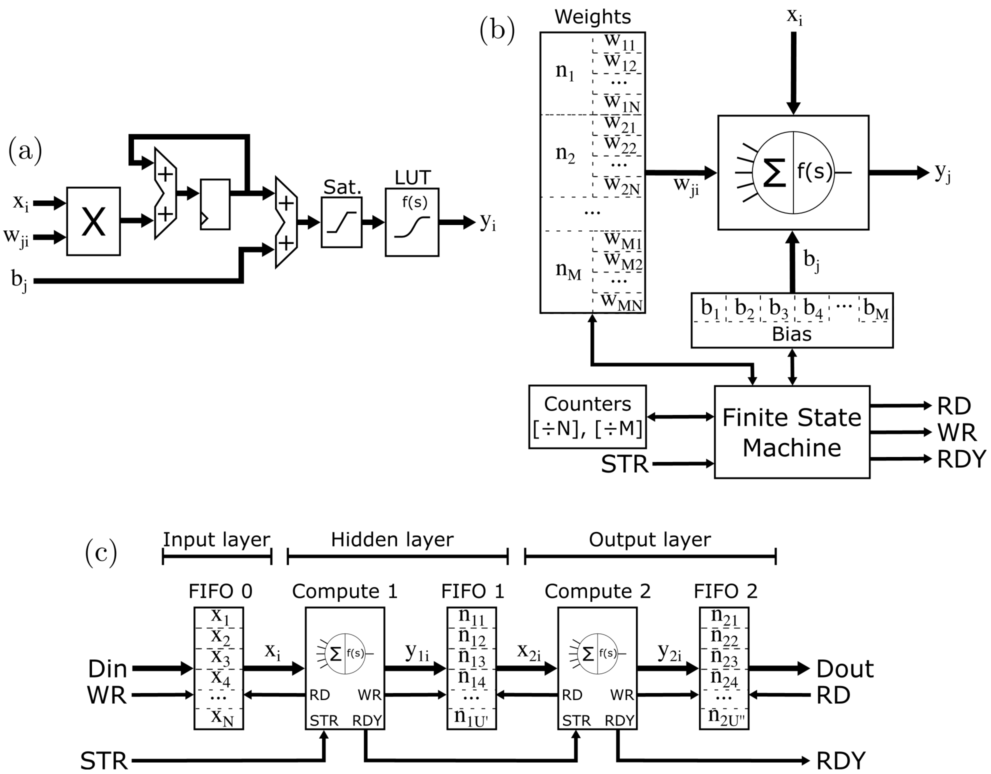


Figure 14. (**a**) Architecture diagram of a simple perceptron. (**b**) Diagram of neural layer core. (**c**) Neural network layer architecture diagram.

It is composed of a multiplier where input x_i is multiplied by the weights w_{ji}, where i represents the input number, and j is the weight number. A register accumulates the result, and the bias b_j is added. Afterward, the operation result is saturated to maintain the required bit size, and the activation function, contained in a look-up table (LUT), is continued. Finally, the output is delivered via y_i.

Figure 14b shows a general diagram of a set of perceptrons composing a neural network layer. This module comprises different blocks: a block of weights, a block of counters, FMS, bias, and an activation function. The block of weights contains the weights w_{MN} corresponding to neuron M and N number of neurons in the layer. Furthermore, the counters block counts the number of neurons and their corresponding weights. The bias block contains the biases for M neurons. Additionally, the activation function block performs the procedure described above in the perceptron, and the FMS directs the entire layer process. Signal STR indicates the start of the calculation, RD and WR are signals for reading and writing data in memory, and the RDY signal indicates that the calculation is complete. Figure 14c presents the general diagram of the classifier based on neural networks. In this network configuration, an input layer stores the data x_N that the network will evaluate in the FIFO 0 register through the Din input and the WR signal to control write enable in memory. The process starts when the STR signal is high, and the x_i samples enter the structure presented in Figure 14b through the RD signal. Moreover, the hidden layer's output is stored in FIFO 1 register by the signal WR. Once all the samples are complete, the RDY signal goes high and activates the next layer. The output layer of the network receives the data stored in FIFO 1 and performs the same process as mentioned in the hidden layer. The layer's output is stored in the FIFO 2 register as the input data is processed. These are accessible when the output layer RDY signal is high and are readable by Dout and the RD control signal. Weights and biases of the hidden and output layers and the activation functions are stored in the device's RAM. Neural network configuration consists of an input layer with eight inputs, a hidden layer of 16 neurons, and an output layer with two outputs; the activation functions are sigmoid. Neural network training was carried out in software independently, and the values of weights and bias were stored for use in the FPGA device.

3. Results and Discussion

The results of the proposed methodology to determine joint deviation in the robot using motor current analysis are presented below. The results are shown in the form of confusion matrices, and a summary in tabular form is provided to help interpret the findings. The confusion matrix chart has the correctly classified observations on its diagonal, while the cells outside the diagonal show the incorrect observations. In both cases, the percentage is shown. The last cell on the diagonal shows the average accuracy. Moving to the last column, the first two cells specifically present the percentages of all samples that were correctly and incorrectly classified for each class. These values correspond to precision and false discovery rate, respectively. Lastly, focusing on the last row, the two cells highlight the percentage of samples belonging to each class that were correctly and incorrectly classified. These values are commonly referred to as recall and false negative rate, respectively. In summary, the last cell of the confusion matrix diagonal shows the average number of samples that the ANN correctly and incorrectly classified, providing an overall accuracy measure for the classifier, this information for the real and synthetic cases is presented in Table 4.

Table 4. Average accuracy obtained in the ANN validation process for the synthetic and real cases.

Accuracy	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Synthetic case	82.7%	98.2%	97.7%	89.8%	100%	82.7%
Real case	93.5%	66.5%	78.8%	88.7%	100%	100%

To evaluate the operation of the proposed method, a computer software simulation of the different techniques that make up the methodology was carried out. Additionally, to evaluate the classifiers 600 samples from each dataset were used for each corresponding joint; these samples account for 15% of the dataset for each of the six scenarios in the synthetic case. These samples were then separated from the original dataset and are treated as unknown samples for the ANN classifiers.

Figure 15 shows the confusion matrices obtained from the classifiers based on neural networks for the synthetic scenario and a summarized version is presented in Table 4.

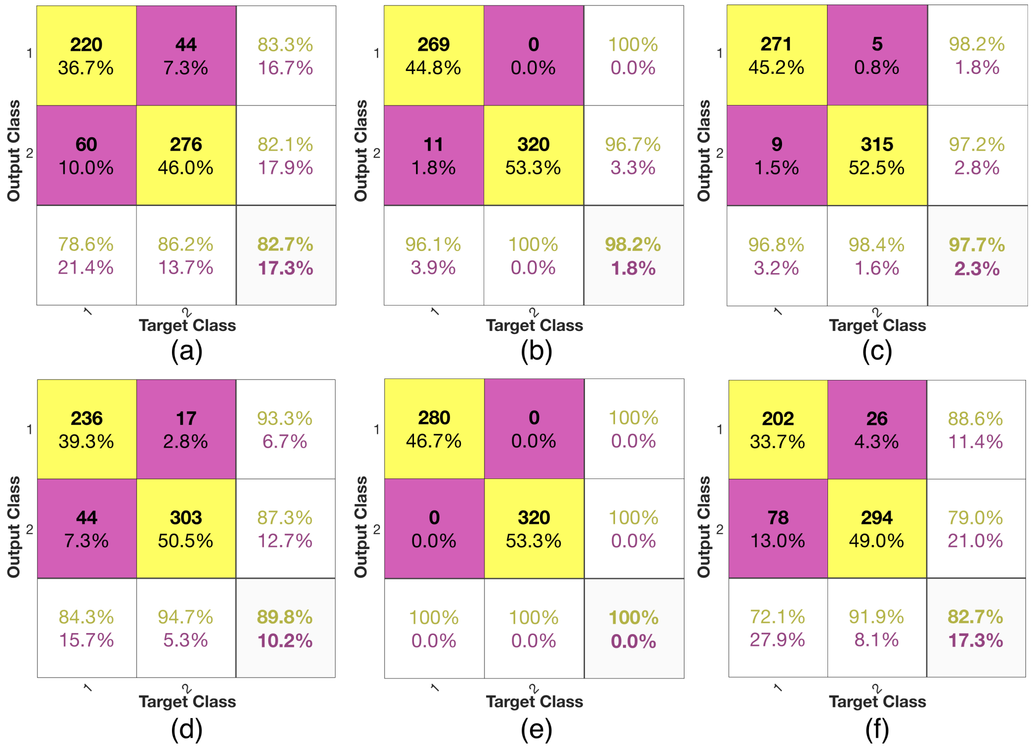


Figure 15. Confusion matrices obtained in the ANN validation process for the synthetic cases where the deviation is induced in one of the robot joints at a time. (**a**–**c**) confusion matrices correspond to joints one to three. (**d**–**f**) confusion matrices correspond to joints four to six.

Figure 15a–c corresponds to the cases of joints one to three, while Figure 15d–f corresponds to joints four to six. In this sense, it is observed that the classifiers can differentiate between the states of operation when there is a deviation in the joints with a minimum average accuracy of 82.7% corresponding to joints one and six, while joint four has an average value of 89.8%, joints two and three have an average accuracy of around 97% and joint five has an average accuracy of 100%. It should be noted that for the synthetic scenario, each case includes a deviation in the respective joint. For instance, in case one, only joint one contains a deviation, while the other five joints remain unaffected. In this regard, joints two and three are well differentiated by their respective classifiers; this can be attributed to the fact that in the trajectory carried out, these joints have a more significant intervention than the rest.

In addition, in the motion equations of the robot, these joints are closely related; therefore, since this test is synthetic, the deviation of one joint can affect the behavior of the other. Regarding joint five, which presents 100% accuracy, this is attributable to its minimal intervention in the trajectory; therefore, by inducing deviation in the joint, the change between both operating states is differentiable by the proposed methodology. For joints, one, four, and six, the difference between their accuracy levels and those of the joints with the best ranking result can be inflated by their level of intervention in the trajectory and the fact that the synthetically induced deviation has a random amplitude.

On the other hand, for the real case, 540 samples were used to test the classifier. As in the synthetic case, these samples also represent 15% of the respective datasets used for training the classifiers and the samples are unknown to the classifiers. The confusion matrices obtained are presented in Figure 16, where Figure 16a–c correspond to articulations one to three and Figure 16d–f correspond to joints four through six and a summarized version is presented in Table 4.

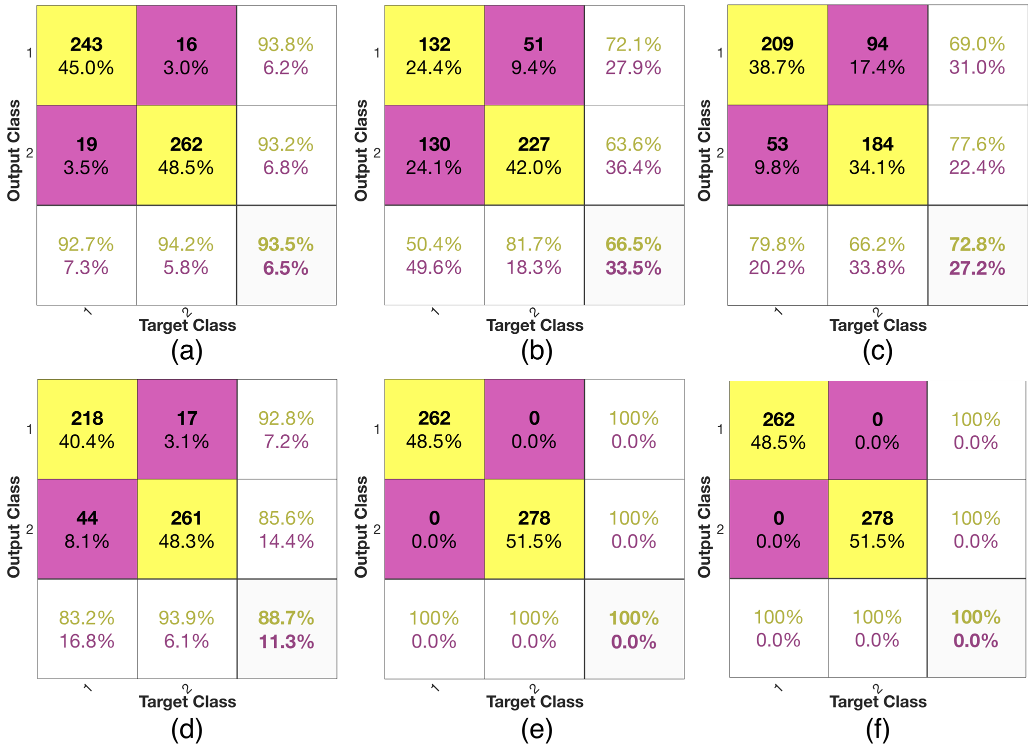


Figure 16. Confusion matrices obtained in the ANN validation process for the real case where the deviation at joint five is known. (**a**–**c**) confusion matrices correspond to joints one to three. (**d**–**f**) confusion matrices correspond to joints four to six.

Figure 16 shows that for joints two and three, an average accuracy of 66.5% and 72.8% is obtained, respectively. Whereas for joints one and four, an average accuracy greater than 88% is obtained.

Furthermore, joints five and six present an average accuracy of 100%. These results are obtained from a real-world dataset; however, the dataset does not provide information on the condition of the robot at the time of obtaining the signals, so it is not possible to determine if there is a mechanical, electrical, or wear failure that has caused a joint failure deviation in the robot. In this sense, the results obtained for joints two and three indicate that the signals of the robot CO and HO's operating states are not different. Additionally, the other joints have results greater than 90% accuracy, for which the classifier can distinguish the two states of operation and indicate a deviation. Emphasize that this dataset has a time difference of 2 h between the initial and final operating states, in addition to the fact that this robot performs a trajectory continuously with a mass of 4.5 lb and operates at 100% of its maximum speed, for which these results could be attributed to the additional inertia caused by the extra weight and speed of operation. The results obtained in the synthetic case and in the real case show that the proposed method can process signals with different noise levels using the DWT, using the DWT, which allows the noisy signal to be decomposed into different frequency bands without losing the characteristics of the original signal, thus enabling the utilization of the frequency band of interest.

Regarding the results of the implementation in the physical system, validation is made that the calculations made in the FPGA device correspond to those obtained in the software simulation. In this sense, Table 5 shows the performance obtained in the main modules used to implement the methodology on hardware. The information in the Table 5 consists of the clock cycles necessary to calculate the module and complete processing of the module (calculation and copy of data to the buffers), as well as the calculation time and complete processing of each module. It is observed that the module that requires less time to perform complete processing is the neural network, while the DWT is the module with the most extended duration. The DWT module decomposes and reconstructs the window of the signal to be analyzed; the module's error for calculating the approximation levels is 0.006%, while for the details, it is 0.02%. The complete system implemented in FPGA includes the DWT, statistical, and neural network modules, the embedded microprocessor, internal data bus, and device drivers such as flash memory, USB, RAM, bootloader, debugger, and programmer, among others.

Table 5. Module performance in hardware implementation.

Soft-Core	Complete Processing (Clock Cycles)	Calculation Processing (Clock Cycles)	Complete Processing (Time)	Calculation Processing (Time)
DWT	265,577	178,280	5.53 ms	3.71 ms
Statistical	7970	5400	166 µs	112.5 µs
Neural Network	3940	3590	82 µs	78.8 µs

Table 6 presents the resources of the FPGA device used for the implementation; it should be noted that this is a low-cost device and that, with the resources used, it is capable of calculating the entire process in less than a second. The resources are referred to in the "logic utilization" column. Specifically, the columns "Used", "Available", and "Utilization" indicate the total number of elements implemented, available elements, and the percentage of elements used, respectively. In general, according to the information in Table 6, it is possible to say that the device used half of its resources for this implementation.

Table 6. System resource utilization.

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	10,090	54,576	18%
Number of Slice LUTs	12,521	27,288	45%
Number of fully used LUT-FF	4662	17,949	25%
Number of bonded IOBs	128	218	58%
Number of Block RAM/FIFO	59	116	50%
Number of BUFG/BUFGCTRLs	1	16	6%
Number of DSP48A1s	32	58	55%

The hardware-software execution time for the complete system is 564,720 ms. Before the neural network classifier, the average error between the software (e.g., Matlab) and the FPGA calculation for the modules before the classifier amounts to 0.87%. However, the accuracy obtained by the neural network module reaches 83% compared to the accuracy obtained in the software implementation. It should be noted that the accuracy obtained by the neural network and the average error are attributable to the fixed point representation. The results obtained from the hardware implementation validate the use of the proposed method to detect deviations in the robot's joints through the analysis of the motor currents.

Regarding the results obtained in this work, it should be noted that specifically for joint five of the real case, the results coincide with those reported in [53], where the same dataset was used, and a manual analysis of velocity and position graphs of the joint was conducted without a systematic approach. In another study [17] MRA and DLSTM are used to obtain the residual error of the end-effector of the robot. However, it is not determined in which joint exhibited deviation causing the error in the end-effector. On the other hand in [18] an expert system based on two neural networks, dimension reduction, and energy and fractal indicators was presented. This system aimed to accurately detect deviations in the robot's joints. However, implementing the fractal approach in FPGA was found to be more complex than the statistical indicators proposed in this current study. It's worth noting that the aforementioned studies were performed using software simulation and offline analysis, which didn't pose resource limitations or fixed-point operation issues, in contrast with an FPGA system. The method proposed identifies in which joint of the robot there is deviation, uses techniques and algorithms that are easy to implement on FPGA systems, and employs a low-cost self-designed board that operates independently without the need for external systems which allows for systematic in-situ analysis.

4. Conclusions

The importance of robotic systems in modern manufacturing processes requires high accuracy and precision to perform tasks with high-quality standards, such as painting, welding, and machining. Achieving this level of precision is critical for the robot's performance, but various factors can degrade it, such as wear and failure of mechanical and electrical components or environmental conditions. In this sense, a methodology for the detection of joint deviation of the robot through the current analysis of the actuators and its implementation in a digital system based on an FPGA is introduced in this work. Synthetic signals, both with and without deviation, are generated using the dynamic model of a UR5 robot. Real-world data from a UR5 robot validates the methodology. To enhance the method's robustness, the signals are contaminated with different noise levels. Then the signals are processed using DWT to separate them into approximation and detail frequency bands, and statistical indicators such as RMS and variance are calculated for each band. Finally, the indicators are used to train and validate a classifier based on neural networks. From the results is concluded that:

- The proposed method is able to work with signals containing noise due to the use of the DWT to decompose the signal into different frequency bands, which allows an analysis of the signals with a smaller amount of information than the original signal. Its implementation in hardware allows the decomposition and reconstruction of the signal with a single module, which allows better management of FPGA resources.
- The statistical indicators used have proven to be effective in the detection of non-visible features in the analyzed signals, they are simple and their hardware implementation has been carried out through a module with a versatile architecture that allows the calculation of other statistical indicators if necessary.
- The classifier, based on an MLP network, showed its effectiveness in differentiating between the different operating states of the signals provided by the statistical indicators. In this sense, the classifier structure is simple with a reduced number of neurons in the hidden layer, which makes its implementation in hardware not complex.
- For the particular case of joint five of the real case, the result is in line with the findings reported in [53], which also highlights the deviation in that joint.
- The designed modules and architectures have been implemented in a low-cost proprietary FPGA-based digital system and have demonstrated the feasibility of the proposed method using about 50% of the resources. In addition to achieving a high processing speed for all joints (less than one second) with minimal error attributed to truncation and fixed point operations. Moreover, the use of an FPGA system allows for concurrent execution and reconfiguration of the device; if an additional module is required, it can be incorporated into the system.

Future work proposes the hardware implementation of a method capable of performing the analysis of the positional degradation of the robot using only a classifier based on neural networks, as well as analyzing the classifier's behavior when multiple joints deviate simultaneously. Furthermore, the proposal includes the exploration of other non-linear indicators, signal processing techniques, and methodologies based on machine learning and deep learning.

Author Contributions: Conceptualization, E.G.-U., R.A.O.-R. and L.M.-V.; methodology, E.G.-U., R.A.O.-R. and L.M.-V.; software, validation, and formal analysis, E.G.-U. and L.M.-V.; investigation, E.G.-U., R.A.O.-R. and L.M.-V.; resources, E.G.-U., R.A.O.-R. and L.M.-V.; writing—original draft preparation, E.G.-U., R.A.O.-R. and L.M.-V.; writing—review and editing, E.G.-U., R.A.O.-R. and L.M.-V.; visualization, E.G.-U. and R.A.O.-R.; supervision, R.A.O.-R. and L.M.-V.; project administration, R.A.O.-R. and L.M.-V.; funding acquisition, R.A.O.-R. and L.M.-V. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Consejo Nacional de Ciencia y Tecnología (CONACYT) under scholarship 783320 and grant FONDEC-UAQ-2021-LMV-6829.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial Neural Network
BUFG	Global Buffer
BUFGCTRLs	Global Clock Control Buffer
CO	Cold Operation
Din	Data input
DMA	Direct Memory Access
Do1	Data output 1
Do2	Data output 2
Dout	Data output
DSP48A1s	Digital Signal Processing element included on certain FPGAs
DWT	Discrete Wavelet Transform
FIFO	First-input First-output
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
HO	Hot Operation
IOBs	Input Output Blocks
IPcore	Intellectual Propriety Core
ISB	In-System Bus
LUT	Look-Up Table
LUT-FF	Look-Up Table-Flip-Flop
MLP	Multi-layer Perceptron
OPC	Option Control signal
OPR	Option Register signal
RAM	Random Access Memory
RD	Read signal
RDC	Read Coefficient signal
RDY	Ready signal
RMS	Root Mean Squared
SEL	Select signal
SRAM	Static Random Access Memory
STR	Start signal
UART	Universal Asynchronous Receiver-Transmitter
Win	Window input
Wout	Window output
WR	Write signal
WRC	Write Coefficient signal

References

1. Kelaiaia, R.; Chemori, A.; Brahmia, A.; Kerboua, A.; Zaatri, A.; Company, O. Optimal dimensional design of parallel manipulators with an illustrative case study: A review. *Mech. Mach. Theory* **2023**, *188*, 105390. [CrossRef]
2. Brahmia, A.; Kelaiaia, R.; Company, O.; Chemori, A. Kinematic sensitivity analysis of manipulators using a novel dimensionless index. *Robot. Auton. Syst.* **2022**, *150*, 104021. [CrossRef]
3. Qiao, G.; Weiss, B.A. Industrial Robot Accuracy Degradation Monitoring and Quick Health Assessment. *J. Manuf. Sci. Eng.* **2019**, *141*, 071006. [CrossRef]

4. Zhang, Y.; An, H.; Ding, X.; Liang, W.; Yuan, M.; Ji, C.; Tan, J. Industrial Robot Rotate Vector Reducer Fault Detection Based on Hidden Markov Models. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6–8 December 2019; pp. 3013–3018. [CrossRef]
5. Wescoat, E.; Kerner, S.; Mears, L. A comparative study of different algorithms using contrived failure data to detect robot anomalies. *Procedia Comput. Sci.* **2022**, *200*, 669–678. [CrossRef]
6. Truc, L.N.; Quang, N.P.; Quang, N.H. Impact analysis of actuator torque degradation on the IRB 120 robot performance using simscape-based model. *Int. J. Electr. Comput. Eng. (IJECE)* **2021**, *11*, 4850. [CrossRef]
7. Rohan, A. Deep Scattering Spectrum Germaneness for Fault Detection and Diagnosis for Component-Level Prognostics and Health Management (PHM). *Sensors* **2022**, *22*, 9064. [CrossRef]
8. Raviola, A.; Martin, A.D.; Guida, R.; Jacazio, G.; Mauro, S.; Sorli, M. Harmonic Drive Gear Failures in Industrial Robots Applications: An Overview. In Proceedings of the PHM Society European Conference, Virtual Event, 28 June–2 July 2021; Volume 6.
9. Lo, C.C.; Lee, C.H.; Huang, W.C. Prognosis of Bearing and Gear Wears Using Convolutional Neural Network with Hybrid Loss Function. *Sensors* **2020**, *20*, 3539. [CrossRef]
10. Lu, K.; Chen, C.; Wang, T.; Cheng, L.; Qin, J. Fault diagnosis of industrial robot based on dual-module attention convolutional neural network. *Auton. Intell. Syst.* **2022**, *2*, 12. [CrossRef]
11. Rohan, A.; Raouf, I.; Kim, H.S. Rotate Vector (RV) Reducer Fault Detection and Diagnosis System: Towards Component Level Prognostics and Health Management (PHM). *Sensors* **2020**, *20*, 6845. [CrossRef]
12. Long, J.; Mou, J.; Zhang, L.; Zhang, S.; Li, C. Attitude data-based deep hybrid learning architecture for intelligent fault diagnosis of multi-joint industrial robots. *J. Manuf. Syst.* **2021**, *61*, 736–745. [CrossRef]
13. Zhou, X.; Zhou, H.; He, Y.; Huang, S.; Zhu, Z.; Chen, J. Harmonic reducer in-situ fault diagnosis for industrial robots based on deep learning. *Sci. China Technol. Sci.* **2022**, *65*, 2116–2126. [CrossRef]
14. Pan, J.; Qu, L.; Peng, K. Sensor and actuator fault diagnosis for robot joint based on deep CNN. *Entropy* **2021**, *23*, 751. [CrossRef] [PubMed]
15. Pan, J.; Qu, L.; Peng, K. Deep residual neural-network-based robot joint fault diagnosis method. *Sci. Rep.* **2022**, *12*, 17158. [CrossRef]
16. Jiao, J.; Zheng, X.J. Fault diagnosis method for industrial robots based on DBN joint information fusion technology. *Comput. Intell. Neurosci.* **2022**, *2022*. [CrossRef] [PubMed]
17. Taha, H.A.; Yacout, S.; Birglen, L. Detection and monitoring for anomalies and degradation of a robotic arm using machine learning. In *Advances in Automotive Production Technology–Theory and Application*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 230–237. [CrossRef]
18. Galan-Uribe, E.; Amezquita-Sanchez, J.P.; Morales-Velazquez, L. Supervised Machine-Learning Methodology for Industrial Robot Positional Health Using Artificial Neural Networks, Discrete Wavelet Transform, and Nonlinear Indicators. *Sensors* **2023**, *23*, 3213. [CrossRef]
19. Qiao, G.; Weiss, B.A. Quick health assessment for industrial robot health degradation and the supporting advanced sensing development. *J. Manuf. Syst.* **2018**, *48*, 51–59. [CrossRef]
20. Qiao, G.; Garner, J. Advanced Sensing Development to Support Accuracy Assessment for Industrial Robot Systems. In Proceedings of the ASME 2020 15th International Manufacturing Science and Engineering Conference, Virtual Online, 3 September 2020. [CrossRef]
21. Qiao, G. Advanced Sensing Development to Support Robot Accuracy Assessment and Improvement. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 917–922. [CrossRef]
22. Izagirre, U.; Andonegui, I.; Eciolaza, L.; Zurutuza, U. Towards manufacturing robotics accuracy degradation assessment: A vision-based data-driven implementation. *Robot. Comput.-Integr. Manuf.* **2021**, *67*, 102029. [CrossRef]
23. Liu, Y.; Li, Y.; Zhuang, Z.; Song, T. Improvement of Robot Accuracy with an Optical Tracking System. *Sensors* **2020**, *20*, 6341. [CrossRef]
24. Sun, Z.; Xu, Y.; Ma, Z.; Xu, J.; Zhang, T.; Xu, M.; Mei, X. Field Programmable Gate Array Based Torque Predictive Control for Permanent Magnet Servo Motors. *Micromachines* **2022**, *13*, 1055. [CrossRef]
25. Liu, W.; Zhao, C.; Liu, Y.; Wang, H.; Zhao, W.; Zhang, H. Sim2real kinematics modeling of industrial robots based on FPGA-acceleration. *Robot. Comput.-Integr. Manuf.* **2022**, *77*, 102350. [CrossRef]
26. Cabrera-Rufino, M.A.; Ramos-Arreguín, J.M.; Rodríguez-Reséndiz, J.; Gorrostieta-Hurtado, E.; Aceves-Fernandez, M.A. Implementation of ANN-Based Auto-Adjustable for a Pneumatic Servo System Embedded on FPGA. *Micromachines* **2022**, *13*, 890. [CrossRef]
27. Liu, C.C.; Lee, T.T.; Xiao, S.R.; Lin, Y.C.; Lin, Y.Y.; Wong, C.C. Real-time FPGA-based balance control method for a humanoid robot pushed by external forces. *Appl. Sci.* **2020**, *10*, 2699. [CrossRef]
28. Zhang, C.; Chen, S.; Zhao, L.; Li, X.; Ma, X. FPGA-Based Linear Detection Algorithm of an Underground Inspection Robot. *Algorithms* **2021**, *14*, 284. [CrossRef]
29. Romanov, A.M.; Romanov, M.P.; Manko, S.V.; Volkova, M.A.; Chiu, W.Y.; Ma, H.P.; Chiu, K.Y. Modular reconfigurable robot distributed computing system for tracking multiple objects. *IEEE Syst. J.* **2020**, *15*, 802–813. [CrossRef]

30. Plancher, B.; Neuman, S.M.; Bourgeat, T.; Kuindersma, S.; Devadas, S.; Reddi, V.J. Accelerating robot dynamics gradients on a cpu, gpu, and fpga. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2335–2342. [CrossRef]
31. Żabiński, T.; Hajduk, Z.; Kluska, J.; Gniewek, L. FPGA-Embedded Anomaly Detection System for Milling Process. *IEEE Access* **2021**, *9*, 124059–124069. [CrossRef]
32. Huang, C.H.; Chen, P.J.; Lin, Y.J.; Chen, B.W.; Zheng, J.X. A robot-based intelligent management design for agricultural cyber-physical systems. *Comput. Electron. Agric.* **2021**, *181*, 105967. [CrossRef]
33. Cañas, J.M.; Fernández-Conde, J.; Vega, J.; Ordóñez, J. Reconfigurable computing for reactive robotics using open-source fpgas. *Electronics* **2021**, *11*, 8. [CrossRef]
34. Pan, J.; Luan, F.; Gao, Y.; Wei, Y. FPGA-based implementation of stochastic configuration network for robotic grasping recognition. *IEEE Access* **2020**, *8*, 139966–139973. [CrossRef]
35. Corke, P.; Haviland, J. Not your grandmother's toolbox–the Robotics Toolbox reinvented for Python. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 11357–11363.
36. Kebria, P.M.; Al-Wais, S.; Abdi, H.; Nahavandi, S. Kinematic and dynamic modelling of UR5 manipulator. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; pp. 004229–004234.
37. Porcelli, G. Dynamic Parameters Identification of a UR5 Robot Manipulator. Ph.D. Thesis, Politecnico di Torino, Turin, Italy, 2020.
38. Kufieta, K.; Gravdahl, J.T. Force Estimation in Robotic Manipulators: Modeling, Simulation and Experiments. The UR5 Manipulator as a Case Study. Ph.D. Thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, 2014.
39. Sundararajan, D. The Haar Discrete Wavelet Transform. In *Discrete Wavelet Transform: A Signal Processing Approach*; John Wiley & Sons, Singapore Pte. Ltd.: Singapore, 2015; pp. 97–130. [CrossRef]
40. Guillén-García, E.; Morales-Velazquez, L.; Zorita-Lamadrid, A.L.; Duque-Perez, O.; Osornio-Rios, R.A.; Romero-Troncoso, R.d.J. Accurate identification and characterisation of transient phenomena using wavelet transform and mathematical morphology. *IET Gener. Transm. Distrib.* **2019**, *13*, 4021–4028. [CrossRef]
41. Hernández, J.C.; Antonino-Daviu, J.; Martínez-Giménez, F.; Peris, A. Comparison of different wavelet families for broken bar detection in induction motors. In Proceedings of the 2015 IEEE International Conference on Industrial Technology (ICIT), Seville, Spain, 17–19 March 2015; pp. 3220–3225. [CrossRef]
42. Caesarendra, W.; Tjahjowidodo, T. A Review of Feature Extraction Methods in Vibration-Based Condition Monitoring and Its Application for Degradation Trend Estimation of Low-Speed Slew Bearing. *Machines* **2017**, *5*, 21. [CrossRef]
43. Li, F.; Wang, L.; Liu, J.; Wang, Y.; Chang, Q. Evaluation of Leaf N Concentration in Winter Wheat Based on Discrete Wavelet Transform Analysis. *Remote Sens.* **2019**, *11*, 1331. [CrossRef]
44. Liu, Y.; Guan, L.; Hou, C.; Han, H.; Liu, Z.; Sun, Y.; Zheng, M. Wind Power Short-Term Prediction Based on LSTM and Discrete Wavelet Transform. *Appl. Sci.* **2019**, *9*, 1108. [CrossRef]
45. Huerta-Rosales, J.R.; Granados-Lieberman, D.; Garcia-Perez, A.; Camarena-Martinez, D.; Amezquita-Sanchez, J.P.; Valtierra-Rodriguez, M. Short-circuited turn fault diagnosis in transformers by using vibration signals, statistical time features, and support vector machines on FPGA. *Sensors* **2021**, *21*, 3598. [CrossRef] [PubMed]
46. Altaf, M.; Akram, T.; Khan, M.A.; Iqbal, M.; Ch, M.M.I.; Hsu, C.H. A new statistical features based approach for bearing fault diagnosis using vibration signals. *Sensors* **2022**, *22*, 2012. [CrossRef]
47. Desai, M.; Shah, M. An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN). *Clin. EHealth* **2021**, *4*, 1–11. [CrossRef]
48. Heidari, A.A.; Faris, H.; Mirjalili, S.; Aljarah, I.; Mafarja, M. Ant lion optimizer: Theory, literature review, and application in multi-layer perceptron neural networks. In *Nature-Inspired Optimizers: Theories, Literature Reviews and Applications*; Springer: Cham, Switzerland, 2020; pp. 23–46.
49. Car, Z.; Baressi Šegota, S.; Anđelić, N.; Lorencin, I.; Mrzljak, V. Modeling the Spread of COVID-19 Infection Using a Multilayer Perceptron. *Comput. Math. Methods Med.* **2020**, *2020*, 5714714. [CrossRef]
50. Qiao, H. Degradation Measurement of Robot Arm Position Accuracy. *Dataset* **2018**, *10*, M31962. [CrossRef]
51. Clemente-Lopez, D.; Rangel-Magdaleno, J.; Munoz-Pacheco, J.; Morales-Velazquez, L. A comparison of embedded and non-embedded FPGA implementations for fractional chaos-based random number generators. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *14*, 11023–11037. [CrossRef]
52. Cureño-Osornio, J.; Zamudio-Ramirez, I.; Morales-Velazquez, L.; Jaen-Cuellar, A.Y.; Osornio-Rios, R.A.; Antonino-Daviu, J.A. FPGA-Flux Proprietary System for Online Detection of Outer Race Faults in Bearings. *Electronics* **2023**, *12*, 1924. [CrossRef]
53. Qiao, G.; Weiss, B.A. Monitoring, Diagnostics, and Prognostics for Robot Tool Center Accuracy Degradation. In Proceedings of the ASME 2018 13th International Manufacturing Science and Engineering Conference, College Station, TX, USA, 18–22 June 2018; p. V003T02A029. [CrossRef]

applied sciences

Review

Generative AI in AI-Based Digital Twins for Fault Diagnosis for Predictive Maintenance in Industry 4.0/5.0

Emilia Mikołajewska [1], Dariusz Mikołajewski [2], Tadeusz Mikołajczyk [3,*] and Tomasz Paczkowski [3]

[1] Faculty of Health Sciences, Nicolaus Copernicus University in Toruń, 85-067 Bydgoszcz, Poland; emiliam@cm.umk.pl
[2] Faculty of Computer Science, Kazimierz Wielki University in Bydgoszcz, 85-067 Bydgoszcz, Poland; dariusz.mikolajewski@ukw.edu.pl
[3] Department of Production Engineering, Bydgoszcz University of Science and Technology, 85-796 Bydgoszcz, Poland; tomasz.paczkowski@pbs.edu.pl
* Correspondence: tadeusz.mikolajczyk@pbs.edu.pl

Featured Application: Potential applications of the work include more reliable dedicated AI-based predictive maintenance systems based on digital twins and generative AI.

Abstract: Generative AI (GenAI) is revolutionizing digital twins (DTs) for fault diagnosis and predictive maintenance in Industry 4.0 and 5.0 by enabling real-time simulation, data augmentation, and improved anomaly detection. DTs, virtual replicas of physical systems, already use generative models to simulate various failure scenarios and rare events, improving system resilience and failure prediction accuracy. They create synthetic datasets that improve training quality while addressing data scarcity and data imbalance. The aim of this paper was to present the current state of the art and perspectives for using AI-based generative DTs for fault diagnosis for predictive maintenance in Industry 4.0/5.0. With GenAI, DTs enable proactive maintenance and minimize downtime, and their latest implementations combine multimodal sensor data to generate more realistic and actionable insights into system performance. This provides realistic operational profiles, identifying potential failure scenarios that traditional methods may miss. New perspectives in this area include the incorporation of Explainable AI (XAI) to increase transparency in decision-making and improve reliability in key industries such as manufacturing, energy, and healthcare. As Industry 5.0 emphasizes a human-centric approach, AI-based generative DT can seamlessly integrate with human operators to support collaboration and decision-making. The implementation of edge computing increases the scalability and real-time capabilities of DTs in smart factories and industrial Internet of Things (IoT) systems. Future advances may include federated learning to ensure data privacy while enabling data exchange between enterprises for fault diagnostics, and the evolution of GenAI alongside industrial systems, ensuring their long-term validity. However, challenges remain in managing computational complexity, ensuring data security, and addressing ethical issues during implementation.

Keywords: artificial intelligence; machine learning; industrial applications; Industry 4.0; Industry 5.0; predictive maintenance; digital twin; generative AI; GenAI

1. Introduction

The Industry 3.0 paradigm introduced automation and early diagnostics. Predictive maintenance is a method of forecasting and managing the condition of machines, devices, production lines based on historical data, mechanism models and domain knowledge that predicts equipment trends, and behavioral patterns and correlations using statistics or artificial intelligence (AI) models. This allows us to predict remaining useful life, upcoming failures, and other key indicators in advance, improving the decision-making process for maintenance activities, reducing the risks associated with failures and avoiding unnecessary equipment downtime [1,2]. Predictive maintenance began with the advent of automation in Industry 3.0, using sensors and condition monitoring systems to track equipment performance. In this phase, techniques such as vibration analysis and thermal imaging were used along with manual data logging for trend analysis, enabling basic failure prediction. Scientists and engineers developed statistical models to predict failures based on historical data, introducing an early form of predictive maintenance. The Industry 4.0 paradigm saw the transition to continuous, real-time data collection from interconnected devices and systems, powered by the Internet of Things (IoT) and Big Data. Machine learning (ML) algorithms began to analyze massive data sets collected from IoT devices to detect patterns, predict failures, and recommend maintenance actions. Digital twins (DTs) were introduced as virtual replicas of physical assets, integrating sensor data and simulations for real-time monitoring and fault diagnosis. Predictive maintenance is related to various areas of research and economic practice [3,4].

ML plays a crucial role in enhancing DT technology by enabling real-time data analysis, predictive modeling, and automation. DTs, which are virtual replicas of physical assets, leverage ML to process vast amounts of sensor data and detect anomalies [5]. By using predictive analytics, ML helps forecast potential failures and optimize maintenance schedules, reducing downtime and costs [6]. In manufacturing, DTs powered by ML enhance production efficiency by identifying inefficiencies and improving quality control [7]. Healthcare applications benefit from AI-driven DTs that simulate patient conditions, allowing personalized treatment planning [8]. In smart cities, ML-based DTs optimize traffic flow, energy usage, and infrastructure management [9]. The aerospace industry uses DTs to simulate aircraft performance under various conditions, ensuring better safety and efficiency [10]. ML also improves DTs in supply chain management by predicting demand fluctuations and optimizing logistics [11]. These AI-enhanced virtual models support sustainability by analyzing environmental impacts and optimizing resource usage. ML significantly enhances the capabilities of DTs, making them more accurate, intelligent, and beneficial across industries.

Industry 4.0 also saw the rise in edge computing, which enabled predictive models to process data locally, reducing latency and increasing responsiveness. In the Industry 5.0 paradigm, predictive maintenance has moved towards a collaborative human-AI approach, emphasizing user-friendly tools and sustainable maintenance practices [12]. GenAI has begun to enhance DTs by simulating complex scenarios, generating synthetic data, and improving fault detection and predictive models [13]. DTs based on GenAI in Industry 5.0 focus on optimizing asset performance while adapting to environmental goals, which is the latest advancement in predictive maintenance [14]. The future implications of GenAI, large language models (LLM) and search-augmented generation (RAG) will influence practices in such distant industrial sectors as construction, but also specialist education [15].

GenAI plays a significant transformational role in AI-based DTs for fault diagnostics, driving the advancement of predictive maintenance for Industry 5.0. It enables realistic, yet human-operator-level perceptual simulations of complex industrial systems by generating synthetic data and scenarios, helping to predict and diagnose faults with greater accuracy. This increases the ability of the operator—cyber-physical system to detect trends, upcoming anomalies, identify their root causes and predict failures before they occur, compared to solutions without GenAI [16]. GenAI also helps improve the quality and quantity of training data by addressing limitations in sensor data availability and improving the robustness of the ML model [17]. Additionally, GenAI supports real-time updates to DTs by synthesizing data from different sources, ensuring that the twin reflects the current state of the asset. It facilitates the analysis of alternative "what if" scenarios by generating multiple potential failure scenarios (instead of the most likely ones, as before), enabling proactive maintenance strategies [18]. The technology improves decision-making by providing actionable insights derived from patterns and correlations identified in the generated data. GenAI also helps optimize system performance by simulating the impact of different maintenance actions, allowing operators to select the most effective intervention [19]. As Industry 5.0 emphasizes human-centric and sustainable solutions, GenAI in DTs aligns with these goals, enabling smarter, more efficient, and less resource-intensive maintenance practices [20].

GenAI is a rapidly growing area of research in deep learning. GenAI algorithms generate new realistic data in various modalities (text, images, music, three-dimensional (3D) models) and multimodalities [21]. Key features of GenAI include the ability to synthesize data and learning distributions. GenAI systems can synthesize new data that has features from other elements of the data set [22]. Thus, a GenAI model trained on images of dogs can create new, realistic images of dogs that do not exist in the training set [23]. GenAI models probability distributions of data, from which it can sample new instances that follow learned patterns [24]. GenAI uses state-of-the-art deep learning architectures that are rapidly evolving. The most important of them are the following:

- Generative Adversarial Networks (GANs)—involve the interaction between two neural networks: a generator and a discriminator. The generator creates synthetic data, while the discriminator distinguishes real from fake data. Through iterative competition, GANs generate highly realistic results and are used for image synthesis (DeepArt, DeepFake), video content generation, and style transfer (painting, photography);
- Variational Autoencoders (VAE)—encode input data into a latent representation and decode it back, following a predefined distribution. This enables smooth interpolation between data points for image reconstruction, anomaly detection, or drug discovery;
- Transformer models—use self-attention mechanisms to generate coherent and contextually relevant sequences within language models (ChatGPT, BERT) or description-based image synthesis (DALL-E);
- Diffusion models—an alternative to GANs, learn to reverse the noise process applied to data, gradually producing realistic results in high-quality image synthesis or molecular structure generation [25–27].

GenAI allows us to create content that integrates multiple modalities, e.g., generating an image from a text description. It offers computationally efficient models that scale to large data sets. For the above reasons, GenAI in a sense expands creativity, providing a selection from a larger set of quickly generated data with similar characteristics [28]. This allows us to generate synthetic data (industrial, medical, etc.) for training AI models or DTs, including without concerns about privacy or security, creating unique narratives and visualizations, accelerating discoveries by generating molecular structures, optimizing engineering designs and simulating physical phenomena [29]. The biggest challenge is still to provide users with greater control over the generated content and make the data generation process more transparent (Table 1). GenAI expands the boundaries of what machines can create, including in cooperation with humans, both as a tool for analysis and as a partner in creation [30].

Table 1. Research gaps observed in the state of the art in GenAI-based DTs for fault diagnosis for predictive maintenance in Industry 4.0/5.0 (own version).

Area	Identified	
	Gap(s)	**Possibilities of Closing Gap(s)**
Real-time data fusion and processing	Current methods struggle to effectively integrate multimodal data (e.g., IoT sensor readings, historical records, and environmental factors) in real time to achieve predictive accuracy.	Explore scalable architectures for real-time data fusion using GenAI capabilities.
Explainability and interpretability	The "black box" nature of many GenAI models limits trust and adoption in industrial environments.	Develop interpretable GenAI models that can explain failure prediction decisions in a way that is understandable to human operators.
Generating synthetic data for rare faults	Industrial systems often lack sufficient labeled data for rare but critical fault types.	Explore GenAI techniques to generate high-quality synthetic datasets that mimic rare fault conditions for robust training.
Adaptive learning in changing environments	Current models are not suited for dynamic industrial environments where machine configurations and operating conditions change frequently.	Develop an adaptive GenAI framework that can learn continuously and update DTs without complete retraining.
Integration of domain knowledge	Many GenAI approaches neglect the integration of domain expert knowledge, leading to less reliable fault diagnosis.	Combine domain knowledge with generative models to improve fault diagnosis reliability and contextual validity.
Generalization across device(s) types	Existing GenAI models are often tailored to specific machines and lack cross-device generalization.	Design generalized GenAI-based DTs that can transfer knowledge across different device types and configurations.
Cybersecurity in GenAI-based DTs	Increased connectivity and dependency on GenAI increase cybersecurity vulnerability in DTs.	Develop secure GenAI frameworks that protect sensitive industrial data while maintaining predictive accuracy.
Low-power, edge-compatible solutions	Many GenAI models are computationally intensive, making them unsuitable for edge deployments in smart factories.	Optimize GenAI algorithms for resource-constrained environments, enabling deployment on edge devices.
Multi-twin collaboration	Collaboration between multiple DTs for complex systems or connected machines is underexplored.	Explore a framework for GenAI-enabled multi-twin ecosystems to improve fault diagnostics in connected environments.
DTs lifecycle management	Limited research addresses long-term DT lifecycle management, such as model updates or retirement of outdated twins.	Develop methods for continuous evolution and maintenance of GenAI-based DTs to ensure continued accuracy and relevance.

To fill these gaps, interdisciplinary approaches combining advances in AI, edge computing, industrial engineering, and cybersecurity are needed to fully leverage the potential of GenAI-based DT for predictive maintenance in Industry 4.0/5.0 [31].

The motivation behind integrating GenAI with AI-based DTs in fault diagnosis for predictive maintenance within Industry 4.0/5.0 stems from the need to enhance real-time decision-making and predictive capabilities. With industrial systems becoming increasingly complex and interconnected, traditional diagnostic methods struggle to cope with the sheer volume and variety of data generated. The primary challenges include accurately modeling dynamic systems, dealing with incomplete or noisy data, and providing timely predictions to avoid costly downtime. GenAI offers a way to simulate various fault scenarios and predict potential failures by creating realistic synthetic data that complement real-world data. This approach can improve the robustness of DTs by enabling them to adapt and learn from evolving system behaviors. A novel contribution of this work is leveraging GenAI to continuously update and refine digital twin models, ensuring they remain accurate and relevant. Additionally, integrating AI-driven anomaly detection with generative models helps in identifying previously unseen faults. This synergy fosters a proactive maintenance strategy, minimizing unexpected failures and optimizing maintenance schedules. By embedding generative capabilities, the DTs evolve beyond static representations, becoming adaptive tools capable of scenario analysis and prescriptive insights. Consequently, this advancement pushes the boundaries of what's possible in predictive maintenance, aligning closely with the smart, automated vision of Industry 5.0 [32].

GenAI is being integrated with AI-based DTs by enhancing their ability to simulate, predict, and adapt to real-world conditions in industrial environments. This integration involves using generative models such as VAEs and GANs to create synthetic datasets that complement real-world sensor data, improving fault detection even when historical failure data are sparse. Device states are described by feature vectors (at points in time) or feature matrices (time-varying feature vectors) in edge computing and include all relevant features (states, parameters) of both normal operation and impending wear and tear, failure, and attack, for example. GenAI also enables anomaly detection by learning the normal behavior of the system and generating deviations that signal potential failures before they become critical failures. GenAI-powered DTs can run multiple failure simulations to predict the impact of different faults, allowing industries to proactively optimize maintenance strategies. One of the major issues they address is the challenge of incomplete or noisy data, as GenAI can fill in missing information and remove signal noise to increase diagnostic accuracy. The integration also mitigates the high costs and risks associated with physical fault testing by creating realistic virtual failure scenarios for analysis. Another problem it addresses is the inability of traditional AI models to generalize across machines and environments, as generative models can adapt to changes in operating conditions. By continuously updating DTs with new synthetic and live data, the system remains dynamic and responsive to evolving industrial processes. This approach enhances predictive maintenance by reducing false positives and false negatives, ensuring that maintenance actions are taken only when necessary. In this way, generative AI empowers DTs, transforming them from static models into self-learning, adaptive systems that provide more accurate and timely fault diagnosis in Industry 4.0/5.0 [33].

This paper presents the current state of the art and prospects for using generative AI-based DTs to diagnose faults for predictive maintenance in Industry 4.0/5.0.

2. Materials and Methods

2.1. Data Set

Our bibliometric analysis aimed to investigate the research landscape and current knowledge and practices related to planning and implementing GenAI-based DTs for fault diagnosis in predictive maintenance within the framework of Industry 4.0/5.0 paradigms. To achieve this, we used bibliometric methods to examine scientific publications by defining research questions to identify key aspects, including the current state of the field, the origin and evolution of research topics, sources of publications (institutions, countries, and funding mechanisms), and the most influential authors and research teams. This methodology provides a comprehensive overview of current research and industry trends in GenAI-based DTs for fault diagnosis in predictive maintenance. By analyzing bibliometric data, this study contributes to the ongoing discussions and helps to establish a solid foundation for future research, identifying priority directions and research teams to follow in the coming years.

2.2. Methods

The study used the bibliographic databases Web of Science (WoS), Scopus, and dblp, selected for their extensive research collection and rich citation data, which facilitated a comprehensive bibliometric analysis of GenAI DT for failure diagnosis in predictive maintenance under Industry 4.0/5.0 (Table 2). To ensure greater relevance of the results, filters were applied to focus on original articles in English. Each of the selected articles was then manually reviewed to confirm its compliance with the inclusion criteria, refining the final set of analyzed articles. Key features of the data set were then examined, including prominent authors, research groups, institutions, countries, topic areas and emerging trends. This analysis helped to trace the evolution of key terminology and major research advances in the field. Furthermore, where possible, temporal trends were analyzed to track changes in research coverage over time, and publications were categorized into topic areas to discover relationships among different clusters of them. This process ultimately highlighted significant themes and subfields within the overall domain of study.

Table 2. Bibliometric analysis procedure (own approach).

Stage	Name	Tasks
1	Defining research objectives	Defining goals of the bibliometric analysis
2	Selecting databases and data collections	Choosing appropriate data set(s) and developing research queries according to the study goals
3	Data preprocessing	Cleaning the collected date to remove duplicates and irrelevant records
4	Bibliometric software selection	Choosing suitable bibliometric software tools for analysis
5	Data analysis	Description, author, journal, area/topics, institution/country, etc.
6	Visualization (where possible)	Visualizing the analysis results to present insights
7	Interpretation and discussion	Interpreting findings in the context of the research goals

This study followed specific elements of the PRISMA 2020 guidelines [34] for bibliographic reviews (Supplementary Materials), focusing on key aspects such as rationale (item 3), objectives (item 4), eligibility criteria (item 5), information sources (item 6), search strategy (item 7), selection process (item 8), data collection (item 9), synthesis methods (item 13a), synthesis results (item 20b), and discussion (item 23a). Bibliometric analysis was performed using tools available in the Web of Science (WoS), Scopus, and dblp databases, as well as the Biblioshiny tool from the Bibliometrix v.4.1.3 package. The results are presented in a table, which allows for flexible analysis and visualization. Considering the interdisciplinary nature and complexity of the topic, the most important results of the review are summarized in a concise table for clarity.

3. Results

3.1. Data Sources

To refine the search, advanced filtering techniques were used, limiting the results to articles in English. In WoS, searches were performed using the "Subject" field, which includes title, abstract, keyword plus, and additional keywords. In Scopus, searches were performed using article title, abstract, and keywords, while in dblp, manual keyword selection was used. Databases were searched using keywords such as "generative artificial intelligence", "digital twin", and "Industry 4.0" or "Industry 5.0" (Table 3).

The selected set of publications was then further refined (see Figure 1) by manually re-reviewing the article and removing irrelevant items and duplicates, which allowed us to determine the final sample size.

The summary of the bibliographic analysis results is presented in Table 4. The review included 21 articles (2023–2024) published in the last two years (no older ones were included).

Table 3. Detail search query over databases.

Parameter	Description
Inclusion criteria	Articles (original, reviews, communication, editorials) and chapters, including conference proceedings, in English
Exclusion criteria	Books older than 10 years, letters, conference abstracts without full text, other languages than English
Keywords used	Artificial intelligence, generative AI, digital twin, predictive maintenance, Industry 4.0, Industry 5.0
Used field codes (WoS)	"Subject" field (consisting of title, abstract, keyword plus, and other keywords)
Used field codes (Sopus)	Article title, abstract, and keywords
Used field codes (dblp)	Manually
Boolean operators used	Yes, e.g., "digital twin" AND ("Industry 4.0" OR "Industry 5.0") AND rehabilitation
Applied filters	Results refined by publication year, document type (e.g., articles, reviews), and subject area (e.g., industry, engineering)
Iteration and validation options	Query run iteratively, refinement based on the results, and validation by ensuring relevant articles appear among the top hits
Leverage truncation and wildcards used	Used symbols like * for word variations (e.g., "digital twin *") and ? for alternative spellings (e.g., "Industry ?.0")

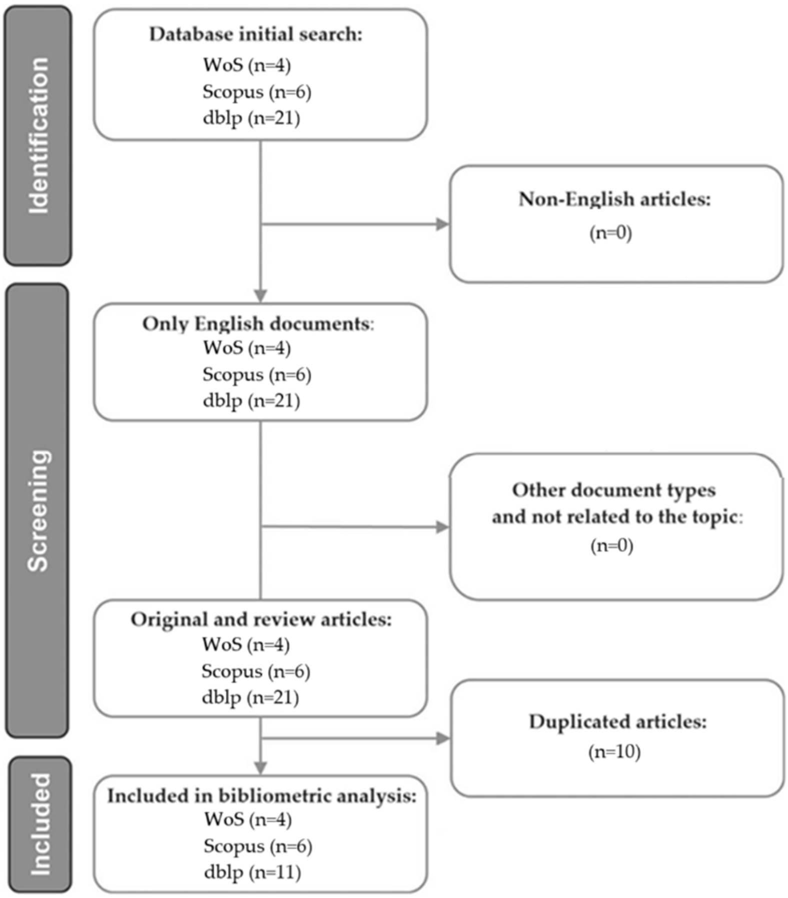


Figure 1. PRISMA flow diagram of the review process using selected PRISMA 2020 guidelines.

Table 4. Summary of results of bibliographic analysis (WoS, Scopus, dblp).

Parameter/Feature	Value
Leading types of publication	Conference review (50.0%), article (16.7%), conference paper (33.3%)
Leading areas of science	Computer science (50.0%), Engineering (20.0%), Mathematics (20.0%), Materials Science (10.0%)
Leading topics	Industrial: Design and Manufacturing
Leading countries	Bulgaria, Germany
Leading scientists	Mateev, M., Jazdi, N., Weyrich, M., Xia, Y., Xiao, Z.
Leading affiliations	University of Architecture, Civil Engineering and Geodesy, Sofia, Bulgaria, Universitat Stuttgart, Germany
Leading funders (where information available)	None
Sustainable development goals	Industry Innovation and Infrastructure, Responsible Consumption and Production

Successful establishment of DT requires high fidelity virtual modeling and strong information interactions. GenAI can use advanced AI algorithms to automatically create, manipulate, and modify the desired sparse, correct, and diverse data. However, the implementation of this technology faces numerous challenges and perhaps should be implemented more quickly in specific areas of Industry 4.0/5.0 [35].

3.2. IIoT Background and the Potential of GenAI-Driven DT

IIoT is revolutionizing industrial sectors by connecting machines, sensors, and devices through networks, enabling real-time data acquisition and intelligent decision-making. Industrial IoT can be used for process monitoring [36,37]. It emphasizes predictive maintenance, process optimization, and asset management, leveraging advanced analytics on massive data sets collected from industrial operations. DTs, acting as virtual replicas of physical systems, leverage IIoT data to model, simulate, and monitor performance [38–41]. By integrating IIoT and GenAI, enterprises can unlock new, higher levels of operational intelligence and decision-making agility. GenAI introduces a new dimension to DTs by enabling them to learn, predict, and generate new data scenarios, significantly increasing their accuracy and usability. Unlike traditional DTs that rely on fixed data patterns, generative DTs powered by AI can model complex, nonlinear systems and propose innovative solutions. This capability is especially transformative in industries such as manufacturing, energy, and transportation, where high variability and system complexity require dynamic adaptation. GenAI can simulate how machines will operate under unprecedented conditions, reducing downtime and mitigating risk. It also enables real-time scenario analysis, helping industries to adapt to disruptions very quickly or optimize resource allocation. Additionally, these intelligent DTs facilitate sustainable practices by modeling energy efficiency and predicting the carbon footprint of industrial operations. This convergence may redefine industry standards, supporting innovation and resilience in an era of rapid technological advancement, not only within the Industry 4.0 and Industry 5.0 paradigms but also their next generations [42].

Recent publications on GenAI in AI-based DTs for fault diagnosis and predictive maintenance in Industry 4.0/5.0 show the growing focus on increasing predictive accuracy and real-time decision-making. Researchers are increasingly exploring the integration of GenAI with IoT, edge computing, and cloud-based architectures to improve system responsiveness. Many studies emphasize the use of synthetic data generation to address the problem of limited failure data in industrial environments. There is a noticeable shift towards explainable AI (XAI) to improve transparency and trust in AI-based maintenance decisions. The most progressive topics include deep learning-based anomaly detection, reinforcement learning for adaptive maintenance strategies, and hybrid AI models combining physics-based and data-based approaches. Publications also emphasize the role of GenAI in enabling self-learning DTs that evolve with operational data. This trend indicates a shift towards more autonomous, scalable, and human-centric AI-based maintenance solutions in smart industries [43].

3.3. Basic Methods of Generative AI-Driven DTs

GenAI-driven DTs leverage advanced ML techniques (such as GANs and VAEs) to simulate and generate realistic data scenarios. These models learn patterns from IIoT data, allowing them to predict, optimize, and replicate complex behaviors of real-world systems. Reinforcement learning (RL) further enables DTs to adapt to dynamic environments by testing and refining decision-making strategies. Natural language processing (NLP) techniques allow these DTs to interpret and integrate textual data such as maintenance logs, operator reports, and technical documentation into their analyses. Additionally, transfer learning helps leverage pre-trained models to accelerate the development and deployment of AI-driven generative DTs in various industrial contexts [44].

GANs enhance AI-based DTs by generating synthetic sensor data that replicates real-world failure conditions, helping predictive maintenance models detect rare and complex failure modes in Industry 4.0/5.0 environments. GANs create different failure scenarios by learning statistical patterns of normal and faulty operating states, enabling digital twins to improve fault diagnostic accuracy even when historical failure data are limited. The generator synthesizes realistic fault signals, while the discriminator improves its ability to distinguish between real and artificial faults, strengthening the fault detection and anomaly recognition capabilities of the digital twin. By continuously learning from live sensor data, GAN-based digital twins detect deviations from normal operation in real time, enabling early failure prediction and reducing unplanned downtime. When combined with reinforcement learning, GANs optimize maintenance decisions based on synthetic fault simulations, while federated learning ensures secure model training across multiple industrial sites without sharing sensitive operational data [45].

VAEs enhance AI-based DTs by learning the underlying distribution of normal operating data, enabling the detection of deviations that signal potential failures in predictive maintenance for Industry 4.0/5.0. VAEs encode high-dimensional sensor data into a lower-dimensional latent space, capturing salient features of machine behavior and facilitating the identification of anomalies that indicate emerging failures. VAEs reconstruct sensor signals from compressed latent representations, and when the reconstruction error exceeds a threshold, it suggests the presence of an unknown or abnormal fault condition. Unlike traditional deterministic models, VAEs generate probabilistic results, enabling digital twins to assess the uncertainty of failure predictions and reduce false positives in predictive maintenance. VAE can be combined with RL to optimize maintenance decisions based on detected anomalies, while federated learning enables secure, decentralized model training across multiple industrial facilities without the need to share raw sensor data [46].

Transformers, such as Bidirectional Encoder Representations from Transformers (BERT), improve predictive maintenance by processing sensor data in large time series, capturing complex relationships in industrial systems, and improving fault diagnostics in Industry 4.0/5.0. Transformers' self-driving mechanism enables AI-based digital twins to analyze long-term relationships in sensor data, identifying subtle fault patterns that traditional machine learning models may miss. Transformer models pre-trained on extensive industrial data sets can be tuned to specific device types, enabling more accurate fault detection and predictive maintenance tailored to unique operating conditions. Transformers analyze streaming data in real time, learning about normal machine behavior and flagging anomalies that indicate early signs of failure, reducing unplanned downtime in manufacturing and industrial processes. Combined with generative AI, transformers improve fault simulation and synthetic data generation, while RL optimizes maintenance strategies by continuously adapting fault diagnosis models based on real-time feedback from DTs [47].

RL enables AI-based DTs to optimize predictive maintenance by continuously learning from real-time equipment data and adapting maintenance strategies based on Industry 4.0/5.0 fault diagnostics results. RL agents interact with digital twins to simulate different maintenance actions, receiving rewards based on reduced downtime, increased fault detection accuracy, and minimized operating costs. Unlike traditional rule-based approaches, RL-based digital twins dynamically refine fault diagnostics and maintenance strategies, improving decision-making as more operational data are processed over time. By combining RL with generative AI, digital twins can simulate rare fault conditions, enabling the RL model to learn from different failure scenarios and make more accurate predictive maintenance recommendations. In complex manufacturing environments, multi-agent RL enables multiple AI-powered digital twins to collaborate, optimizing fault diagnosis and predictive maintenance strategies for interconnected industrial assets [48].

Federated learning enables multiple industrial plants to jointly train predictive maintenance models while maintaining data privacy, and generative AI enhances this process by generating synthetic fault data to increase model robustness in Industry 4.0/5.0. Instead of sharing raw sensor data, federated learning enables AI-based digital twins to exchange model updates, keeping sensitive operational data secure while leveraging collective intelligence across manufacturing plants. Generative models such as GANs and VAEs generate realistic fault scenarios to enrich federated learning models, compensating for imbalanced data sets where certain failure conditions may be underrepresented. AI-based digital twins within a federated learning network continuously update fault diagnostic models using locally generated synthetic data, enabling real-time adaptation to unique operating conditions in different industrial environments. By integrating federated learning with generative AI, industrial systems develop more accurate, context-aware predictive maintenance strategies that reduce downtime and improve fault detection without compromising data security or requiring centralized data storage [49]. The generation of extended data to improve error detection in images or time series signals has been proposed in [50,51].

Numerical comparisons of different generative AI methods in AI-based DTs for fault diagnosis and predictive maintenance reveal distinct advantages based on accuracy, efficiency, and computational cost. GANs typically improve fault classification accuracy by 5–15% by generating realistic failure data for training models. VAEs achieve anomaly detection precision of 85–95% on average, depending on system complexity and available data. Transformer-based models such as BERT time series variations outperform traditional recursive models by 10–20% in predictive accuracy due to their ability to capture long-range dependencies. Reinforcement learning (RL) approaches reduce maintenance costs by 20–40% by optimizing predictive scheduling strategies compared to rule-based systems. Physics-based generative models reduce false positive rates by 30–50% because they combine physical simulations with AI-based analyses to provide more reliable fault diagnosis. Federated learning with Generative AI improves model generalization across multiple industrial sites while maintaining accuracy at 95%+, although it may require 20–30% more computational resources due to decentralized processing. These comparisons highlight the trade-offs between accuracy, computational cost, and scalability, which is the basis for selecting the right Generative AI techniques for different industrial applications [52,53].

The increasing understanding of AI algorithms brings about a deeper understanding and classification of them by researchers and practitioners, who can apply the appropriate ones to obtain optimal results in the shortest possible time with less effort for their specific application area problems in a novel and significant way [54].

Here are some important frameworks that can help understand generative AI for fault diagnosis in AI-based digital twins for predictive maintenance in Industry 4.0/5.0:

1. Conceptual diagram of GenAI in DTs:
    - Shows interaction between real-world industrial systems, IoT sensors, AI-based digital twins, and predictive maintenance systems;
    - Highlights how real-time data are collected, processed by GenAI models, and used to predict faults.
2. Comparison chart of GenAI methods comparing GANs, VAEs, transformer-based models, reinforcement learning, and physics-informed models based on accuracy, computational cost, scalability, and fault detection capabilities.
3. Workflow of GenAI-Based fault diagnosis, i.e., a step-by-step process illustrating the following:
    - Data collection from IoT sensors;
    - Preprocessing and feature extraction;
    - Model training using GenAI;
    - Fault detection and predictive maintenance actions.
4. Illustration of GANs vs. VAEs for anomaly detection showing how GANs generate synthetic failure data while VAEs reconstruct normal operation data to detect anomalies.
5. Transformer-based time-series prediction showing how transformer models analyze historical machine data and predict failures with multi-step forecasting.
6. Impact of GenAI on maintenance costs and downtime comparing reactive, preventive, and predictive maintenance strategies, highlighting the efficiency gains achieved by Generative AI.
7. Federated learning for cross-site fault diagnosis shows how decentralized AI models share insights across multiple industrial sites while preserving data privacy [55,56].

The digitization of data throughout the product life cycle provided by DTs in cyber physical systems enables a rapid transition from current industrial solutions to intelligent and adaptive solutions. GenAI promotes the construction, modernization, and updating of data in DTs to increase the predictive accuracy and ensure differentiated smart manufacturing by detecting IIoT devices and sharing data. The problem here is the adverse selection caused by information asymmetry. Here, a contract theory model based on a balanced soft actor-critic algorithm based on diffusion is proposed. It provides the identification of the optimal feasible contract and also reduces the number of actor network parameters through the dynamic structural pruning technique [57].

It does this by analyzing huge amounts of data from various sources, such as sensors, smart meters, and historical production and energy consumption patterns, and AI algorithms can identify patterns and anomalies that are difficult for humans to detect. This enables the development of predictive models that optimize production and consumption.

The reliability of data generated by Generative AI in AI-based DTs for fault diagnosis and predictive maintenance is critical to ensuring accurate decision-making in Industry 4.0/5.0. One way to establish reliability is through rigorous validation techniques, such as comparing generated data with real-world sensor readings to assess accuracy. Generative models can be trained using high-quality, diverse data sets to reduce bias and improve their ability to generalize across industrial environments. Another approach is uncertainty quantification, where confidence levels are assigned to generated results, helping maintenance teams assess the reliability of AI-generated predictions. Hybrid AI models that combine generative approaches with physics-based simulations increase data reliability by ensuring that the patterns generated are consistent with known system behaviors. Domain adaptation techniques can further refine generative models to match specific machine characteristics, increasing the relevance of synthetic data for fault diagnosis. Continuous learning mechanisms enable generative models to be updated based on real-time feedback, ensuring they evolve with changing system conditions and avoiding outdated or misleading predictions. Implementing explainability techniques such as attention mechanisms or feature attribution can increase transparency and help engineers understand how the model generates and uses synthetic data. Cross-validation with expert knowledge ensures that AI-generated insights align with human expertise, reducing the risk of incorrect error predictions. Finally, regulatory compliance and adherence to industry standards help build trust in generative AI-powered DTs, ensuring that the generated data meets the reliability and safety requirements for industrial applications.

A lightweight automatic data augmentation framework (ALADA) is proposed to optimize data augmentation rules and industrial defect detection solutions. It provides more efficient augmentation and generation of augmented images for joint optimization, hyperparameter tuning for retraining with searched rules, and also reduces the risk of defect failure in four situations: textured background, non-uniform brightness, low contrast, and intra-class difference, which is validated on three industrial defect detection datasets, namely Tianchi-TILE, GC10-DET, and NEU-DET [58].

Effective communication is the backbone of GenAI-driven DTs, enabling robust, adaptive, real-time industrial operations. Fast and error-free communication in AI-driven DTs is essential for seamless interaction between physical and virtual systems. Data integration is achieved through IIoT devices, where sensors and edge devices transmit real-time data to DTs for analysis and simulation. Advanced protocols such as MQTT, OPC-UA, and 5G networks facilitate low-latency and secure data transfer, ensuring synchronized operations between physical and digital twins. It is worth nothing that GenAI-driven DTs leverage cloud and edge computing for scalable and efficient communication, enabling rapid processing of complex data streams. Interoperability standards and application programming interfaces (APIs) are key, enabling DTs to interact with diverse systems, applications, and stakeholders in industrial ecosystems. Bidirectional communication ensures that insights or decisions generated by DT can be fed back to physical systems for execution or intervention. GenAI enhances communication by interpreting unstructured data sources (such as natural language maintenance reports, image data) and integrating them into the DT model. User interfaces, including dashboards and voice command systems, enable intuitive communication with human operators, making insights accessible and actionable. Collaboration capabilities allow multiple GenAI-based DTs or systems to share insights, enabling optimized performance at the network or organizational level.

Data management in GenAI-powered DTs is the foundation of their functionality and effectiveness. It starts with robust data acquisition from IIoT devices, collecting real-time information such as sensor readings, operational parameters, and environmental conditions. These data are then preprocessed using techniques such as normalization, filtering, and outlier detection to ensure quality and relevance. Centralized or distributed data architectures, often leveraging cloud and edge computing, are used to store and process massive amounts of structured and unstructured data. Advanced data integration frameworks enable the merging of heterogeneous data sources such as machine logs, video feeds, and maintenance records into a unified platform, and multi-modal data are increasingly being integrated. GenAI algorithms analyze and synthesize these data, generating predictive insights, simulations, or new system optimization scenarios. Metadata management (data about data) is also critical, ensuring that data provenance, context, and relationships are well documented to support the interpretability/explainability of GenAI. Security and compliance protocols that protect sensitive industrial data implement encryption, access controls, and compliance with regulations such as the General Data Protection Regulation (GDPR) or industry standards. Scalable storage solutions and intelligent indexing facilitate efficient retrieval and manipulation of data in real time. Periodic data collection and lifecycle management ensure that DT operates on accurate, timely, and meaningful data sets. By combining advanced analytics with disciplined data practices, AI-powered generative DT can drive transformational improvements in industrial processes.

A Weighted Extreme Learning Machine (WELM) is proposed to provide balanced class distribution and reduce data complexity by generating new samples and removing overlapping noisy samples at class boundaries. The effectiveness of the above solution is demonstrated by allocating the most efficient resources to the most urgent orders to avoid delays in the supply chain [59].

GenAI extends DTs beyond their current capabilities into more dynamic, predictive, and interactive tools that simulate complex scenarios and predict future conditions with remarkable accuracy. Depending on the level of GenAI integration into DTs, DTs can be extended to varying degrees to generate synthetic data sets, simulate events/scenarios that have no previous equivalents (e.g., isolated failures), and provide second opinions for decision-making based on LLM agent networks. This has varying implications for operational efficiency, innovation, and decision-making processes [60]. Different GenAI models allow for DT state emulation, function abstraction, and decision-making based on the interaction between GAI-based and model-driven data processing [61]. Three approaches have been proposed for network management:

- Light model weighting;
- Adaptive model selection;
- Data model-driven management [61].

Modeling in the absence of data, e.g., higher resolution photovoltaic (PV) systems (down to individual households or hourly), is a huge obstacle to making informed and accurate decisions. This requires new methods to generate detailed realistic data sets—such as integrated ML models identifying PV users—and methods to augment data using explainable AI techniques based on key features and their interactions and to generate hourly solar energy production at the household level using an analytical model. The synthetic data sets obtained by the above method are validated against real-world data for DTs for further modeling tasks [62]. Depending on the type and method of providing input data, predictive analysis within GenAI-based DT can be based on different LLM: GPT, DALL-E, DAVINCI, or WHISPER.

3.4. Typical Applications

Typical applications include adaptive monitoring and diagnostics, and adaptive response. Dynamic, evolving features of the physical world require a huge amount of data transmission/exchange to ensure synchronization between the physical world and its virtual image. Such a communication framework can be based on the "look only once" (YOLO) principle. The YOLOv7-X object detector in the case of an apple orchard was used to extract semantic information from captured images of edge devices, reducing the amount of data needed to be transmitted. The meaning of each piece of semantic information is determined based on the trust generated by the object detector. Two resource allocation schemes are proposed:

- Trust-based scheme;

- AI-generated scheme acrlong.

Diffusion models generate an optimal allocation scheme that outperforms the results obtained from the schemes used separately. An additional improvement is provided by the attention modules of the ELAN-H and SimAM layer aggregation network that reduce model parameters and computational complexity when using edge devices with limited performance [63]. The complexity of the supply chain poses a particular challenge due to its lack of transparency, generally accepted standards, and regulations. A blockchain-based process data management solution for recycling and reuse of used electronic devices is proposed, combining DT and GenAI to solve the blockchain performance bottleneck by predicting future data flows. This improves the adaptability and throughput of the system, as well as traceability, prediction accuracy, and efficiency throughout the process [64]. GenAI-supported cellular network DT is also proposed to learn complex network data distribution (environmental, user, and service) from samples from the distribution [65]. GenAI uses these data to generate different scenarios, improving flexibility and practically solving network optimization [66,67]. The integration of GenAI and urban DTs is used to address challenges in the planning and management of built environments, including various urban subsystems (transportation, energy, water, and construction and infrastructure) [68].

Adaptive Response

Enabling sustainable development and efficient use of resources, especially expensive energy, is becoming critical for today's economy. This is due to a number of factors: climate change, resource depletion, and the need for decarbonization and increased innovation in solutions. GenAI, DTs, and big data can help the energy sector achieve greater efficiency, optimize operations, and facilitate decision-making to optimize energy use and reduce waste [69]. The confusion in construction stems from the need to differentiate between two technologies: building information modeling (BIM) and DT, which differ in terms of technologies, maturity levels, data layers, enablers, and functionalities. The research emphasis here is on the convergence of BIM and DT, data integrity, their integration and transmission, bidirectional interoperability, non-technical factors, and data security [12]. Asset Administration Shell (AAS) is a digital twin model in the context of Industry 4.0, assuming that semantic-based communication and meaningful textual data generation are directly related and that these processes are equivalent. An LLM-supported system is implemented to generate standard DT models as instances from raw textual data collected automatically from data sheets describing technical assets. The achievable effective generation rate was 62–79%. The resulting AAS model can be integrated with compliant DT software for data exchange and DT communication and interoperability in industrial applications [70]. OpenAI GPT-4 Turbo with Vision LLM can interpret images and provide textual answers to queries about those images by combining natural language processing and visual understanding. This allows for intelligent extraction of key metadata from images and videos to assess the state of real-world systems and propose sustainability measures. This helps to implement efficient image analysis and prediction models and optimize the cost of the solution using a hybrid approach. GenAI in data analysis increasingly offers efficient and cost-effective solutions for predictive analysis based on vector search and other data analysis methods, including image analysis, case decomposition, hybrid search, and generation of self-adaptive models to find trends and offer preventive actions, even for smaller companies [71].

4. Discussion

AI is based on replicating human intelligence in machine control systems, enabling them to perform tasks that require human cognitive abilities (perception, learning, reasoning, and problem-solving). AI encompasses various methodologies and technologies such as ML, natural language processing, computer vision, and robotics, and GenAI further extends these capabilities with rapid creativity previously unavailable to humans [72].

Generative AI development in AI-based DTs for predictive maintenance fault diagnosis benefits the community by increasing industrial efficiency, reducing operational costs and improving safety. By enabling early fault detection, it minimizes unexpected equipment failures, leading to fewer disruptions in key sectors such as manufacturing, energy, and transportation. The technology supports sustainability efforts by optimizing resource utilization, reducing waste, and extending machine life. Small- and medium-sized enterprises benefit from cost-effective predictive maintenance solutions that were previously available only to large corporations. The workforce also benefits from AI-assisted maintenance by reducing unsafe manual checks and freeing skilled professionals to focus on higher-value tasks. From a managerial perspective, integrating Generative AI with DTs requires a shift to data-driven decision-making, which requires investment in AI knowledge and infrastructure. Managers must ensure ethical AI implementation, balancing automation with human oversight to maintain accountability and transparency. Real-time insights from DTs enable proactive maintenance planning, improved asset utilization, and reduced downtime. Additionally, industries adopting this technology gain competitive advantage by improving service reliability and customer satisfaction. Overall, the advancement of Generative AI in DTs aligns with the Industry 5.0 vision of human-centric, sustainable, and resilient industrial ecosystems [73].

Introducing Generative AI into AI-based DTs for fault diagnosis and predictive maintenance in Industry 4.0/5.0 raises several difficulties and scientific questions beyond the usual technical AI challenges. One major issue is the interpretability of AI-generated insights—how can engineers and decision-makers trust and understand the synthetic fault scenarios generated by AI models? There is also a concern about data authenticity and bias, as synthetic data might reinforce existing biases in training datasets, leading to skewed predictions. A key scientific question is how to balance the trade-off between real-world physics-based models and AI-generated simulations to ensure reliability without excessive computational costs. The integration of Generative AI with human decision-making poses an epistemological challenge: to what extent should AI-driven insights override human expertise in maintenance decisions? Ethical considerations arise when AI is used to automate critical fault diagnosis, particularly regarding accountability in cases of incorrect predictions leading to safety risks or economic losses. Standardization remains an unresolved issue—how can industries establish universal guidelines for using Generative AI in DTs across different sectors? Organizational resistance is another difficulty, as industries must overcome skepticism from stakeholders who may not fully trust AI-generated diagnostics. The scalability of this approach in highly diverse industrial settings raises scientific questions about adaptability—how can generative models generalize across different machine types and operational environments? Additionally, regulatory and compliance concerns present challenges, as industries must ensure AI-driven fault diagnosis meets safety and legal requirements. The economic implications of adopting Generative AI for predictive maintenance need further exploration—how can businesses quantify the long-term cost savings and return on investment from implementing these advanced AI-driven systems [74]?

Improved fault diagnostics and data augmentation for robust models enables faster and more reliable detection of anomalies in real time based on live sensor data, identifying patterns that indicate, for example, early signs of equipment degradation. This allows maintenance schedules to be dynamically adjusted based on real-time information, on the one hand avoiding downtime and on the other reducing unnecessary maintenance.

Clear, explainable GenAI models with easily interpretable results/hints are key to earning operator trust and ensuring that generative AI-based fault predictions are consistent with expert knowledge. Combining AI-generated insights with expert judgment improves decision-making, ensuring that maintenance personnel can verify and refine GenAI-based fault diagnosis recommendations. For these reasons, protecting sensor communications, databases, and GenAI algorithms from attacks and data manipulation is essential to maintaining reliable predictions [75].

4.1. Limitations of Current Solutions and Concepts

The modernity of technologies using GenAI in AI-based DTs for fault diagnostics brings with it a number of limitations that must be taken into account when planning, building, operating, modernizing, and decommissioning/replacing such systems. They are presented in Table 5.

Table 5. Limitations of AI in AI-based DTs for fault diagnostics within Industry 4.0/5.0 paradigm (own version).

Limitation	Description
Dependence on data quality	GenAI models rely heavily on the quality and diversity of training data, so incomplete, uncertain, or biased data can lead to inaccurate simulations and fault diagnoses.
Computational complexity	The computational power required to train and deploy GenAI models can be significantly higher, making it challenging for real-time applications in resource-constrained environments and cost- and energy-intensive. Regular updates and retraining of GenAI models are necessary to keep them current, which increases operational costs.
Scenarios validity	Generated data or scenarios may not always reflect realistic or physically plausible conditions, which can lead to misleading conclusions. This often requires consulting experts.
Model interpretability/explainability	GenAI models, especially those using DL, are often black boxes, making it difficult to understand or undermining the trust in the decisions they generate.
Integration challenges	Integrating GenAI into existing DT frameworks can be complex and require significant AI and domain-specific expertise.
Risk of overfitting	Generative models can overfit to specific patterns in training data, reducing their ability to generalize to unseen error conditions.
Lack of domain-specific context	Without sufficient domain expertise incorporated into the AI model, generative AI may not account for the nuances of operational behavior of specific industrial systems.
Dependence on AI expertise	Successful implementation requires skilled AI practitioners who understand generative models and digital twin technologies, which can be a limiting factor in many industries.
Ethics and security concerns	Data generated by GenAI could potentially raise ethical issues or be used maliciously, such as by creating misleading error scenarios.

Overcoming the above limitations, even partially, will increase the effectiveness of the discussed group of systems and accelerate their full implementation [76,77].

4.2. Directions of Further Research

Complementing the overcoming of the fundamental limitations described above are the most promising directions for further research on GenAI in AI-based DTs for fault diagnosis for predictive maintenance in Industry 4.0/5.0. Advances in generating high-fidelity synthetic data that closely reflect real-world conditions can address data scarcity and improve model performance [78]. Research into GenAI techniques that enable real-time updates and learning can make DTs more dynamic and responsive to changing system conditions. Developing hybrid models that combine GenAI with physics/chemistry/mechanics-based simulations can increase the realism and reliability of DTs for fault diagnosis [79]. Research into interpretable/explainable GenAI (XAI) models can help build trust and make it easier to understand the insights provided by AI-based DTs. Adaptively tailoring GenAI architectures to specific industries or assets can improve their accuracy and relevance in predictive maintenance applications [80]. Furthermore, research into lightweight GenAI models can enable scalability and real-time processing in resource-constrained industrial environments. Exploring how GenAI can seamlessly integrate with IoT sensors and edge computing can enhance data collection and error detection capabilities [81]. Collaborative interdisciplinary research involving AI experts and industry practitioners can ensure that GenAI solutions are practical and tailored to real-world needs [82]. Focusing on the secure implementation of GenAI in DTs can prevent vulnerabilities related to data manipulation and model exploitation [83]. Exploring how GenAI can optimize resource utilization and minimize energy consumption aligns with Industry 5.0's emphasis on sustainability and human-centric approaches [84,85].

Future work on generative AI in AI-based DTs for fault diagnosis and predictive maintenance in Industry 4.0/5.0 will focus on increasing model adaptability, scalability, and real-time decision-making capabilities. One key direction is to integrate multimodal data sources, such as sensor data, historical maintenance logs, and expert knowledge, to increase the accuracy and robustness of fault predictions. Advanced reinforcement learning techniques can be combined with generative AI to enable self-learning DTs that continuously evolve without human intervention. Future developments may also include federated learning to ensure data privacy and enable collaborative intelligence across multiple industrial sites. The use of quantum computing can further accelerate training and inference of generative models, enabling more complex simulations and faster fault diagnosis. Another promising extension is the implementation of AI-based edge computing, where generative models run on localized devices to provide immediate fault predictions without relying on cloud infrastructure. Generative AI can also enable synthetic data augmentation, improving model generalization for rare or invisible fault conditions. As AI-based DTs become more advanced, they can integrate with augmented reality and virtual reality systems to provide immersive diagnostics and training for maintenance personnel as part of Industry 5.0 [86,87]. Additionally, human-in-the-loop AI frameworks will be key to maintaining the interpretability and trust of automated fault diagnostic systems. Future research should also consider regulatory compliance, ethical issues, and standardization of AI-based predictive maintenance technologies to ensure safe and responsible implementation across industries [88,89].

5. Conclusions

DT technologies, including those based on GenAI, enable early detection and correct diagnosis of faults, which will facilitate corrective actions to replace predicted damaged components before failures occur. The number of publications on GenAI-based DTs is not large in relation to the needs, nor does it cover all the observed research gaps, which is why there should be more emphasis on interdisciplinary scientific and economic cooperation in this area. This applies to both collected and generated data, as well as entire environments for their processing. This will not only allow for maintaining control over the development of this group of solutions, but also for their standardization and synchronization of development with the consideration of Explainable AI (XAI).

With respect to the previously observed knowledge and experience gaps, it can be said that Generative AI plays a key role in AI-based DTs for fault diagnosis and predictive maintenance in Industry 4.0 and 5.0. By creating realistic simulations, it enables accurate modeling of machine behavior under different conditions. These AI-driven DTs continuously learn from real-time data, enhancing the ability to detect and predict faults. Generative AI helps in synthesizing missing or sparse failure data, improving diagnostic accuracy. It also enables adaptive maintenance strategies by predicting potential failures before they occur. This reduces downtime, minimizes maintenance costs, and optimizes resource utilization. Moreover, the integration of Generative AI with IoT and edge computing improves real-time monitoring and decision-making. The synergy between AI and digital twins facilitates a proactive, data-driven approach to maintenance in smart industries. As Industry 5.0 emphasizes human-AI collaboration, Generative AI increases interpretability and decision support for human operators. Using Generative AI in DTs transforms predictive maintenance, ensuring reliability, efficiency, and sustainability in industrial operations.

In predictive maintenance, GenAI DTs enable realistic operational profiles, identifying potential failure modes that traditional methods may miss. New opportunities in GenAI-based DTs include:

- Incorporating XAI to enhance decision-making clarity and improve reliability in key industries such as manufacturing, energy, and even employee healthcare as part of preventive medicine;
- Emphasizing a human-centric approach, so GenAI-based DTs can better integrate with human operators to support collaboration and decision-making;
- Implementation of edge AI and distributed computing further increases the scalability and real-time capabilities of DT, and federated learning ensures data privacy.

In this way, increasingly effective DT technologies will increasingly cooperate with operators within the Industry 5.0 paradigm. Challenges remain in managing computational complexity, ensuring data security, and addressing ethical issues during implementation.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app15063166/s1, Partial PRISMA 2020 Checklist.

Author Contributions: Conceptualization, E.M., D.M., T.M. and T.P.; methodology, E.M., D.M., T.M. and T.P.; software, E.M., D.M., T.M. and T.P.; validation, E.M., D.M., T.M. and T.P.; formal analysis, E.M., D.M., T.M. and T.P.; investigation, E.M., D.M., T.M. and T.P.; resources, E.M., D.M., T.M. and T.P.; data curation, E.M., D.M., T.M. and T.P.; writing—original draft preparation, E.M., D.M., T.M. and T.P.; writing—review and editing, E.M., D.M., T.M. and T.P.; visualization, E.M., D.M., T.M. and T.P.; supervision, T.M.; project administration, T.M.; funding acquisition, T.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data set was generated.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhong, D.; Xia, Z.; Zhu, Y.; Duan, J. Overview of predictive maintenance based on digital twin technology. *Heliyon* **2023**, *9*, e14534. [CrossRef] [PubMed]
2. Pech, M.; Vrchota, J.; Bednář, J. Predictive Maintenance and Intelligent Sensors in Smart Factory: Review. *Sensors* **2021**, *21*, 1470. [CrossRef] [PubMed]
3. Mattera, G.; Vozza, M.; Polden, J.; Nele, L.; Pan, Z. Frequency informed convolutional autoencoder for in situ anomaly detection in wire arc additive manufacturing. *J. Intell. Manuf.* **2024**, 1–16. [CrossRef]

4. Dominguez-Monferrer, C.; Guerra-Sancho, A.; Caggiano, A.; Nele, L.; Miguélez, M.H.; Cantero, J.L. Multiresolution analysis for tool failure detection in CFRP/Ti6Al4V hybrid stacks drilling in aircraft assembly lines. *Mech. Syst. Signal Process.* **2024**, *206*, 110925. [CrossRef]
5. Rezazadeh, J.; Ameri Sianaki, O.; Farahbakhsh, R. Machine Learning for IoT Applications and Digital Twins. *Sensors* **2024**, *24*, 5062. [CrossRef]
6. Hamel, C.; Manjurul Ahsan, M.; Raman, S. PMI-DT: Leveraging Digital Twins and Machine Learning for Predictive Modeling and Inspection in Manufacturing. *arXiv* **2024**, arXiv:2411.01299.
7. Colwell, M.; Abolghasemi, M. Digital Twins for forecasting and decision optimisation with machine learning: Applications in wastewater treatment. *arXiv* **2024**, arXiv:2404.14635.
8. Marfoglia, A.; Nardini, F.; Mellone, S.; Carbonaro, A. Representation of Machine Learning Models to Enhance Simulation Capabilities Within Digital Twins in Personalized Healthcare. In Proceedings of the 2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), Biarritz, France, 11–15 March 2024; pp. 100–105.
9. Kumi, S.; Lomotey, R.K.; Deters, R. Integrating Machine Learning and Social Sensing in Smart City Digital Twin for Citizen Feedback. In Proceedings of the 2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Melbourne, Australia, 17–21 December 2023; pp. 980–987.
10. Kabashkin, I. Digital Twin Framework for Aircraft Lifecycle Management Based on Data-Driven Models. *Mathematics* **2024**, *12*, 2979. [CrossRef]
11. Hirata, E.; Watanabe, D.; Chalmoukis, A.; Lambrou, M. A Topic Modeling Approach to Determine Supply Chain Management Priorities Enabled by Digital Twin Technology. *Sustainability* **2024**, *16*, 3552. [CrossRef]
12. Afzal, M.; Li, R.Y.M.; Shoaib, M.; Ayyub, M.F.; Tagliabue, L.C.; Bilal, M.; Ghafoor, H.; Manta, O. Delving into the Digital Twin Developments and Applications in the Construction Industry: A PRISMA Approach. *Sustainability* **2023**, *15*, 16436. [CrossRef]
13. Rojek, I.; Dostatni, E.; Mikołajewski, D.; Pawłowski, L.; Wegrzyn-Wolska, K. Modern approach to sustainable production in the context of Industry 4.0. *Bull. Pol. Acad. Sci. Tech. Sci.* **2022**, *70*, e143828. [CrossRef]
14. Ucar, A.; Karakose, M.; Kırımça, N. Artificial Intelligence for Predictive Maintenance Applications: Key Components, Trustworthiness, and Future Trends. *Appl. Sci.* **2024**, *14*, 898. [CrossRef]
15. Lim, J.-B.; Jeong, J. Factory Simulation of Optimization Techniques Based on Deep Reinforcement Learning for Storage Devices. *Appl. Sci.* **2023**, *13*, 9690. [CrossRef]
16. Wang, Y.; Qi, Y.; Li, J.; Huan, L.; Li, Y.; Xie, B.; Wang, Y. The Wind and Photovoltaic Power Forecasting Method Based on Digital Twins. *Appl. Sci.* **2023**, *13*, 8374. [CrossRef]
17. Rojek, I.; Mikołajewski, D.; Dostatni, E.; Kopowski, J. Specificity of 3D Printing and AI-Based Optimization of Medical Devices Using the Example of a Group of Exoskeletons. *Appl. Sci.* **2023**, *13*, 1060. [CrossRef]
18. Han, X.; Lin, Z.; Clark, C.; Vucetic, B.; Lomax, S. AI Based Digital Twin Model for Cattle Caring. *Sensors* **2022**, *22*, 7118. [CrossRef]
19. Shaposhnyk, O.; Lai, K.; Wolbring, G.; Shmerko, V.; Yanushkevich, S. Next Generation Computing and Communication Hub for First Responders in Smart Cities. *Sensors* **2024**, *24*, 2366. [CrossRef]
20. Akter, N.; Molnar, A.; Georgakopoulos, D. Toward Improving Human Training by Combining Wearable Full-Body IoT Sensors and Machine Learning. *Sensors* **2024**, *24*, 7351. [CrossRef]
21. Aru, J.; Larkum, M.E.; Shine, J.M. The feasibility of artificial consciousness through the lens of neuroscience. *Trends Neurosci.* **2023**, *46*, 1008–1017. [CrossRef]
22. Mogi, K. Artificial intelligence, human cognition, and conscious supremacy. *Front. Psychol.* **2024**, *15*, 1364714. [CrossRef]
23. Zador, A.; Escola, S.; Richards, B.; Ölveczky, B.; Bengio, Y.; Boahen, K.; Botvinick, M.; Chklovskii, D.; Churchland, A.; Clopath, C.; et al. Catalyzing next-generation Artificial Intelligence through NeuroAI. *Nat. Commun.* **2023**, *14*, 1597. [CrossRef] [PubMed]
24. Kanai, R.; Fujisawa, I. Toward a universal theory of consciousness. *Neurosci. Conscious.* **2024**, *2024*, niae022. [CrossRef] [PubMed]
25. Bengesi, S.; El-Sayed, H.; Sarker, M.K.; Houkpati, Y.; Irungu, J.; Oladunni, T. Advancements in Generative AI: A Comprehensive Review of GANs, GPT, Autoencoders, Diffusion Model, and Transformers. *IEEE Access* **2024**, *12*, 69812–69837. [CrossRef]
26. Combs, K.; Bihl, T.J.; Ganapathy, S. Utilization of generative AI for the characterization and identification of visual unknowns. *Nat. Lang. Process. J.* **2024**, *7*, 100064. [CrossRef]
27. Bresson, M.; Xing, Y.; Guo, W. Sim2Real: Generative AI to Enhance Photorealism through Domain Transfer with GAN and Seven-Chanel-360°-Paired-Images Dataset. *Sensors* **2024**, *24*, 94. [CrossRef]
28. Lifelo, Z.; Ding, J.; Ning, H.; Ain, Q.U.; Dhelim, S. Artificial Intelligence-Enabled Metaverse for Sustainable Smart Cities: Technologies, Applications, Challenges, and Future Directions. *Electronics* **2024**, *13*, 4874. [CrossRef]
29. Zhang, L.; Du, Q.; Lu, L.; Zhang, S. Overview of the Integration of Communications, Sensing, Computing, and Storage as Enabling Technologies for the Metaverse over 6G Networks. *Electronics* **2023**, *12*, 3651. [CrossRef]

30. Singh, D.; Akram, S.V.; Singh, R.; Gehlot, A.; Buddhi, D.; Priyadarshi, N.; Sharma, G.; Bokoro, P.N. Building Integrated Photovoltaics 4.0: Digitization of the Photovoltaic Integration in Buildings for a Resilient Infra at Large Scale. *Electronics* **2022**, *11*, 2700. [CrossRef]
31. Takaffoli, M.; Li, S.; Mäkelä, V. Generative AI in User Experience Design and Research: How Do UX Practitioners, Teams, and Companies Use GenAI in Industry? In Proceedings of the 2024 ACM Designing Interactive Systems Conference, Copenhagen, Denmark, 1–5 July 2024.
32. Zhou, J.; Cao, Y.; Lu, Q.; Zhang, W.; Liu, X.; Ni, W. Industrial Large Model: Toward A Generative AI for Industry. In Proceedings of the 2024 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Kingston, ON, Canada, 6–9 August 2024; pp. 80–81.
33. Héjja, F.; Bartók, T.; Dakroub, R.; Kocsis, G. Generative AI for Productivity in Industry and Education. In Proceedings of the 9th International Conference on Complexity, Future Information Systems and Risk, Angers, France, 28–29 April 2024; pp. 128–135.
34. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, n71. [CrossRef]
35. Chen, J.; Shi, Y.; Yi, C.; Du, H.; Kang, J.; Niyato, D. Generative-AI-Driven Human Digital Twin in IoT Healthcare: A Comprehensive Survey. *IEEE Internet Things J.* **2024**, *11*, 34749–34773. [CrossRef]
36. Tran, M.Q.; Elsisi, M.; Mahmoud, K.; Liu, M.K.; Lehtonen, M.; Darwish, M.M. Experimental setup for online fault diagnosis of induction machines via promising IoT and machine learning: Towards industry 4.0 empowerment. *IEEE Access* **2021**, *9*, 115429–115441. [CrossRef]
37. Mattera, G.; Yap, E.W.; Polden, J.; Brown, E.; Nele, L.; Van Duin, S. Utilising unsupervised machine learning and IoT for cost-effective anomaly detection in multi-layer wire arc additive manufacturing. *Int. J. Adv. Manuf. Technol.* **2024**, *135*, 2957–2974. [CrossRef]
38. Chen, J.; Li, S.; Teng, H.; Leng, X.; Li, C.; Kurniawan, R.; Ko, T.J. Digital twin-driven real-time suppression of delamination damage in CFRP drilling. *J. Intell. Manuf.* **2024**, *36*, 1459–1476. [CrossRef]
39. Li, H.; Shi, X.; Wu, B.; Corradi, D.R.; Pan, Z.; Li, H. Wire arc additive manufacturing: A review on digital twinning and visualization process. *J. Manuf. Process.* **2024**, *116*, 293–305. [CrossRef]
40. Mattera, G.; Polden, J.; Norrish, J. Monitoring the gas metal arc additive manufacturing process using unsupervised machine learning. *Weld. World* **2024**, *68*, 2853–2867. [CrossRef]
41. Lopes, T.G.; Aguiar, P.R.; Monson, P.M.D.C.; D'Addona, D.M.; Conceição Júnior, P.D.O.; de Oliveira Junior, R.G. Machine condition monitoring in FDM based on electret microphone, SVM, and neural networks. *Int. J. Adv. Manuf. Technol.* **2023**, *129*, 1769–1786. [CrossRef]
42. Ciolacu, M.I.; Marghescu, C.; Mihailescu, B.; Svasta, P. Does Industry 5.0 Need an Engineering Education 5.0? Exploring Potentials and Challenges in the Age of Generative AI. In Proceedings of the 2024 IEEE Global Engineering Education Conference (EDUCON), Kos Island, Greece, 8–11 May 2024; pp. 1–10.
43. Zacharias, J. Public and Expert Insights into Generative AI: The potential for the Financial Industry. In Proceedings of the INFORMATIK, Wiesbaden, Germany, 24–26 September 2024; pp. 1491–1500.
44. Ali, A.R.; Kumar, K.; Siddiqui, M.A.; Zahid, M. An Open-source Cross-Industry and Cloud-agnostic Generative AI Platform. In Proceedings of the 2024 International Joint Conference on Neural Networks (IJCNN), Yokohama, Japan, 30 June–5 July 2024; pp. 1–10.
45. Available online: https://github.com/he-zh/vibration_gan (accessed on 10 January 2025).
46. Available online: https://github.com/BlingBlingss/VAE-CWGAN-GP (accessed on 10 January 2025).
47. Available online: https://github.com/huggingface/transformers (accessed on 10 January 2025).
48. Available online: https://github.com/opendilab/GenerativeRL (accessed on 10 January 2025).
49. Available online: https://github.com/FederatedAI/research (accessed on 10 January 2025).
50. Mucllari, E.; Cao, Y.; Ye, Q.; Zhang, Y. Modeling imaged welding process dynamic behaviors using Generative Adversarial Network (GAN) for a new foundation to monitor weld penetration using deep learning. *J. Manuf. Process.* **2024**, *124*, 187–195. [CrossRef]
51. Shao, S.; Wang, P.; Yan, R. Generative adversarial networks for data augmentation in machine fault diagnosis. *Comput. Ind.* **2019**, *106*, 85–93. [CrossRef]
52. Kilsby, P.; Ah Kun, L. Enabling Intelligent Robotic Visual Inspection in the Railway Industry with Generative AI. In Proceedings of the 2024 Eighth IEEE International Conference on Robotic Computing (IRC), Tokyo, Japan, 11–13 December 2024; pp. 275–277.
53. Liu, J.; Adsumilli, B.; Yanagawa, Y.; Dong, H. An Innovative Industry Program in A New Era of Multimedia with Generative AI. In Proceedings of the 32nd ACM International Conference on Multimedia, Melbourne, Australia, 28 October–1 November 2024; pp. 11125–11126.

54. Khan, W.A.; Chung, S.H.; Awan, M.U.; Wen, X. Machine learning facilitated business intelligence (Part I): Neural networks learning algorithms and applications. *Ind. Manag. Data Syst.* **2019**, *1*, 164–195. [CrossRef]
55. Taiwo, R.; Bello, I.T.; Abdulai, S.F.; Yussif, A.M.; Salami, B.A.; Saka, A.; Zayed, T. Generative AI in the Construction Industry: A State-of-the-art Analysis. *arXiv* **2024**, arXiv:2402.09939.
56. Alt, T.; Ibisch, A.; Meiser, C.; Wilhelm, A.; Zimmer, R.; Berghoff, C.; Droste, C.; Karschau, J.; Laus, F.; Plaga, R.; et al. Generative AI Models: Opportunities and Risks for Industry and Authorities. *arXiv* **2024**, arXiv:2406.04734.
57. Wen, J.; Kang, J.; Niyato, D.; Zhang, Y.; Mao, S. Sustainable Diffusion-based Incentive Mechanism for Generative AI-driven Digital Twins in Industrial Cyber-Physical Systems. *EEE Trans. Ind. Cyber-Physical Syst.* **2024**, *3*, 139–149. [CrossRef]
58. Wang, Y.; Chung, S.H.; Khan, W.A.; Wang, T.; Xu, D.J. ALADA: A lite automatic data augmentation framework for industrial defect detection. *Adv. Eng. Inform.* **2023**, *58*, 102205. [CrossRef]
59. Khan, W.A. Balanced weighted extreme learning machine for imbalance learning of credit default risk and manufacturing productivity. *Ann. Oper. Res.* **2023**. [CrossRef]
60. Huang, Y.; Zhang, J.; Chen, X.; Lam, A.H.F.; Chen, B.M. From Simulation to Prediction: Enhancing Digital Twins with Advanced Generative AI Technologies. In Proceedings of the ICCA, Reykjavík, Iceland, 18–21 June 2024; pp. 490–495.
61. Huang, X.; Yang, H.; Zhou, C.; He, M.; Shen, X.; Zhuang, W. When Digital Twin Meets Generative AI: Intelligent Closed-Loop Network Management. *arXiv* **2024**, arXiv:2404.03025. [CrossRef]
62. Kishore, A.; Thorve, S.; Marathe, M.V. A Generative AI Technique for Synthesizing a Digital Twin for U.S. Residential Solar Adoption and Generation. *arXiv* **2024**, arXiv:2410.08098.
63. Du, B.; Du, H.; Liu, H.; Niyato, D.; Xin, P.; Yu, J.; Qi, M.; Tang, Y. YOLO-Based Semantic Communication With Generative AI-Aided Resource Allocation for Digital Twins Construction. *IEEE Internet Things J.* **2024**, *11*, 7664–7678. [CrossRef]
64. Wang, J.; Li, Y.; Zhou, S.; Zhang, Y.; Xiong, X.; Zhai, W. Traceability and Performance Optimization: Application of Generative AI, Digital Twin, and DRL in the Recycling Process of WEEE. *IEEE Internet Things Mag.* **2024**, *7*, 22–28. [CrossRef]
65. Chai, H.; Wang, H.; Li, T.; Wang, Z. Generative AI-Driven Digital Twin for Mobile Networks. *IEEE Netw.* **2024**, *38*, 84–92. [CrossRef]
66. Tao, Z.; Xu, W.; Huang, Y.; Wang, X.; You, X. Wireless Network Digital Twin for 6G: Generative AI as a Key Enabler. *IEEE Wirel. Commun.* **2024**, *31*, 24–31. [CrossRef]
67. Zhang, L.; Sun, H.; Zeng, Y.; Hu, R.Q. Spatial Channel State Information Prediction With Generative AI: Toward Holographic Communication and Digital Radio Twin. *IEEE Netw.* **2024**, *38*, 93–101. [CrossRef]
68. Xu, H.; Omitaomu, F.; Sabri, S.; Zlatanova, S.; Li, X.; Song, Y. Leveraging generative AI for urban digital twins: A scoping review on the autonomous generation of urban data, scenarios, designs, and 3D city models for smart city advancement. *Urban Inform.* **2024**, *3*, 29. [CrossRef]
69. Tomar, P.; Grover, V. Transforming the Energy Sector: Addressing Key Challenges through Generative AI, Digital Twins, AI, Data Science and Analysis. EAI Endorsed Trans. *Energy Web* **2023**, 10. [CrossRef]
70. Xia, Y.C.; Xiao, Z.W.; Weyrich, M. Generation of Asset Administration Shell With Large Language Model Agents: Toward Semantic Interoperability in Digital Twins in the Context ofIndustry4.0. *IEEE Access* **2024**, *12*, 84863–84877. [CrossRef]
71. Mateev, M. Implementing Hybrid (AI and Data Analytics) Solutions for Optimal Performance and Cost Optimization for Image Analysis with GPT-4 Turbo with Vision for Predictive Analysis. In Proceedings of the World Multi-Conference on Systemics, Cybernetics and Informatics, WMSCI, Orlando, FL, USA, 9–12 September 2024; pp. 74–80.
72. Rojek, I.; Mikołajewski, D.; Mroziński, A.; Macko, M. Machine Learning- and Artificial Intelligence-Derived Prediction for HomeSmart Energy Systems with PV Installation and Battery Energy Storage. *Energies* **2023**, *16*, 6613. [CrossRef]
73. Xu, D.; Zhang, D.; Yang, G.; Yang, B.; Xu, S.; Zheng, L.; Liang, C. Survey for Landing Generative AI in Social and E-commerce Recsys—The Industry Perspectives. *arXiv* **2024**, arXiv:2406.06475.
74. Lykov, A.; Altamirano Cabrera, M.; Konenkov, M.; Serpiva, V.; Gbagbe, K.F.; Alabbas, A.; Fedoseev, A.; Moreno, L.; Khan, M.H.; Guo, Z.; et al. Industry 6.0: New Generation of Industry driven by Generative AI and Swarm of Heterogeneous Robots. *arXiv* **2024**, arXiv:2409.10106.
75. Wan, H.; Zhang, J.; Chen, Y.; Xu, W.; Feng, F. Generative AI Application for Building Industry. *arXiv* **2024**, arXiv:2410.01098.
76. Bickel, S.; Goetz, S.; Wartzack, S. Symbol Detection in Mechanical Engineering Sketches: Experimental Study on Principle Sketches with Synthetic Data Generation and Deep Learning. *Appl. Sci.* **2024**, *14*, 6106. [CrossRef]
77. Lahnsteiner, L.; Größbacher, D.; Bürger, M.; Zauner, G. Automatic Object Detection in Radargrams of Multi-Antenna GPR Systems Based on Simulation Data for Railway Infrastructure Analysis. *Appl. Sci.* **2024**, *14*, 3521. [CrossRef]
78. Serôdio, C.; Mestre, P.; Cabral, J.; Gomes, M.; Branco, F. Software and Architecture Orchestration for Process Control in Industry 4.0 Enabled by Cyber-Physical Systems Technologies. *Appl. Sci.* **2024**, *14*, 2160. [CrossRef]
79. Duchanoy, C.A.; Calvo, H.; Moreno-Armendáriz, M.A. ASAMS: An Adaptive Sequential Sampling and Automatic Model Selection for Artificial Intelligence Surrogate Modeling. *Sensors* **2020**, *20*, 5332. [CrossRef] [PubMed]

80. Martinez, E.M.; Ponce, P.; Macias, I.; Molina, A. Automation Pyramid as Constructor for a Complete Digital Twin, Case Study: A Didactic Manufacturing System. *Sensors* **2021**, *21*, 4656. [CrossRef] [PubMed]
81. Huang, Z.; Shen, Y.; Li, J.; Fey, M.; Brecher, C. A Survey on AI-Driven Digital Twins in Industry 4.0: Smart Manufacturing and Advanced Robotics. *Sensors* **2021**, *21*, 6340. [CrossRef]
82. Jin, J.; Xu, H.; Leng, B. Adaptive Points Sampling for Implicit Field Reconstruction of Industrial Digital Twin. *Sensors* **2022**, *22*, 6630. [CrossRef]
83. Singh, R.; Akram, S.V.; Gehlot, A.; Buddhi, D.; Priyadarshi, N.; Twala, B. Energy System 4.0: Digitalization of the Energy Sector with Inclination towards Sustainability. *Sensors* **2022**, *22*, 6619. [CrossRef]
84. Tang, X.; Wang, Z.; Deng, L.; Wang, X.; Long, J.; Jiang, X.; Jin, J.; Xia, J. A Review of the Intelligent Optimization and Decision in Plastic Forming. *Materials* **2022**, *15*, 7019. [CrossRef]
85. Medhi, T.; Hussain, S.A.I.; Roy, B.S.; Saha, S.C. An intelligent multi-objective framework for optimizing friction-stir welding process parameters. *Appl. Soft Comput.* **2021**, *104*, 107190. [CrossRef]
86. Wang, Q.; Ma, H.; Wei, W.; Li, H.; Chen, L.; Zhao, P.; Zhao, B.; Hu, B.; Zhang, S.; Zheng, Z.; et al. Attention Paper: How Generative AI Reshapes Digital Shadow Industry? *arXiv* **2023**, arXiv:2305.18346.
87. Fu, B.; Hadid, A.; Damer, N. Generative AI in the context of assistive technologies: Trends, limitations and future directions. *Image Vis. Comput.* **2025**, *154*, 105347. [CrossRef]
88. Doron, G.; Genway, S.; Roberts, M.; Jasti, S. New Horizons: Pioneering Pharmaceutical R&D with Generative AI from lab to the clinic—An industry perspective. *arXiv* **2023**, arXiv:2312.12482.
89. Sauvola, J.J.; Tarkoma, S.; Klemettinen, M.; Riekki, J.; Doermann, D.S. Future of software development with generative AI. *Autom. Softw. Eng.* **2024**, *31*, 26. [CrossRef]

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Applied Sciences Editorial Office
E-mail: applsci@mdpi.com
www.mdpi.com/journal/applsci

Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-4398-5