Reliability Engineering and System Safety 254 (2025) 110560

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Self-adaptive fault diagnosis for unseen working conditions based on digital twins and domain generalization

Mehdi Saman Azari [a,*], Stefania Santini [b], Farid Edrisi [a], Francesco Flammini [c]

[a] Department of Computer Science and Media Technology, Linnaeus University, Växjo, Sweden
[b] Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy
[c] School of Innovation, Design and Engineering, Mälardalen University, Sweden

ARTICLE INFO

Keywords:
Fault diagnosis system
MAPE-K
Domain generalization
Data augmentation
Rotating machine
Digital twin
Unseen working conditions

ABSTRACT

In recent years, intelligent fault diagnosis based on domain adaptation has been used to address domain shifts in cyber–physical systems; however, the need for acquiring target data sufficiently limits their applicability to unseen working conditions. To overcome such limitations, domain generalization techniques have been introduced to enhance the capacity of fault diagnostic models to operate under unseen working conditions. Nevertheless, existing approaches assume access to extensive labeled training data from various source domains, posing challenges in real-world engineering scenarios due to resource constraints. Moreover, the absence of a mechanism for updating diagnostic models over time calls for the exploration of self-adaptive generalized diagnosis models that are capable of autonomous reconfiguration in response to new unseen working conditions. In such a context, this paper proposes a self-adaptive fault diagnosis system that combines several paradigms, namely Monitor-Analyze-Plan-Execute over a shared Knowledge (MAPE-K), Domain Generalization Network Models (DGNMs), and Digital Twins (DT). The MAPE-K loop enables run-time adaptation to dynamic industrial environments without human intervention. To address the scarcity of labeled training data, digital twins are used to generate supplementary data and continuously tune parameters to reflect the dynamics of new unseen working conditions. DGNM incorporates adversarial learning and a domain-based discrepancy metric to enhance feature diversity and generalization. The introduction of multi-domain data augmentation enhances feature diversity and facilitates learning correlations among multiple domains, ultimately improving the generalization of feature representations. The proposed fault diagnosis system has been evaluated on three publicly available rotating machinery datasets to demonstrate its higher performance in cross-work operation and cross-machine tasks compared to other state-of-the-art methods.

1. Introduction

The evolution of industrial sensing technologies and computational capabilities has significantly propelled data-driven methodologies in mechanical equipment fault diagnosis within the cyber–physical framework [1]. Herein, although Deep Learning (DL) approaches surpass traditional Machine Learning (ML) in extracting implicit fault-related features from large datasets, they still rely on the assumption of consistent data distribution across training and testing phases. This strict assumption barely holds in real-world applications due to the highly dynamic nature of mechanical equipment, which leads to the constant change of work conditions and data distribution. The above discrepancy, commonly known as domain shift, poses a crucial challenge, as DL models trained in specific work conditions may struggle to perform well in different conditions [2]. In attempting to overcome the domain shifts problem, the DL approach necessitates the collection of an extensive amount of data specific to the new working conditions and the training of the model from scratch. However, this process is resource-intensive and may not always be practical, thus confirming the need for alternative strategies to enhance model adaptability across varying working conditions [3].

To address the issue of domain shift, Domain Adaptation (DA) techniques have emerged to enhance the resilience and adaptability of DL models [2]. DA strategies seek to minimize the differences between training (source) and testing (target) data. Despite their effectiveness, the practicality of DA-based fault diagnosis methods is contingent on the availability of target domain data for the adaptation process. However, in real situations — especially when mechanical equipment

---

* Corresponding author.
E-mail addresses: mehdi.samanazari@lnu.se (M.S. Azari), stefania.santini@unina.it (S. Santini), farid.edrisi@lnu.se (F. Edrisi), francesco.flammini@mdu.se (F. Flammini).

https://doi.org/10.1016/j.ress.2024.110560
Received 29 February 2024; Received in revised form 3 August 2024; Accepted 4 October 2024

operates under new working conditions — only data representing normal health conditions are accessible during training, while faulty data remain inaccessible. These conditions, commonly known as the unseen working conditions, are prevalent in highly dynamic mechanical equipment and ever-changing environments, such as rotating machinery [4], consequently, the applicability of most DA methods is restricted to offline cross-domain fault diagnosis tasks.

In response to the open challenge posed by unseen working conditions, researchers have increasingly focused on Domain Generalization (DG) techniques within the realm of intelligent fault diagnosis systems. These techniques aim to generalize the applicability of diagnostic knowledge in the source domains to the unseen working conditions [5]. Despite achieving promising results in generalization, existing approaches based on DG for unseen working conditions face the following challenges. Firstly, many DG approaches aim to learn domain-invariant representations by consolidating all domains into a single latent space, often overlooking critical domain-specific features essential for precise diagnosis [6]. This oversight not only restricts the diversity of features but also increases the risk of overfitting to the characteristics of source domains. Consequently, the model may struggle to generalize effectively to new, unseen working conditions. Therefore, there is a pressing need to advance DG-based fault diagnosis systems to better incorporate and utilize domain-specific information, thereby enhancing their ability to generalize across diverse industrial working condition. Secondly, to develop models with sufficient generalization capability, these approaches rely on the assumption of access to a huge amount of labeled training data from different source domains with diverse data distributions. However, in practical engineering scenarios, gathering such extensive training datasets from identical machines under different working conditions or from different but similar machines presents challenges. These challenges stem from the high costs associated with data collection, including equipment setup, maintenance, and operational downtime. Furthermore, the diversity and variability inherent in real-world industrial settings often lead to sparse or incomplete datasets, particularly concerning rare failure events or specific operational scenarios. This scarcity of data can severely limit the generalizability of diagnostic models trained solely on limited and homogeneous datasets. Therefore, exploring a reliable supplementary source to provide a training dataset becomes essential. Thirdly, industrial machinery operates in dynamic and continuously evolving environments, necessitating diagnostic models capable of adapting and updating their assessments in response to new data and evolving working conditions. Traditional approaches often lack mechanisms for ongoing model updates, relying instead on static datasets that may quickly become outdated. Therefore, there is an urgent need to explore self-adaptive generalized diagnosis models that can autonomously monitor changes in operational conditions and adjust their diagnostic strategies accordingly. Such models would not only enhance reliability over time but also ensure that diagnostic systems remain effective and accurate in the face of evolving industrial environments.

To tackle and solve the above-mentioned open issues, this paper proposes a self-adaptive fault diagnosis system, combining MAPE-K, domain generalization network model, and digital twin techniques in an highly integrated fashion. Herein the DGNM is exploited for dealing with the problems related to discarding domain-specific features and overfitting due to limited feature diversity. To this aim a combination of techniques is leveraged, including multi-domain data augmentation, adversarial learning, and domain-based discrepancy metric. More specifically, the multi-domain augmentation employs a Dirichlet distribution-based mixup technique, creating augmented data that blends features from various domains. This enriched data is then utilized in adversarial learning and a domain-based discrepancy metric to extract more diverse features, improving the generalization of the network. The DT is instead embedded in the proposed platform as a reliable data supplier so solving the issue of obtaining sufficient training datasets from actual machines. Indeed, the DT model is continuously tuned, allowing reflection of the operational dynamics of the new unseen working condition and providing a rich source of diagnostic knowledge. Finally, to cope with the dynamic nature of industrial machinery environments, the MAPE-K loop for self-adaptation is incorporated into the developed fault diagnosis system. This loop facilitates real-time adaptation to the new unseen work conditions, thus reducing the need for direct human intervention and enhancing system autonomy.

The key novel contributions of this work can be therefore summarized as follows:

- Adoption of the MAPE-K loop for self-adaptive fault diagnosis system.
- Integration of digital twins and appropriate updating strategy to provide reliable supplementary source domain data.
- Leveraging multi-domain data augmentation to improve the generalization ability of the fault diagnosis system to out-of-distribution data.
- Embedding domain discriminator and domain-based discrepancy metric modules into a DL-based generalization network, named DGNM, to learn domain-invariant and discriminative features from multiple source domains.
- Providing an experimental evaluation of the approach on a rotating machinery case-study using three publicly available datasets for cross-work operation and cross-machine tasks.

The structure of this paper is as follows: Section 2 offers a concise review of related research, followed by preliminaries and background information in Section 3. In Section 4, a comprehensive description of the proposed self-adaptive fault diagnosis system is provided. The application of this system to a rotating machine as a case study is elucidated in Section 5. Furthermore, Section 6 presents the experimental results along with their analysis. Finally, in Section 7, the paper is concluded, and potential directions for future research are discussed.

2. Related works

This section reviews the state-of-the-art concepts relevant to this paper. To enhance organization and clarity, each concept—namely traditional machine learning and deep learning, domain adaptation, domain generalization, digital twin, and MAPE-K—will be discussed individually.

2.1. Traditional machine learning and deep learning

In the predictive maintenance field, traditional ML and DL have proven effective in developing fault diagnosis systems due to their capabilities in failure detection and isolation. For instance, Rajabi et al. [7] proposed a fault diagnosis system that combines traditional ML algorithms with signal processing techniques, achieving a better balance in terms of data imbalance, and computational resource usage. Similarly, Moradi et al. [8] introduced a novel mathematical architecture integrating Bayesian Networks (BN) and DL for condition and operational risk monitoring of Complex Engineering Systems (CES). Moreover, Zhao et al. [9], developed a model-driven deep unrolling method to design interpretable DL models against noise attacks for fault diagnose system to achieve ante-hoc interpretability. However, since mechanical equipment operates in highly dynamic environments, this domain shift poses a significant challenge for existing data-driven fault diagnosis systems based on traditional ML and DL.

2.2. Domain adaptation

Recent advancements in domain adaptation methodologies have significantly addressed the challenges posed by domain shifts. These methods are generally categorized into statistics-based and domain adversarial-based approaches. Statistics-based approaches, such as Correlation Alignment [10] and Maximum Mean Discrepancy (MMD) [11], integrate discrepancy metrics into their objective functions to minimize domain distribution differences. On the other hand, domain adversarial-based approaches leverage adversarial frameworks to identify transferable features across domains [12]. Notable scenarios in this latter field include Transfer from Identical Machine (TIM) [13] and Transfer from Different related Machines (TDM) [14], where the significant challenge is due to the scarcity of faulty data from Real Machines (RM), as machine failures are infrequent and sometimes rare. A popular alternative is transferring diagnostic knowledge from Laboratory to Real-world Machines (TLRM) [15], although it is often time-consuming and costly due to the need to induce various faults in lab machines. Consequently, the Transfer from Virtual to Real Machine (TVRM) approach has gained popularity in diagnostics. For example, Li et al. [16] demonstrated the potential of using simulated data for anomaly detection in physical processes. However, DA-based fault diagnosis methods heavily depend on the availability of data from the target domain.

2.3. Domain generalization

To address the challenge of unseen target domains, research is exploring domain generalization approaches, which can be categorized into data augmentation, regularization, and domain-invariant representation learning. Data augmentation techniques, such as those explored by Fan et al. [17], integrate domain discrepancy and domain discriminators to diagnose faults and aim to enhance generalization by diversifying source data. However, traditional mixup techniques used by Fan et al. have limitations as they generate samples only between two domains, neglecting consideration of other domains. Shi et al. [18] developed multisource augmentation combined with adversarial training to boost feature diversity and learn correlations among multiple domains, thereby enhancing generalization of feature representation. Regularization methods, such as Yang et al.'s [19] approach, prevent overfitting by applying regularization techniques that optimize both center loss and softmax loss for improved fault diagnosis under unknown conditions. Domain-invariant representation learning methods, such as the adversarial domain-invariant generalization (ADIG) method by Chen et al. [20], focus on extracting consistent representations across different source domains using feature normalization to boost diagnostic accuracy. Wang et al. [21] designed multiple domain-specific auxiliary classifiers to explicitly learn domain-specific features among different source domains, then used a convolutional auto-encoder module to remove learned domain-specific features through prediction uncertainty and reconstruction loss. Despite their effectiveness, DG methods often overlook specific private features crucial for fault diagnosis in unseen working conditions and generally require large, varied, labeled training data from different source domains, which can be challenging and costly to collect in real-world engineering settings.

In scenarios where limited training datasets are available in the source domain, Jun et al. [22] proposed a single domain generalization model, termed multi-scale style generative and adversarial contrastive networks (MSG-ACN), which learns diagnostic knowledge from a single working condition and generalizes it to new conditions. However, single domain generalization methods often struggle in real-world applications due to limited generalization to new, unseen domains, leading to performance drops, overfitting, and sensitivity to domain shifts.

2.4. Digital twin

To address these challenges and provide a practical alternative, digital twins have been introduced to simulate fault conditions for training data generation [23]. In general, it provides a more targeted approach to generating data that closely matches the specific characteristics and operational conditions of the target machine condition, thereby improving the performance and reliability of fault diagnosis systems. For example, Ma et al. [24] developed a DT model using the finite element method, combining simulation and real-world data for DA. Similarly, Don et al. [25] created a dynamic bearing model to generate extensive synthetic data, demonstrating its applicability to real rotating machinery in DA. Jia et al. [26] developed DT models for DG, incorporating various combinations of simulation and real-world data to enhance diagnostic capabilities. However, a strategy to update DT models is necessary to reflect real-world changes and maintain accuracy in alignment with actual systems. In such a view, Xia et al. [27] presented an intelligent fault diagnosis framework using a DT model and deep transfer learning, where the DT model is updated with real-time data from the physical asset. However, since the machine's working conditions are dynamic and change over time, it is crucial to develop self-adaptive diagnostic models that automatically update both the diagnostic and DT models to adjust to new, unseen working conditions. Recent approaches, such as those by Mishra et al. [28] and Huo et al. [29], address this need, though models tailored for new unseen working conditions remain essential.

2.5. MAPE-K

The MAPE-K loop, standing for Monitor, Analyze, Plan, and Execute over a shared Knowledge base, is the widely recognized reference model to realize self-adaptation in software-intensive systems [30]. It originated from the need for computing systems to automatically handle complex management tasks without human intervention [31]. By providing a structured approach for separating various functionalities and interactions, the MAPE-K loop operates continuously to ensure that the system can adapt to changes and maintain its goals, autonomously [32].

Many successive applications of MAPE-K have been implemented in various domains, showing its versatility. For example, in robotics, the MAPE-K loop has been utilized to enhance the autonomy and efficiency of robotic systems, allowing them to adapt to dynamic environments and perform complex tasks with minimal human oversight [33]. In smart factories, it is utilized in optimizing production processes and ensuring uninterrupted operations by adapting to varying manufacturing conditions and detecting anomalies [34]. Similarly, in the Internet of Things (IoT) applications, MAPE-K supports the dynamic management of connected devices to ensure that they can respond to changing contexts and user requirements effectively [35]. However, to the authors' knowledge, no research currently addresses the development of a structured self-adaptive fault diagnosis system, specifically, designed for new and unseen working conditions.

Finally, to tackle and solve all the above open issues and fill the gaps in the current literature, this paper introduces a new self-adaptive fault diagnosis framework, that combines multiple techniques in a fully integrated fashion. To clearly outline the contributions of this paper, Table 1 presents a comparative summary for disclosing the enhancements of the proposed approach w.r.t. state-of-the-art.

3. Preliminaries

3.1. Building a diagnostic digital twin

Developing DTs is a challenging task since it requires finding a balance between sufficiently acceptable simulation performance and the complexity of real machines. In other words, the level of abstraction

Table 1
Related Work Summary versus our proposed approach.

	Different distribution across S & T	Different label space across S & T	Use DT as a source domain	Target domain data			Self-adaptive
				Full	limited	Unavailable	
Rajabi et al. [7], Zhao et al. [9] Moradi et al. [8]	✗	✗	✗	✓	–	–	✗
Jiang et al. [10], Zenghui et al. [11] Liu et al. [13], Chen et al. [12]	✓	✗	✗	–	✓	–	✗
Yang et al. [15], Kim et al. [14]	✓	✓	✗	–	✓	–	✗
Li et al. [16], ma et al. [24] Dong et al. [25], Xia. [27]	✓	✗	✓	–	✓	–	✗
Fan et al. [17], Yang et al. [19] Chen et al. [20], Shi et al. [18] Wang et al. [21], Jun et al. [22]	✓	✗	✗	–	–	✓	✗
Jia et al. [26]	✓	✗	✓	–	–	✓	✗
mishra et al. [28], Huo et al. [29]	✗	✗	✗	✓	–	–	✓
Proposed approach	✓	✗	✓	–	–	✓	✓

must be selected constructively to cover all the relevant details. For diagnostic goals, it is necessary to define a detailed model of the most critical components, rather than a detailed representation of the entire machine.

In the following, in order to facilitate and offer a standard framework, the design process is broken down into the following 4 high-level steps:

1. **Select the critical components**. At first, it is necessary to identify which components must be subject to condition monitoring, which can be related to their criticality, in terms of impact on machine safety and performance, and fault susceptibility.
2. **Derive the dynamic behavior of the system under observation**. This step focuses on developing the dynamic model of the critical components identified in the previous step. We remark that it is only necessary to emulate the dynamic behavior of the critical machine components. The modeling of the critical component should follow a white-box approach, meaning that the exact component's working mechanisms should be used, with any rules and restrictions of the real machine.
3. **Define and model the faults**. This step addresses the integration of proper fault mechanisms into the dynamic model developed in the previous step, in order to inject and simulate faults that could lead to machine failures. In general, fault mechanisms have to be integrated so as to affect the dynamic behavior of the machinery and provide faulty datasets.
4. **Specify the virtual sensors**. This step is about defining and specifying the data that needs to be collected through DT simulation by means of virtual sensors. Virtual sensors, if it is necessary their dynamics, should be integrated into the DT of the machine. This enables the user to collect data that arises throughout the simulation of the DT model.

Once the digital model that predicts the value of some target variables is created and initially calibrated, its prediction abilities are boosted via continuous updating and tuning according to working condition data and measurements.

3.2. Domain generalization problem setting

The goal of domain generalization is to train a model on multiple known labeled source domains $\{D^{s,k}\}_{k=1}^{K}$ so that it can effectively operate on an unseen target domain D^t. In this notation, s and t refer to source and target, respectively. Each source domain $D^{s,k} = (X^{s,k}, Y^{s,k}, d^{s,k})$, indexed by k, comprises a set of data points $X^{s,k} = \{x_i^{s,k}\}_{i=1}^{n_{s,k}}$ and their corresponding labels $Y^{s,k} = \{y_i^{s,k}\}_{i=1}^{n_{s,k}}$, drawn from a distinct distribution $P^{s,k}$. The domain label $d^{s,k}$ ranges within $\{1, 2, \ldots, K\}$ and $n_{s,k}$ signifies the sample number in each domain. The label space for each category in a source domain is defined as $\mathscr{Y}^{s,k} = \{1, 2, \ldots, N_c\}$, where N_c represents the total number of categories.

The target domain, represented as $D^t = (X^t)$, shares the same category label space as the source domains, meaning there is no category shift considered in this study ($\mathscr{Y}^{s,k} = \mathscr{Y}^t$). The target domain's distribution is P^t, distinct from those of the source domains ($P^{s,1} \neq P^{s,2} \neq \ldots \neq P^{s,K} \neq P^t$).

The challenge lies in training the model using data from these K source domains, supplemented by augmented data, to ensure it gains sufficient generalization ability. This will enable the model to perform effectively in diagnostic tasks on the unseen target domain D^t, despite the differences in data distribution between the source and target domains

3.3. Mixup

Enhancing the diversity and quantity of training data is a well-recognized method for improving the generalization performance of intelligent diagnostic models. Data augmentation is a widely employed strategy for achieving this, particularly when faced with limited training data.

One particularly effective method for data augmentation is Mixup. This technique generates diverse data by creating augmented samples through a mathematical combination of two training samples and their associated labels, denoted as (x_i, y_i) and (x_j, y_j), as defined by the equation:

$$\begin{aligned} \tilde{x} = &\quad \mathcal{M}(x_i, x_j; \mu) = \mu x_i + (1-\mu) x_j \\ \tilde{y} = &\quad \mathcal{M}(y_i, y_j; \mu) = \mu y_i + (1-\mu) y_j \end{aligned} \tag{1}$$

Here, $\mathcal{M}$ signifies the convex combination function, and μ is a randomly selected mixup factor drawn from a beta distribution $Beta(\alpha, \alpha)$ for $\alpha > 0$. Mixup is valuable because it can capture relationships between samples from different domains and categories, emphasizing linear correlations between training examples. Furthermore, it compensates for missing information, like domain-specific features, between different domain samples, enabling models to learn diverse features and explore richer, domain-invariant representations.

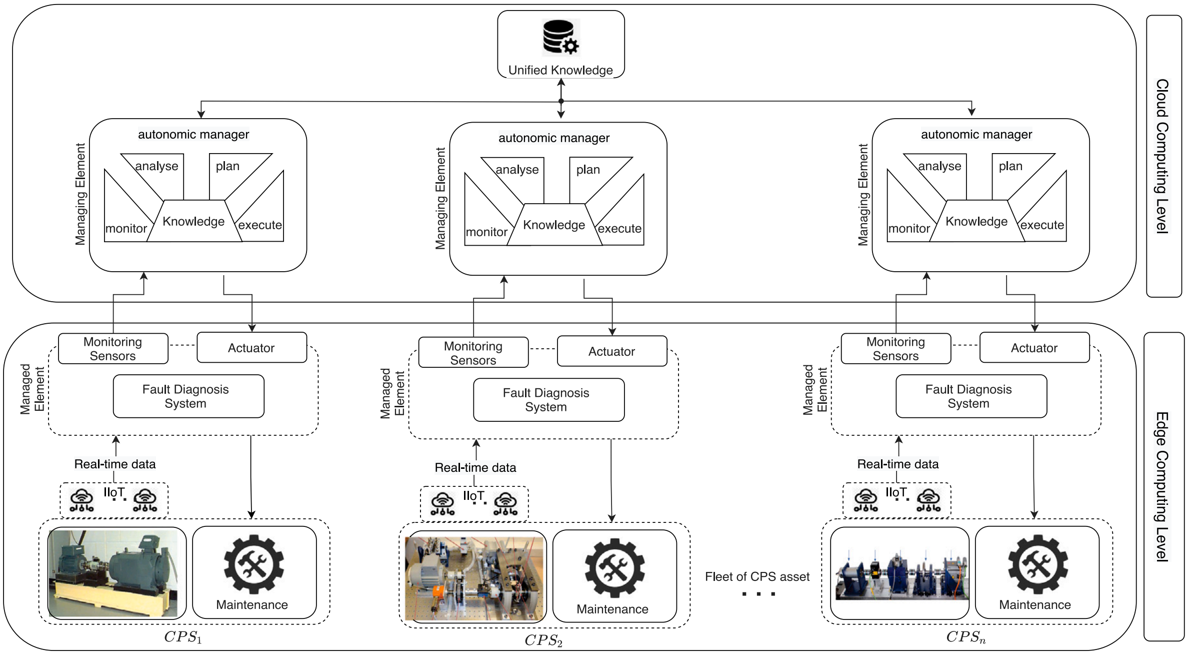

Fig. 1. Reference architecture for self-adaptive fault diagnosis.

4. Self-adaptive fault diagnosis system design

The fault diagnosis system (FDS) is used to detect latent defects, isolate them, and identify the specific type of fault. In real-world applications, the real machine performance is influenced by varying work operations and environmental conditions. In those conditions, the distribution of the real machine data is generally changing over new working conditions. The traditional FDS, based on DL and domain adaptation techniques, needs to access both healthy and faulty data in new work condition at the training phase to either keep its functionality or adapt its behavior; however, only health conditions are normally available during the model training, while faulty data are often inaccessible. To overcome this problem, eliminating dependencies and enabling the generalization of diagnostic models to new working conditions is required. To this end, we introduce self-adaptation with the main aim of enabling the autonomous reconfiguration of FDS for the new unseen working conditions without having access to its faulty data.

The reference framework of the proposed self-adaptive FDS in the context of emerging technologies and computing paradigms for cyber–physical systems is shown in Fig. 1. It comprises managed and managing element layers, whereas the managing element layer is responsible for implementing self-adaptation. The MAPE-K loop is purposely tailored for designing, developing, and realizing FDS self-adaptation, as illustrated in Fig. 2. The MAPE-K loop consists of the following four essential Functional Components (FC):

- FC_1: **Monitor.** This component acquires data from the managed element and the environment, processing it through a probe to update the knowledge repository.
- FC_2: **Analyzer.** Utilizing the updated knowledge, the Analyzer determines whether adaptation is necessary and analyzes potential scenarios for adaptation.
- FC_3: **Planner.** If adaptation is deemed necessary, the Planner executes the most suitable adaptation scenarios, each comprising two or more adaptation actions.
- FC_4: **Executor.** The Executor implements the developed FDS through an effector, adjusting the managed element as per the devised plan.

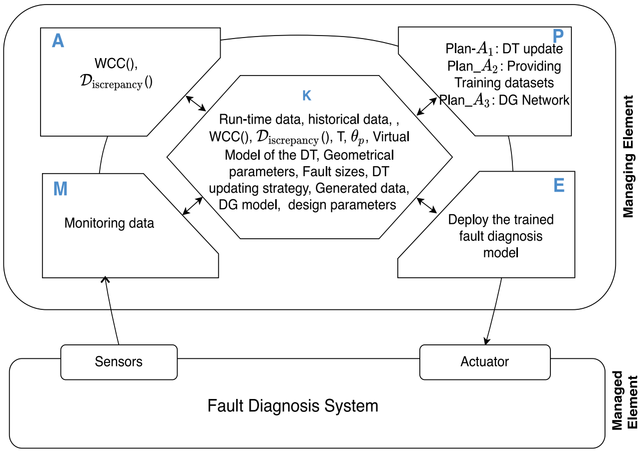
A
WCC(),
$\mathcal{D}_{\text{iscrepancy}}()$
P
Plan-A_1: DT update
Plan_A_2: Providing Training datasets
Plan_A_3: DG Network
K
Run-time data, historical data, , WCC(), $\mathcal{D}_{\text{iscrepancy}}()$, T, θ_p, Virtual Model of the DT, Geometrical parameters, Fault sizes, DT updating strategy, Generated data, DG model, design parameters
M
Monitoring data
E
Deploy the trained fault diagnosis model
Managing Element
Sensors
Actuator
Fault Diagnosis System
Managed Element

Fig. 2. Self-adaptive fault diagnosis using MAPE-K.

The managing element uses a shared **Knowledge** among those components to collaboratively realize self-adaptation. In the following subsections, we will provide details about each MAPE-K component.

4.1. Monitor

Monitoring is the initial stage of the loop and has the responsibility to provide essential data for the subsequent stages. At the monitoring stage, data is collected at regular intervals from sensing units in the *FDS*. The *Monitor* component processes data from sensors for homogenization and noise removal, making it suitable for the *Analyze* and *Planner* stages. The processed information is then stored in the shared Knowledge.

4.2. Analyzer

After updating the shared *Knowledge* with the machine's most recent operational data through the *Monitor*, the *Analyzer* utilizes those data to detect any alterations in the machine's work condition. To measure alterations in the machine's work conditions, we need to introduce some metrics. First of all, we define the index WCC (Work Condition Change) as follows:

$$WCC = \text{TRUE} \quad \text{if} \quad |WC_t - WC_{t-1}| > T, \tag{2}$$

where WC_t and WC_{t-1} represent the work condition in the current and one preceding sample time, respectively. T is the design parameter that is defined by the system designer and is the threshold for preventing false alarms in the case of very small changes. In other words, $WCC = TRUE$ means a variation in the work condition is detected, and hence adaptation is required.

To measure the discrepancy between the response of the DT (p_{DT}) and the one of the real machine (p_R) in a new working condition, the following index is defined:

$$\mathcal{D}_{\text{iscrepancy}} = \begin{cases} L & \text{if } E(p) > \theta_p, \\ S & \text{otherwise}, \end{cases} \qquad where \quad E(p) = \{p_{DT} - p_R\} \tag{3}$$

where $E(p)$ represents the calculated error between the response signals, θ_p is the adjusted threshold for determining the acceptable error between the response signals, L implies a large discrepancy between the response of the DT and the real machine, and S indicates a small and acceptable discrepancy. The results of the computation of those indices are stored by the *Analyzer* in the *Knowledge* for future reference.

4.3. Planner

The indices computed by the *Analyzer* and stored in the *Knowledge* serve as triggers for the *Planner* to guide this functional component. Indeed, the value of WCC activates or deactivates the *Planner*, and the value of $\mathcal{D}_{\text{iscrepancy}}$ drives the appropriate response to system changes. Specifically, when a significant discrepancy is detected (i.e., $\mathcal{D}_{\text{iscrepancy}} = L$), indicating that the current DT model is obsolete, an update is required and hence the *Planner* selects three adaptation tasks, i.e., update strategy, data generation, and domain generalization, $\mathcal{P}_{\mathrm{lan}\,A_1}$, $\mathcal{P}_{\mathrm{lan}\,A_2}$, $\mathcal{P}_{\mathrm{lan}\,A_3}$, respectively. The updating strategy detailed in Section 4.3.1 is executed in $\mathcal{P}_{\mathrm{lan}\,A_1}$, while $\mathcal{P}_{\mathrm{lan}\,A_2}$ provides the labeled training data from multiple domains with the required data distributions to enable the subsequent domain generalization network model (see Section 4.3.2). Finally, the $\mathcal{P}_{\mathrm{lan}\,A_3}$ described in Section 4.3.3 is responsible for the generalization of existing diagnostic knowledge from multiple source domains. Conversely, in the case of a limited discrepancy (i.e., $\mathcal{D}_{\text{iscrepancy}} = S$), i.e., when the DT is able to operate effectively without adjustments, only $\mathcal{P}_{\mathrm{lan}\,A_2}$ and $\mathcal{P}_{\mathrm{lan}\,A_3}$ are executed.

4.3.1. $\mathcal{P}_{\mathrm{lan}\,A_1}$: Updating the DT model

In the context of the proposed self-adaptive fault diagnosis system, updating the DT model is a critical process that is treated as an optimization problem. The main goal is to minimize discrepancies between the DT model and the evolving conditions of the real machine, ensuring the DT remains synchronized with real-world changes. As depicted in Fig. 3, a model update strategy based on parameter sensitivity analysis is introduced to align the DT model with the target real machine condition. This model update strategy is framed as an optimization problem, with the objective function aimed at minimizing these discrepancies.

To implement the DT updating concept, it is crucial to select appropriate response signals to measure the similarity between the actual behavior of the real target machine condition and the behavior predicted by the DT model. Based on the selected response signals, the optimization function for the developed update strategy can be defined as Eq. (3). The success of the update strategy hinges on selecting suitable metrics to measure the similarity between the real system's behavior and the DT model's predicted behavior. For example, vibrational data is one of the most informative signals for monitoring the health condition of industrial systems. However, comparing vibration data directly in the time domain can be challenging, necessitating a specialized metric to evaluate the similarity between the spectra derived from simulated and real-world scenarios.

After computing the similarity between the real and simulated responses based on the response signals and defined metrics, the updating parameters are selected and adjusted to reduce the difference between the RM and its DT. Once the squared norm of the error falls within an acceptable range, indicating that the DT's response aligns with the new, unseen operating conditions of the machine, the update strategy is considered successful.

4.3.2. $\mathcal{P}_{\mathrm{lan}\,A_2}$: Providing training dataset

Access to a substantial amount of labeled training data from multiple domains with different data distributions is essential to enable the DG model to generalize effectively. As a response to this necessity, $\mathcal{P}_{\mathrm{lan}\,A_2}$ focuses on providing training datasets from multiple source domains. This activity serves as a foundation for the subsequent stages within $\mathcal{P}_{\mathrm{lan}\,A_3}$, and it is divided into two steps:

- $\mathcal{P}_{\mathrm{lan}\,A_{2.1}}$ - *Definition of the multiple source domains*
  The scope of $\mathcal{P}_{\mathrm{lan}\,A_{2.1}}$ is to define a training dataset from multiple source domains. The first step is the construction of a source domain dataset by executing the DT model of the real machine. Therefore, The DT is exploited for simulations in diverse health conditions, encompassing various failure sizes. To supplement this primary dataset, we explore historical data collected from previous work operations of the target machinery, stored within *Knowledge*. However, the practical challenges of obtaining training datasets from the same test bearing under different work operations render this task unfeasible in some situations. In such cases, another reliable training dataset can be obtained from other bearings, where a wealth of diagnostic knowledge is available in cloud repositories, owing to the existence of similar machines across different industrial plants, companies, and digital platforms. That allows for the creation of a diverse training dataset, which serves as the foundation for the domain generalization process in $\mathcal{P}_{\mathrm{lan}\,A_3}$.
- $\mathcal{P}_{\mathrm{lan}\,A_{2.2}}$ - *Data Augmentation*
  The aim is to generate new samples (inter-domain samples) by using samples from different domains and thus exploring domain correlation representations. The module hence plays a dual role in augmenting the data diversity while preserving domain specificity to a certain extent, thereby enhancing the generalizability of the learned features. In addition, it ensures smoothing of data distributions between domains, fostering the learning of domain-invariant representations.

  To achieve this goal, we drew inspiration from the mixup concept (refer to Section 3.3). However, traditional mix-up techniques are limited in generating samples only between two domains, overlooking considerations for other domains. Recognizing these limitations in multi-source scenarios, we introduced multi-domain mixup, as depicted in Fig. 4. To effectively utilize information from multiple domains, we adopted the Dirichlet distribution instead of the beta distribution for sampling mixup factors.

  Choosing the Dirichlet distribution over the Beta distribution offers significant advantages, especially in multiclass and complex scenarios. The Dirichlet distribution naturally handles multiple classes by generating mixing weights that sum to one, simplifying implementation and ensuring consistency. It provides greater control over the weight distribution among samples, thereby enhancing mixing flexibility. In contrast, the Beta distribution is primarily suited for two-class scenarios and becomes less efficient when extended to multiclass problems. Thus, Dirichlet's ability to manage multiclass data naturally, coupled with its simplicity and flexibility, makes it the preferred choice for mixup techniques.

  This adjustment empowers our multi-domain mixup approach to generate samples with enriched information and diversity,

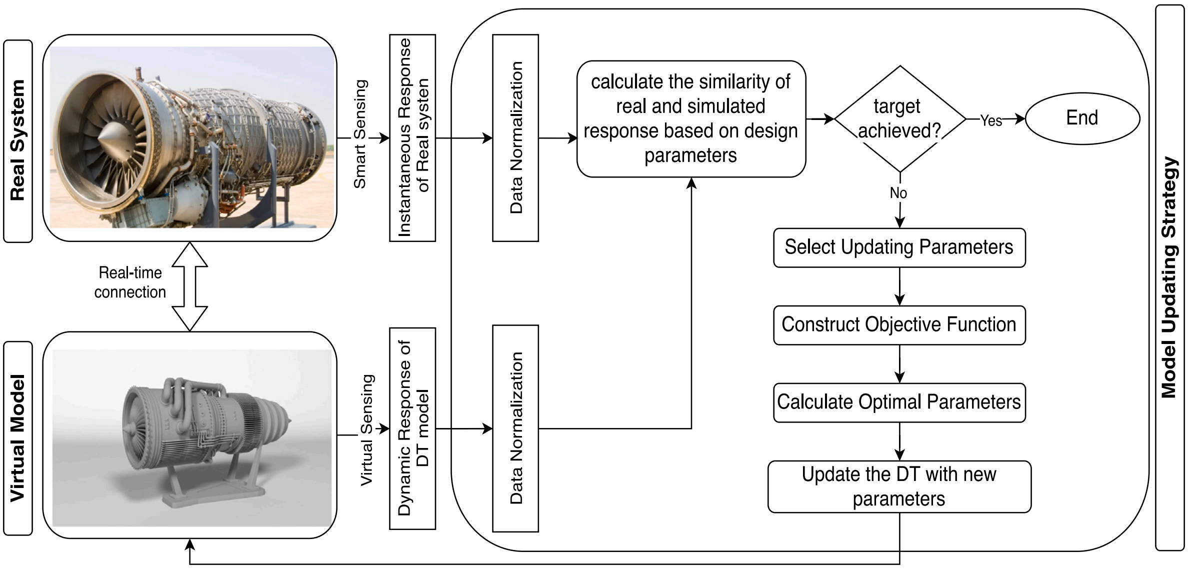
Real System
Smart Sensing
Instantaneous Response of Real system
Data Normalization
calculate the similarity of real and simulated response based on design parameters
target achieved?
Yes
End
No
Select Updating Parameters
Construct Objective Function
Calculate Optimal Parameters
Update the DT with new parameters
Real-time connection
Virtual Model
Virtual Sensing
Dynamic Response of DT model
Data Normalization
Model Updating Strategy

Fig. 3. DT update strategy.

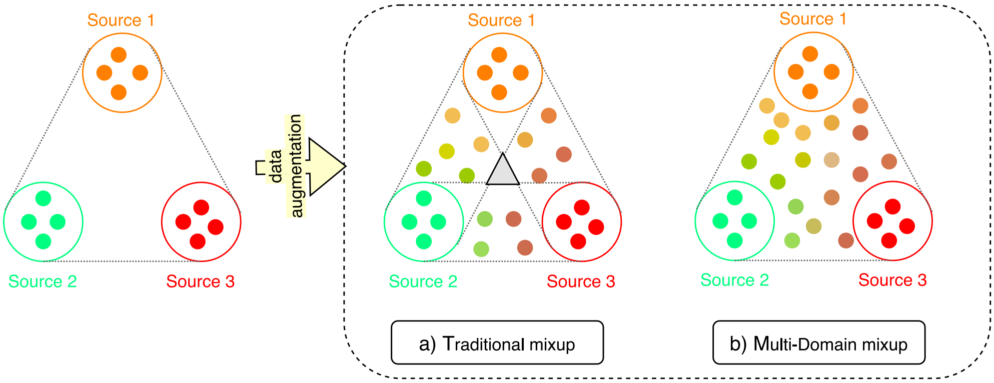

Fig. 4. Comparison between traditional mixup and multi-domain mixup.

enabling the model to extract comprehensive and generalized diagnostic insights effectively. The multisource augmentation sample denoted as $(\widetilde{x}_i^s, \widetilde{y}_i^s)$, is obtained from samples from different source domains, $(x_i^{s,1}, x_i^{s,2}, \dots, x_i^{s,K})$, using the following equation:

$$\widetilde{x}_i^s = \mathcal{M}\big(\{x_i^s\}_{k=1}^K;\mu\big) = \sum_{k=1}^{K} \mu_k x_i^{s,k} \tag{4}$$

$$\widetilde{y}_i^s = \mathcal{M}\big(\{y_i^s\}_{k=1}^K;\mu\big) = \sum_{k=1}^{K} \mu_k y_i^{s,k} \tag{5}$$

Here, μ represents the set of mixup factors randomly sampled from a Dirichlet distribution $\mu \sim Dirichlet(\alpha)$, with α being an M-tuple array in which its all elements values are set to be the same in this paper. Additionally, to promote domain-invariant representations and minimize domain discrepancy, soft domain labels corresponding to the augmented samples are generated for use in adversarial training, as described by the following equation:

$$\widetilde{d}_i^s = \mathcal{M}\big(\{d_i^s\}_{k=1}^K;\mu\big) = \sum_{k=1}^{K} \mu_k d_i^{s,k} \tag{6}$$

In summary, by emphasizing sample-level augmentation and introducing soft domain labels for adversarial training, we empower the domain generalization network to extract domain-invariant representations more effectively.

4.3.3. $\mathcal{P}_{\text{lan}_{A_3}}$: *Domain generalization network model*

The coherent integration of the domain generalization networks model within the broader self-adaptive fault diagnosis system underscores the system's capacity to generalize its diagnostic capabilities to the unseen working conditions of the target machine. More specifically, the DGNM is designed to address challenges associated with multiple distribution shifts and the unavailability of new working conditions data (faulty data). As it illustrated in Fig. 5, by leveraging the training data obtained from the earlier stages $\mathcal{P}_{\text{lan}_{A_2}}$, the DGNM focuses on learning domain-invariant feature representations that can be effectively generalized to identify faults in unseen working conditions.

To ensure the generalization and adaptability of the FDS, we employ a combination of techniques, including adversarial learning, and a domain-based discrepancy metric. The DGNM comprises a feature extractor $G(.;\theta_g)$ and a classifier $C(.;\theta_c)$, allowing us to effectively handle cross-domain fault diagnosis tasks. Moreover, a discriminator $D(.;\theta_d)$ based on adversarial learning aids in training the network to treat all source domains equally, thereby fostering the learning of domain-invariant representations critical for effective fault diagnosis across diverse operational conditions. Furthermore, the domain-based discrepancy metric $D_m(.;\theta_{d_m})$ plays a vital role in minimizing distribution discrepancies between multiple domains, thereby enhancing the system's capability to generalize diagnostic knowledge effectively. Finally, the developed DGNM is optimized to extract domain-general features through explorations of data in both the original and augmented data.

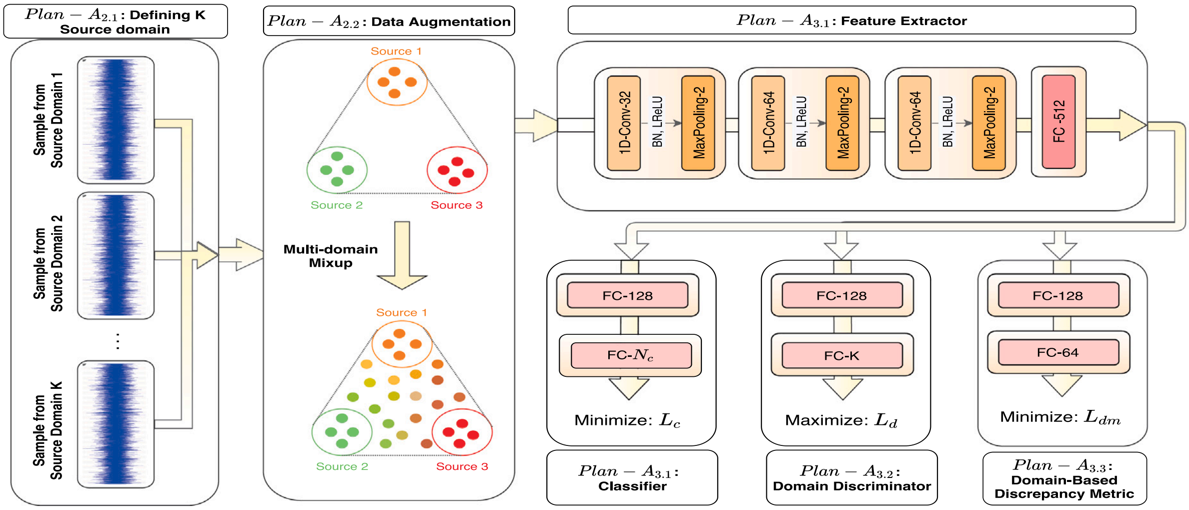
$Plan - A_{2.1}$: Defining K Source domain
Sample from Source Domain 1
Sample from Source Domain 2
⋮
Sample from Source Domain K
$Plan - A_{2.2}$: Data Augmentation
Source 1
Source 2
Source 3
Multi-domain Mixup
Source 1
Source 2
Source 3
$Plan - A_{3.1}$: Feature Extractor
1D-Conv-32
BN, LReLU
MaxPooling-2
1D-Conv-64
BN, LReLU
MaxPooling-2
1D-Conv-64
BN, LReLU
MaxPooling-2
FC -512
FC-128
FC-N_c
Minimize: L_c
FC-128
FC-K
Maximize: L_d
FC-128
FC-64
Minimize: L_{dm}
$Plan - A_{3.1}$: Classifier
$Plan - A_{3.2}$: Domain Discriminator
$Plan - A_{3.3}$: Domain-Based Discrepancy Metric

Fig. 5. Overall architecture of Plan-A_2&A_3.

When the model training is finished, the DGNM ensures that the learned features are robust, thereby enhancing the system's fault identification performance under unseen working conditions of the target machine.

The execution of this activity is segmented into the following fundamental steps.

- $\mathcal{P}_{\mathrm{lan}\,A_{3.1}}$ - *Feature Extractor and Fault Classifier Module*
  In line with the broader framework of the $\mathcal{P}_{\mathrm{lan}\,A_3}$, the Feature Extractor and Fault Classifier Module serve as integral components in DGNM. Leveraging the widespread effectiveness of CNN-based models, our approach utilizes a 1-D CNN for both the source label classifier and feature extractor. This choice is driven by the demonstrated efficacy of 1-D CNNs in handling signals commonly encountered in industrial machinery, particularly vibration signals [36].
  The feature extraction process involves a sequence of three convolutional layers, followed by a flatten layer and a fully-connected layer. This sequence generates high-level representations denoted as $X_g = G(X)$. We employ LeakyReLU as the non-linear activation layer and incorporate batch normalization to expedite the training process. These representations serve as the foundation for the subsequent fault diagnosis classifier module, which aims to create a discriminative classifier relying on domain-invariant feature representations. This classifier module takes X_g as input and comprises two fully-connected layers, effectively reducing the high-dimensional feature representation from the feature extractor to an N_c-dimensional vector, represented as X_c. For original samples from multiple source domains and augmented data by multi-domain mixup algorithm, the classification loss L_c can be written as:

$$L_c = -\frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{n_{s,k}}\sum_{i=1}^{n_{s,k}}\ell\left(C(G(x_i^{s,k})), y_i^{s,k}\right)\right) + \frac{1}{\widetilde{n}_s}\sum_{i=1}^{\widetilde{n}_s}\ell\left(C(G(\widetilde{x}_i^s)), \widetilde{y}_i^s\right) \tag{7}$$

  where $\widetilde{n}_s$ is the number of augmented samples.
- $\mathcal{P}_{\mathrm{lan}\,A_{3.2}}$ - *Domain Discriminator Module*
  In the context of the proposed framework, the Domain Discriminator Module operates on the principles of a generative adversarial network, consisting of a generator and a discriminator, often depicted as a two-player game [37]. To foster the network's capacity to treat all data sources impartially, we have designed a multi-domain discriminator using adversarial learning. This approach facilitates the acquisition of domain-invariant feature representations from diverse source domains, ultimately enhancing the fault classifier's generalization performance across different domains.
  The domain discriminator module employs a structure similar to the classifier module to produce the probability associated with the feature representation X_g. In line with the domain adversarial training scheme, we employ the cross-entropy function as the domain classification loss, defined as:

$$L_d = -\frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{n_{s,k}}\sum_{i=1}^{n_{s,k}}\ell\left(D(G(x_i^{s,k})), d_i^{s,k}\right)\right) + \frac{1}{\widetilde{n}_s}\sum_{i=1}^{\widetilde{n}_s}\ell\left(D(G(\widetilde{x}_i^s)), \widetilde{d}_i^s\right) \tag{8}$$

- $\mathcal{P}_{\mathrm{lan}\,A_{3.3}}$ - *Domain-Based Discrepancy Metric Module*
  In the context of our proposed DG network model, we observe that supervised learning and adversarial training can project data samples with the same machine condition labels from various domains into the same regions within a learned sub-space. However, there is a potential risk of overfitting, and the clustering effect of different classes cannot be guaranteed. This vulnerability may result in reduced generalization performance when applied to new domains. To mitigate this challenge, we introduce a domain-based discrepancy metric module within the DGNM framework. The primary goal of this module is to minimize the distance between data distributions from different source domains, fostering cohesion among them. In essence, this aligns data from diverse domains in a common distribution space while maintaining reasonable distances between data from the same domain. To achieve this, the module maximizes intra-domain distance and simultaneously minimizes inter-domain distance through regularizations applied to its output, connecting two fully connected layers to the feature extractor's output. The intra-class distance, denoted as L_{intra}, measures clustering compactness, ensuring that data within the same class clusters closely. This can be defined as:

$$\begin{aligned} L_{intra} &= \tfrac{1}{N_c N_{dm}} \textstyle\sum_{l=1}^{N_c} \tfrac{1}{n_s^l} \sum_{j=1}^{n_s^l} \parallel D_m(\bar{G}(X^l)) - D_m(G(X_j^l)) \parallel_1, \\ D_m(\bar{G}(X^l)) &= \tfrac{1}{n_s^l} \textstyle\sum_{j=1}^{n_s^l} D_m(G(X_j^l)), \\ X^l &= \textstyle\sum_{k=1}^{K} \sum_{i=1}^{n_{s,k}^l} (x_{i,l}^{s,k}) + \sum_{i=1}^{\widetilde{n}_s^l} (\widetilde{x}_{i,l}^s), \end{aligned} \tag{9}$$

  where X^l represents the set of samples associated with the lth machine health condition in our training dataset, which includes both the original and augmented data. Within X^l, X_j^l signifies the jth individual sample. The variable n_s^l denotes the total number of samples within X^l. Furthermore, N_{dm} refers to the dimension of the output vector produced by D_m, and $D_m(\bar{G}(X^l))$ signifies the mean value derived from $D_m(G(X^l))$. Similarly, the inter-class distance, L_{inter}, quantifies separability, can be defined as:

$$\begin{aligned} &L_{inter} = \tfrac{1}{N_c N_{dm}} \textstyle\sum_{l=1}^{N_c} \parallel D_m(\tilde{G}(X)) - D_m(\bar{G}(X^l)) \parallel_1; \\ &D_m(\tilde{G}(X)) = \tfrac{1}{N_c} \textstyle\sum_{l=1}^{N_c} D_m(\bar{G}(X^l)), \end{aligned} \tag{10}$$

where $D_m(\tilde{G}(X))$ represents the average value of $D_m(\bar{G}(X^l))$ in different classes. To summarize, by integrating the concepts of clustering compactness L_{intra} and separability L_{inter}, we formulate the optimization objective for learning the distance metric as follows:

$$L_{dm} = L_{intra} - L_{inter}. \tag{11}$$

- ***Optimization Objectives:***
  The optimization of the DGNM involves three primary objectives, focusing on the parameters of the feature extractor θ_g, fault diagnosis classifier θ_c, domain discriminator θ_d, and domain-based discrepancy metric θ_{dm}. A suitable optimization objective is threefold: to minimize the losses $L_c(\theta_g, \theta_c)$ and $L_{dm}(\theta_g, \theta_{dm})$, while simultaneously maximizing the loss $L_d(\theta_g, \theta_d)$.
  In so doing, the comprehensive objective function L can be formulated as follows:

  $$L(\theta_g, \theta_c, \theta_d, \theta_{dm}) = L_c(\theta_g, \theta_c) - \beta_1 L_d(\theta_g, \theta_d) + \beta_2 L_{dm}(\theta_g, \theta_{dm}), \tag{12}$$

  where the hyperparameters β_1 and β_2 are used to weight the three components.
  Once the overall loss function L is established, the model is optimized so that:

  $$\begin{aligned} &(\hat{\theta}_g, \hat{\theta}_c, \hat{\theta}_{dm}) = \underset{\theta_g, \theta_c, \theta_{dm}}{\operatorname{argmin}} L(\theta_g, \theta_c, \hat{\theta}_d, \theta_{dm}), \\ &(\hat{\theta}_d) = \underset{\theta_d}{\operatorname{argmax}} L(\hat{\theta}_g, \hat{\theta}_c, \theta_d, \hat{\theta}_{dm}). \end{aligned} \tag{13}$$

  The network parameters are iteratively updated using the stochastic gradient descent (SGD) algorithm within each training batch. SGD is a fundamental optimization technique in machine learning that computes gradients of the loss function with respect to the network parameters using a subset of the training data (batch), facilitating efficient parameter updates. This approach enables the network to continually refine its parameters towards minimizing the loss function, thereby enhancing its capability to generalize across different domains in fault diagnosis. By iteratively adjusting parameters based on the gradients computed from each batch, SGD ensures that the network progressively learns representations that are robust and adaptable to diverse operational conditions encountered in real-world scenarios.

4.4. *Executor*

The last activity in the Planner (i.e., $\mathcal{P}_{\text{lan}A_3}$) generates a trained model which is stored in *Knowledge* and must be executed by the FDS unit in the managed element. Hence, the *Executor* is in charge of loading and delivering the developed diagnostic model to the FDS.

4.5. *Knowledge*

Knowledge plays an important role in providing mechanisms for all MAPE elements to work together and collaboratively to realize self-adaptation. The *Knowledge* includes different information as:

- Run-time data from the target machine collected via the *Monitor*, including the relevant values for operational data
- "WCC" and "$\mathcal{D}_{\text{iscrepancy}}$" indices evaluated by the *Analyzer*
- Data from the DT, including the virtual model of the machinery and its related geometrical parameters, and fault size
- Activity models ($\mathcal{P}_{\text{lan}A_1}, \mathcal{P}_{\text{lan}A_2}, \mathcal{P}_{\text{lan}A_3}$), that the *Planner* needs to execute at each scenario
- Design parameters (e.g., threshold values $T, \theta_p, \beta_1, \beta_2$)
- Historical data, including previous work conditions of the target machinery, as well as data derived from similar machines across diverse industrial plants, companies, and digital platforms.

5. The rotating machine case-study

In this study, the proposed self-adaptive fault diagnosis system is implemented as an experimental case study on an industrial rotating machine. It is important to note that rotating machinery is extensively used in various practical industrial applications, such as railways, mining vehicles, electric cars, civil aircraft, wind turbines, and helicopters. The digital twin model of rotating machinery is detailed in the appendix (see Appendix A). In the following, we present the customization of the MAPE-K loop framework specifically for rotating machinery, including comprehensive details on the adaptation process.

In the context of a rotating machine, the *Monitor* functional component considers data such as the speed and load of the rotating machine as operational data. These data enable the monitoring of variations in the working conditions of the rotating machine, allowing for the detection of any changes. Additionally, the vertical acceleration of the bearing outer race is chosen as a response signal, serving as the basis for evaluating the similarity between simulated and real systems.

Within the *Analyzer* functional component, two indices are computed. The first index, defined in Eq. (2), utilizes machine speed and load data to monitor variations in operational conditions. The second index (see Eq. (3)) is formulated by running the DT model of the rolling element bearing with new operational data, assessing the similarity between real and simulated responses (bearing outer race vibration data). However, to overcome the limitations of comparing vibration data in the time domain for rotating machines, metrics based on spectral divergence and geodesic distance, as suggested by [38], are employed to evaluate similarity. The first metric, spectral divergence, akin to the Kullback–Leibler divergence, is computed as follows:

$$\log(\rho(S_{DT}, S_R)) = \log\left(\int_{-T/2}^{T/2} \frac{S_{DT}}{S_R} dt\right) - \int_{-T/2}^{T/2} \log\left(\frac{S_{DT}}{S_R}\right) dt, \tag{14}$$

where S_{DT} and S_R are the power spectral density (PSD) of the vertical acceleration of the bearing's outer race signal obtained from the DT and the real system, respectively. Second, the geodesic distance d_g between two spectra which is equivalent to the conventional idea of standard deviation can be defined as well:

$$d_g(S_{DT}, S_R) = \sqrt{\int_{-\pi}^{\pi} \log\left(\frac{S_{DT}(\theta)}{S_R(\theta)}\right)^2 \frac{d\theta}{2\pi} - \left(\int_{-\pi}^{\pi} \log\left(\frac{S_{DT}(\theta)}{S_R(\theta)}\right) \frac{d\theta}{2\pi}\right)^2}. \tag{15}$$

The response comparison outcome between the real machine and the DT is expressed as:

$$E(p) = \{p_{DT} - p_R\} = \left\{ \begin{array}{l} \log(\rho(S_{DT}, S_R)) \\ d_g(S_{DT}, S_R) \end{array} \right\} \tag{16}$$

where E represents the calculated error between the response of the DT (p_{DT}) and the response of the real machine (p_R) and p is the response signals.

In the *Planner* functional component, when a significant discrepancy (i.e., $\mathcal{D}_{\text{iscrepancy}} = L$) is detected, the load–deflection factor K_b and the radial clearance of the bearing c_r are considered critical parameters for the updating strategy in $\mathcal{P}_{\text{lan}A_1}$. To minimize the discrepancy, an optimization function through the defined critical parameters and introduced metrics is defined as follows:

$$\min \parallel E(p) \parallel_2^2, \quad E(\{K_b, c_r\}) = f(S_{DT}, S_R) = \left\{ \begin{array}{l} \log(\rho(S_{DT}, S_R)) \\ d_g(S_{DT}, S_R) \end{array} \right\}. \tag{17}$$

Table 2
Details information of experimental data.

Dataset	Bearing type	Code	Work operation	Health conditions	Sampling frequency	No. of samples
Paderborn	6203	A_1 A_2 A_3	(0.7 N m, 1500 rpm) (0.7 N m, 900 rpm) (0.1 N m, 1500 rpm)	(H, IF, OF)	64 kHz	(600,3*200,2*300)
CWRU	6205	B_1 B_2 B_3	(0 Hp, 1797 rpm) (1 Hp, 1772 rpm) (2 Hp, 1750 rpm)	(H, IF, OF)	12 kHz	(600,3*200,3*200)
MFPT	NICE	C	(200 lbs, 1500 rpm)	(H, IF, OF)	48828/97656 Hz	(600,600,600)
DT	*target**	D	*target**	(H, IF, OF)	12 kHz	(600,3*200,3*200)

6. Experimental validation and performance analysis

In order to confirm the effectiveness of the proposed methodology, the proposed approach is tested through an experimental analysis of the case study 5. The validation analysis considers comprehensive experiments leveraging public datasets (see Section 6.1) as well as data provided by the DT (see Appendix A).

A comparison w.r.t. alternative methods in the technical literature on this allows a more in-depth analysis of the experimental results and of the overall performance assessment.

6.1. Dataset

In this paper, we utilize four distinct datasets to assess the performance of our proposed approach. These datasets encompass three publicly available datasets and one simulation dataset. We focus on three specific health conditions, namely, the normal condition (NC), inner race fault (IF), and outer race fault (OF), as these represent the most prevalent bearing faults. For a comprehensive overview of these datasets, please refer to Table 2.

6.1.1. Paderborn

The first dataset used in this study was created at the KAT data center in Paderborn University and was recorded at a sampling rate of 64 kHz, as documented in [39]. The experiments involved the use of a 6203 ball bearing with real damage, which was obtained from an accelerated lifetime test. Vibration data was collected under three distinct operational conditions, where the loading torque ranged from 0.1 to 0.7 N m, and the radial force was consistently set at 1000 N. Additionally, the rotational speed varied from 400 to 1500 RPM. To ensure uniformity, all vibration signals were resampled to a common frequency of 12 kHz for inclusion in the dataset.

6.1.2. CWRU

The CWRU bearing dataset [40] serves as a widely recognized benchmark dataset for rolling element fault diagnosis. The dataset focuses on bearing tests conducted on the drive end bearing SKF-6205. Vibration signals were recorded for a motor with loads ranging from 0 to 3 horsepower and speeds from 1797 to 1730 RPM, sampled at 12 kHz. Notably, each fault type includes three fault sizes (0.18, 0.36, and 0.53 mm), and various fault severities are grouped into one category.

Table 3
The geometrical parameters of the DT model for three types of bearings.

Parameters (Units)	SKF-6203	SKF-6205	NICE
Mass of shaft (M_S) (kg)	50	5	40/45/50/55
Stiffness of shaft (K_S) (N/m)	$7.42 * 10^7$	$7.42 * 10^7$	$7.42 * 10^7$
Damping of shaft (R_S) (Ns/m)	1376.8	1376.8	1376.8
Mass of pedestal (M_P) (kg)	5	60	5
Stiffness of pedestal (K_P) (N/m)	$1.51 * 10^7$	$1.51 * 10^7$	$1.51 * 10^7$
Damping of pedestal (R_P) (Ns/m)	2210.7	2210.7	2210.7
Mass of Sprung (M_R) (kg)	1	1	1
Stiffness of Sprung (K_R) (N/m)	$8.8826 * 10^9$	$8.8826 * 10^9$	$8.8826 * 10^9$
Damping of Sprung (R_R) (Ns/m)	9424.8	9424.8	9424.8
Load–deflection factor (K_b) (N/m$^{1.5}$)	$8.753 * 10^9$	$9.545 * 10^9$	$9 * 10^9$
Acceleration of gravity (g) (m/s^2)	9.8	9.8	9.8
Ball diameter (D_b) (mm)	6.746	7.94	5.97
Pitch diameter (D_p) (mm)	28.5	39.040	31.62
Fault depth (C_d) (mm)	1	1	1
Fault width (L) (mm)	0.3/0.5/0.7	0.3/0.5/0.7	0.3/0.5/0.7
Number of balls (n_b)	8	9	8
Radial clearance (c_r) (m)	$2 * 10^{-6}$	$2 * 10^{-6}$	$2 * 10^{-6}$

6.1.3. MFPT

The MFPT dataset [41] collects accelerations of vibration signals data coming from NICE bearing. Under normal conditions, the data is collected at a load of 270 lbs, and under each fault state, data is gathered under 200 lbs loads. In the MFPT bearing experiment, the actual sampling frequency is 97 656 Hz for normal conditions and 48 828 Hz for fault situations. All vibration signals are essentially re-sampled to 12 kHz to ensure that all samples in the dataset have the same sampling frequency.

6.1.4. DT

Considering bearing and its work operation, simulated data from a purposely designed target-bearing DT model is exploited to support the diagnosis. To this aim, following the successful adjustment of the DT's performance to fall within an acceptable range, the updated DT is then leveraged to conduct simulations in different health conditions, encompassing three different fault sizes for each fault type (in $\mathcal{P}_{\mathrm{lan}\,A_{2.1}}$). Also, as the real machine in the target domain, the sampling frequency is set to 12 kHz. The geometrical parameters for the DT model of the different types of target bearings are detailed in Table 3. More details of the bearing DT model and its experimental analysis can be found in our previous study [42].

6.2. Alternative methodologies for comparison

In order to comprehensively assess the effectiveness and superiority of our proposed method, we employ a range of well-established

Table 4
Detailed information of diagnostic tasks.

	Task	Source domain	Target domain
Cross-work operation	Case 1	$A_1 + A_2 + D$	A_3
	Case 2	$A_1 + A_3 + D$	A_2
	Case 3	$A_2 + A_3 + D$	A_1
	Case 4	$B_1 + B_2 + D$	B_3
	Case 5	$B_1 + B_3 + D$	B_2
	Case 6	$B_2 + B_3 + D$	B_1
Cross-Machine	Case 7	$A + B + D$	C
	Case 8	$A + C + D$	B_1
	Case 9	$B + C + D$	A_1

state-of-the-art methods for comparison. These methods are outlined below:

- **Convolutions Neural Network (CNN):** This method uses a basic CNN trained on data from multiple sources. The method focuses on diagnosing the target data directly, without incorporating any knowledge transfer or broader generalization techniques.
- **Domain-Adversarial Network (DAN)** [43]: This method leverages adversarial training across various source domains. Its primary goal is to develop feature representations that are invariant to domain differences, making them robust across multiple source domains and applicable to the target task.
- **Multi-Kernel Maximum Mean Discrepancy (MK-MMD)** [11]: In this method, different kernel versions of maximum mean discrepancy are applied. The aim is to minimize the distribution differences between each pair of source domains within the deep latent spaces.
- **Adversarial Domain-Invariant Generalization (ADIG)** [20]: ADIG represents a versatile framework for domain regression. It employs instance normalization and spectral normalization to extract characteristics that are both relevant to faults and invariant across domains.
- **Deep Mixed Domain Generalization Network (DMDGN)** [17]: This novel approach includes a network that utilizes traditional data augmentation at the feature level, combined with adversarial learning. The network introduces perturbations through adversarial training and utilizes a domain-based discrepancy metric to balance distances within and between domains effectively.
- **Proposed Approach without Augmentation (Proposed-NoAug):** To demonstrate the value of the data augmentation module in our proposed method ($\mathcal{P}_{\mathrm{lan}A_{2.2}}$), we also test a variant where the multi-domain mixup is excluded.

6.3. Implementation details

To assess the proposed method, we conduct experiments under various cross-work operating conditions and cross-machine diagnosis tasks, utilizing the datasets mentioned earlier. We perform comparisons with state-of-the-art methods to assess the efficacy of our approach. In each task, a specific working condition or machine is chosen as the unseen target domain to assess the model's performance, while data from the remaining conditions or machines, along with simulation data, are utilized for model training. For instance, $A_1 + A_2 + D \rightarrow A_3$ denotes that A_1, A_2 and D serve as source domains for training, and A_3 used as the target domain for testing. It is worth highlighting that, given the larger domain shifts between machines compared to shifts between working conditions within the same machine, we combine data from all working conditions of a machine into a single domain in cross-machine tasks. Table 4 provides an overview of the different diagnostic cases we have designed based on these datasets.

The success of the DGNM model we introduced is significantly influenced by its hyperparameters, specifically β_1 and β_2. After thorough experimental evaluation, these were set at $\beta_1 = 1$ and $\beta_2 = 1$. Details about the methodical tuning of these parameters can be found in the subsection dedicated to parameter sensitivity analysis (see; Section 6.4.3). Additionally, the α parameter for the Dirichlet distribution was configured as $(0.5, 0.5, 0.5)$.

The optimization procedure is carried out using the training set, and we select the model with the mean accuracy on the validation set for the current task. This process is repeated ten times, with 80% of the data randomly allocated to the training set and the remaining 20% to the validation set for each iteration. To strike a balance between optimization efficiency and computational resources, we opt for a batch size of 64 in our study. Furthermore, we set a maximum epoch limit of 100 for training. In the optimization phase, we employ SGD with a momentum of 0.9 and weight decay of $1 * 10^{-3}$ for parameter updates. Additionally, a dropout mechanism with a probability 0.5 is utilized.

To ensure a fair comparison with other methods, we maintain a consistent network structure and experimental settings across all compared methods. The hyperparameters for these comparison methods are determined using a similar tuning process as that employed for our proposed method.

6.4. Diagnosis results and analysis

6.4.1. Comparison with latest state-of-the-art models

The performance evaluation of our DGNM model, along with several state-of-the-art methods, was conducted across various diagnostic tasks, including Cross-work operation and Cross-Machine challenges. The outcomes, as shown in Figs. 6 and 7, lead to several insightful findings:

- The CNN approach showed lower average accuracies of 83.06% and 53.05% in the two primary tasks. This underperformance is likely due to the adverse effects of training on data with varying distributions, which hinders the model's ability to generalize. This indicates the impracticality of directly applying source-domain knowledge to the target domain without proper generalization techniques. The significant drop in performance underscores the need for more advance methods that can handle domain shifts effectively.
- Compared to CNN, MK-MMD and DAN demonstrate an improved average accuracy by 4.56% and 5.73% (or 9.33% and 10.07%) on the Cross-Work Operation (Cross-Machine) task, respectively. The statistical moment matching approach in MK-MMD and the adversarial training method in DAN both aim to reduce distribution discrepancies across source domains and help acquire domain-invariant representations.
- ADIG and DMDGN, two state-of-the-art methods based on adversarial training, further improve diagnostic performance. ADIG excels in learning invariant fault representations through domain regression, surpassing DAN and MK-MMD methods. However, DMDGN outperforms ADIG by $1.72\%(7.33\%)$, owing to its data augmentation techniques that enhance model generalization.
- Incorporating multi-domain mixup in our proposed method resulted in a significant accuracy increase of $3.18\%(5.77\%)$. This improvement underscores the role of augmentation in enhancing the fault diagnosis method's generalization against variations in cross-work operation and cross-machine tasks. By effectively combining data from multiple domains, our method can better capture the underlying patterns and improve diagnostic accuracy across different scenarios.
- All methods experienced a performance drop in cross-machine tasks compared to cross-work operations, attributable to the greater distribution discrepancies that need to be addressed in cross-machine diagnostics. This highlights the inherent difficulty in transferring knowledge across different machines, where variations in operational conditions and machine characteristics pose significant challenges.

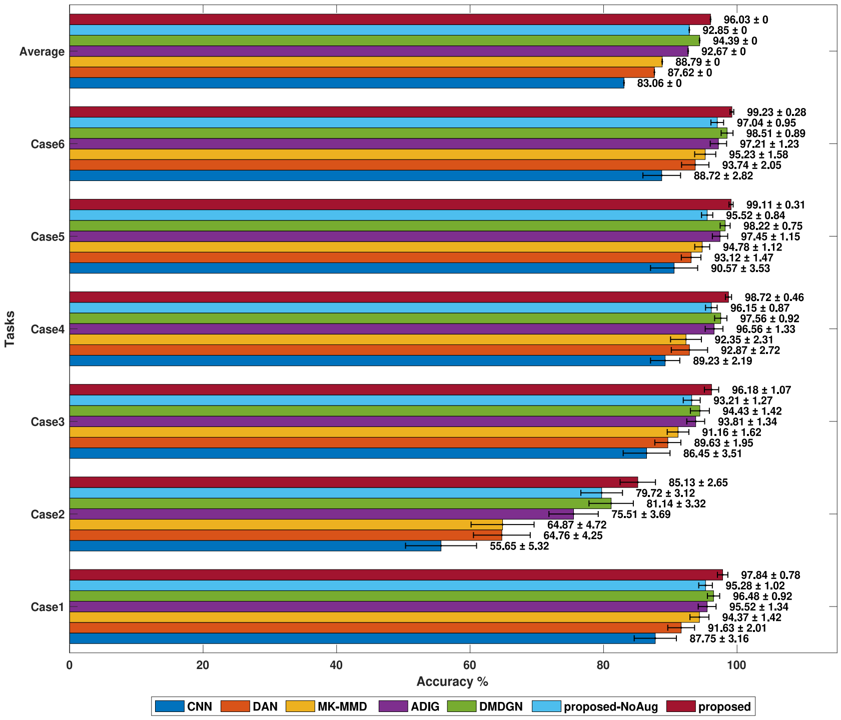

Fig. 6. Diagnosis performance comparisons in cross-work operation tasks with DT samples.

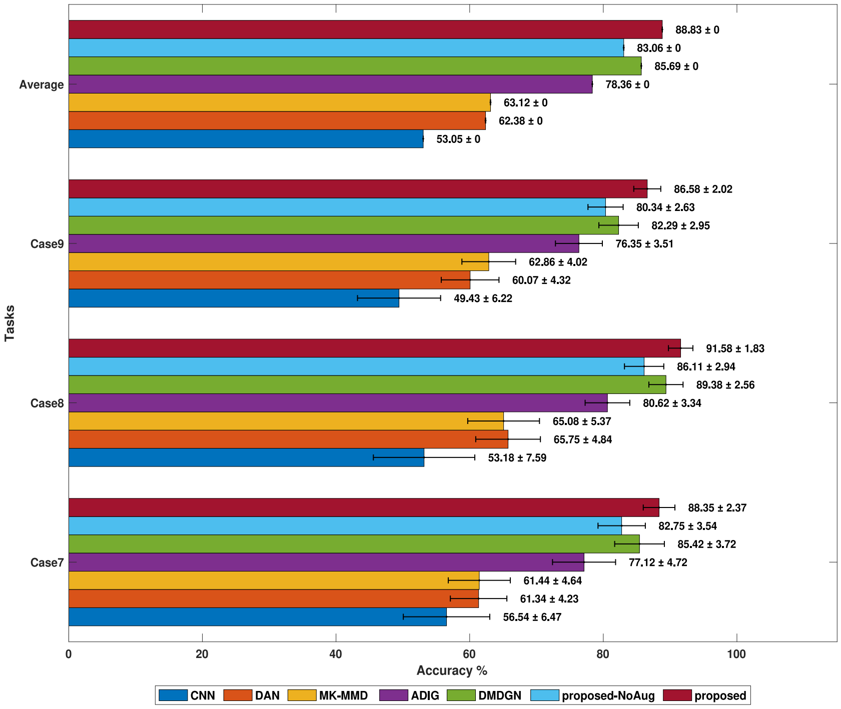

Fig. 7. Diagnosis performance comparisons in cross-machine tasks with DT samples.

- As shown in Figs. 6 and 7, our proposed method consistently outperforms all other approaches across nearly all DG tasks, achieving the highest average accuracy rates of 96.03% in cross-work operation tasks and 88.83% in cross-machine tasks. Additionally, our method demonstrates greater stability compared to alternative methods in most cases. These findings highlight the exceptional knowledge generalization capabilities of our approach, making it highly robust for diverse diagnostic situations and effectively addressing unseen working conditions in real-time diagnosis scenarios. The consistent performance across different tasks and domains emphasizes the potential of our method to provide reliable fault diagnosis in practical applications, ensuring machinery operates efficiently and safely under various conditions.

6.4.2. Diagnosis result without DT samples

Figs. 8 and 9 illustrate the performance of our proposed method compared to other methods for Cross-work operation and cross-machine

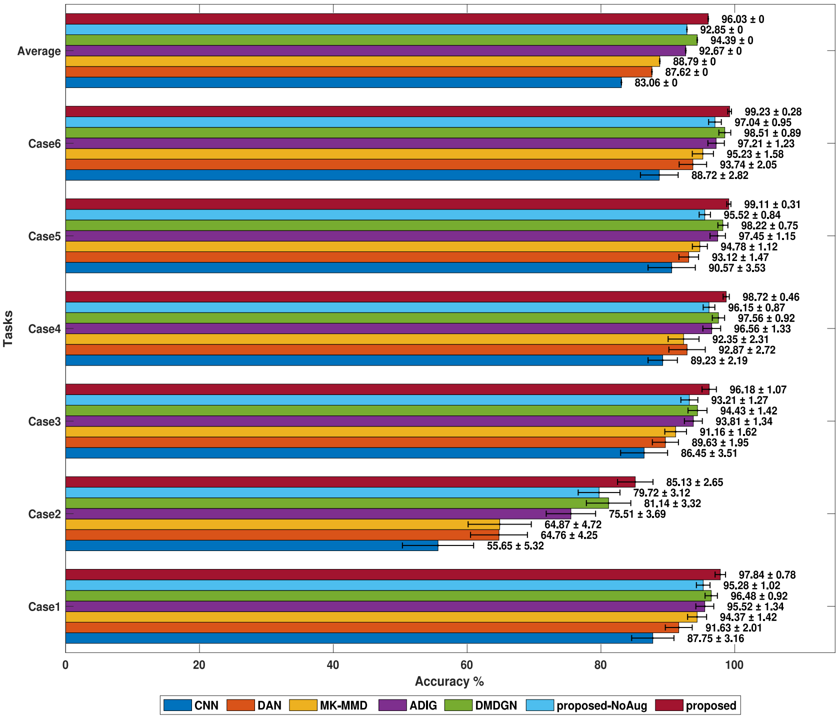

Fig. 8. Diagnosis performance comparisons in cross-work operation tasks without DT samples.

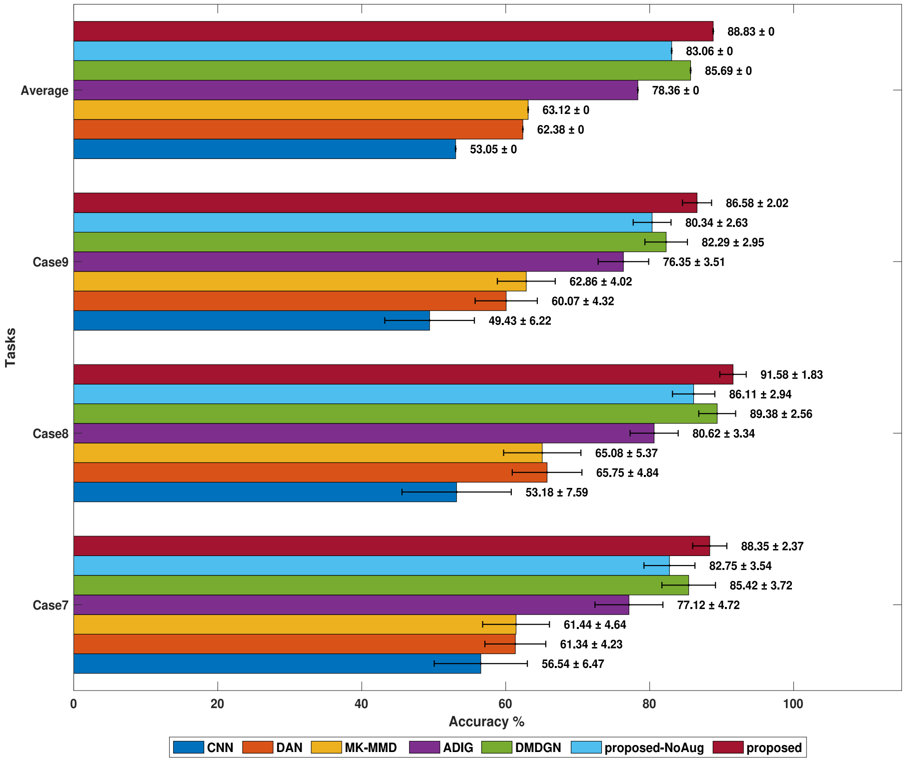

Fig. 9. Diagnosis performance comparisons in cross-machine tasks without DT samples.

tasks, specifically in scenarios where the DT dataset is not utilized. The findings reveal that our method consistently surpasses competing models, showcasing its superior ability to generalize across different tasks. However, it is crucial to recognize that the diagnostic performance improves significantly when the DT dataset is included, as evident from the comparison of results in Figs. 6 and 7, and Figs. 8 and 9. This improvement is attributed to the DT dataset's role in minimizing the variance in data distribution between source and target domains. As a result, domain pairs with reduced discrepancy may lead to more accurate diagnostic knowledge. For instance, in the cross-machine case,

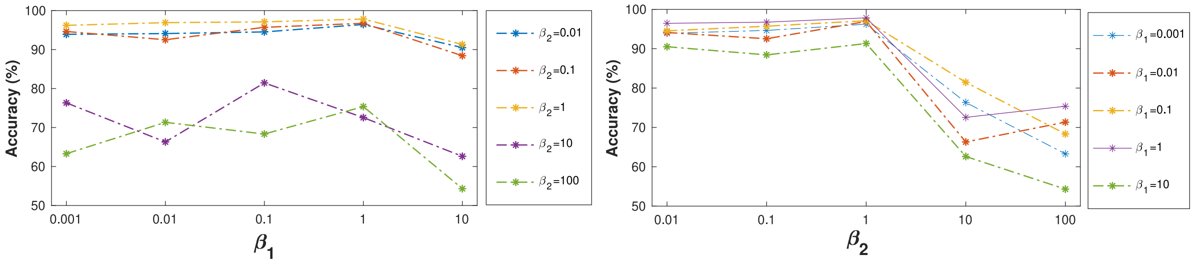

Fig. 10. Diagnosis results on case 1 for different β_1 and β_2.

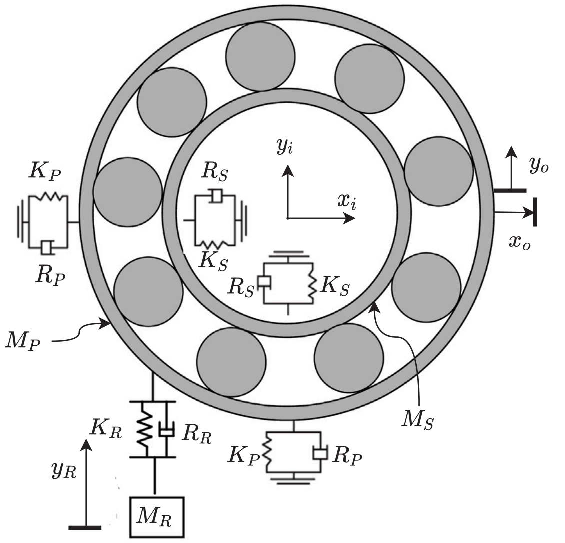

Fig. 11. Schematic of 5-DOF rolling element bearing dynamic model. $M_P, K_P and R_P$ are the pedestal mass, stiffness, and damping, respectively; $M_S, K_S and R_S$ illustrate rotor shaft mass, stiffness, and damping, respectively; M_R, $K_R and R_R$, are sprung system mass, stiffness, and damping, respectively; K_b is the load–deflection factor of balls. (see Table 3 for further nomenclature, measure units and parameters nominal values.).

the average accuracy of the proposed model is increased from 81.37% to 88.83% when considering the DT dataset.

6.4.3. Parameter sensitivity analysis

The effectiveness of our model is significantly influenced by the selected values of the trade-off coefficients for different loss functions in the overall objective. Specifically, coefficients β_1 and β_2 are pivotal in balancing the interplay between the generalization and classification aspects, which is crucial for maintaining accuracy in complex engineering contexts. To optimize these coefficients, we employed a grid search strategy. This involved varying $\beta_1$across a range of $\{0.001, 0.01, 0.1, 1, 10\}$ and β_2 across $\{0.01, 0.1, 1, 10, 100\}$. We then assessed combinations of these values to determine the most effective pair.

In our examination of all 25 possible pairings, we used the accuracy index as the primary metric for determining the optimal combination. For instance, in case 1, we analyzed the diagnostic accuracies under various settings, as depicted in Fig. 10. The analysis revealed in Fig. 10(a) demonstrates that accuracy is highly sensitive to changes in β_2, while β_1 appears to have a more consistent and minor impact. This suggests that the domain-based discrepancy aspect, associated with β_2, is more critical to our model's performance compared to the element linked with β_1. The highest accuracy was achieved when both β_1 and β_2 were set to 1, leading us to select $(\beta_1, \beta_2) = (1, 1)$ as the optimal coefficient combination for this experiment.

Table 5
Average of Training time (s) and testing time (s) of different methods.

Task	CNN	DAN	MK-MMD	ADIG	DMDGN	proposed
Training	129.21	187.32	224.82	310.51	354.72	373.45
Testing	0.23	0.23	0.24	0.23	0.24	0.24

6.4.4. Computational efficiency analysis

Efficient computational performance is essential for real-time cross-domain fault diagnosis tasks, as it directly impacts the promptness of decision-making and control. This section focuses on analyzing the computational efficiency of various models in the context of domain generalization. All models were developed using the PyTorch framework and evaluated on a computer equipped with an Intel UHD Graphics 630 GPU, 32 GB RAM, and an Intel Core i9 processor. Table 5 presents a comparative analysis of the average training and testing durations for each method. It is important to note that the testing time refers to the total time needed to predict outcomes for all test samples.

Our findings indicate that the CNN model is the most time-efficient, primarily due to its singular focus on supervised classification loss, which simplifies its optimization objective. In contrast, our proposed method exhibits a longer training duration, approximately double that of the DAN method. This increased time requirement is largely attributed to the computational demands of processing augmented samples. However, it is noteworthy that the testing times for both the proposed and other comparative methods are relatively similar, thanks to comparable network configurations. In practice, the testing phase for all samples is completed in less than a quarter of a second. This observation underscores the practicality of our proposed method for online, real-time diagnostic applications.

7. Conclusion

In this paper, we have addressed some major challenges in fault diagnosis for mechanical equipment posed by dynamic working conditions due to limitations in existing domain adaptation and generalization approaches. To tackle those challenges, we introduced a novel self-adaptive fault diagnosis system, combining MAPE-K, domain generalization network model, and digital twin techniques. We have shown how the proposed system offers several key advantages. First, by incorporating the MAPE-K loop, our system achieves real-time adaptation to new, unseen working conditions, reducing the reliance on direct human intervention and enhancing system autonomy. The MAPE-K loop ensures that the diagnostic model continuously evolves to ensure reliable decisions, taking into account changes in operational and environmental conditions. Second, the utilization of digital twins as a supplementary data source addresses the practical challenges associated with obtaining labeled training datasets from actual machines. The continuous tuning of DT parameters based on the new unseen working condition of the target machine ensures that the generated data aligns with the geometrical characteristics and new working conditions of the real system. Third, integrating the DGNM addresses the drawbacks of existing domain generalization methods by preserving domain-private

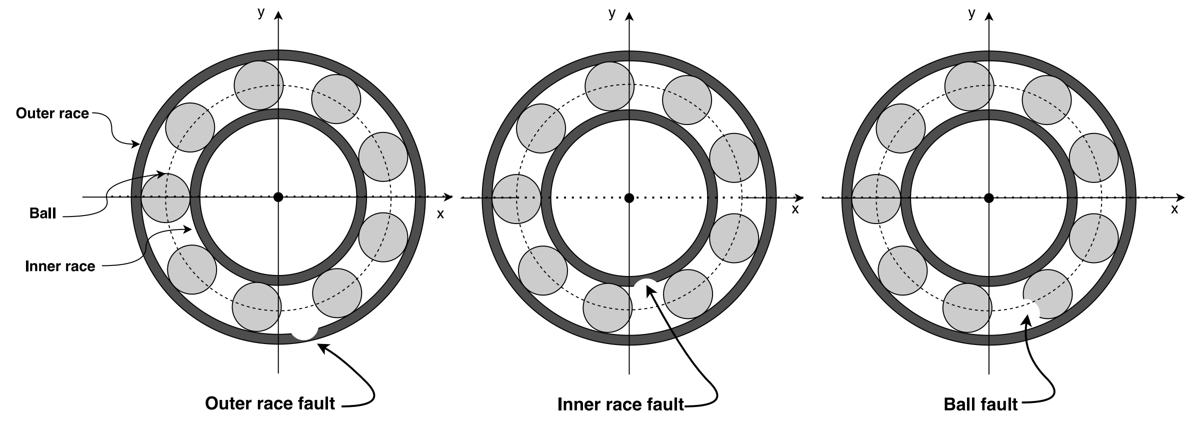

Fig. 12. Simplified model of spalling defect on outer race, inner race, and ball of the bearing.

features and improving feature diversity through multi-domain data augmentation, adversarial learning, and a domain-based discrepancy metric. We have evaluated the system on three rotating machinery datasets, showcasing superior performance in cross-work operation and cross-machine tasks compared to state-of-the-art methods. Therefore, we have demonstrated that the system is effective in handling the complexities of dynamic industrial machinery environments, paving the way for future research in autonomous fault diagnosis systems within next-generation cyber–physical systems. For further work, we will concentrate on exploring methods to enhance diagnostic models, particularly in scenarios where a category shift exists between the source domains and the target domain.

CRediT authorship contribution statement

Mehdi Saman Azari: Writing – original draft, Validation, Software, Methodology, Formal analysis, Conceptualization. **Stefania Santini:** Writing – review & editing, Writing – original draft, Supervision. **Farid Edrisi:** Resources, Investigation. **Francesco Flammini:** Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We utilized publicly available datasets, which are already cited in the paper.

Acknowledgment

All authors approved the version of the manuscript to be published.

Appendix A. DT of rotating machines

Herein, following the procedure in Section 3.1, the DT for allowing diagnosis in rotation machinery is derived.

- **1st Step: Select the critical components.** In the case of the rotating machinery, the most crucial component is the rolling element bearing (REB) [44], and, commonly, in order to prevent damage from spreading to the rest of the rotating machine, it is necessary to diagnose its potential faults within a reasonable time.
- **2nd Step: Derive the dynamic model.** Consider the following advanced dynamic bearing modeling that accounts for the impacts of inner and outer race inertia, and contact force of the bearing. In so doing, the overall dynamic bearing model has 5 degree-of freedom (DOF), where the outer and inner races have 2 DOF each, while an additional DOF is associated with sprung mass reflecting high-frequency structural vibrations. Now, consider the schematic of a rolling element bearing with 5 DOF as in Fig. 11. Namely, the inner race (shaft) dynamics, that spins at a constant speed while being firmly coupled to the rotor shaft, can be represented as the following 2-DOF model:

$$\begin{aligned} &M_S\ddot{x}_i + R_S\dot{x}_i + K_S x_i + f_x = 0, \\ &M_S\ddot{y}_i + R_S\dot{y}_i + K_S y_i + f_y = M_S g, \end{aligned} \tag{18}$$

where g is the gravity acceleration, x_i, y_i are the inner race displacements of mass in x, y direction, respectively, f_x and f_y are the contact forces, and $M_S, K_S and R_S$ illustrate rotor shaft mass, stiffness, and damping, respectively.
Similarly, the outer race (pedestal) dynamics, which has translational motion in (x, y) directions and does not rotate, can be modeled as the following 2-DOF system:

$$\begin{aligned} &M_P\ddot{x}_o + R_P\dot{x}_o + K_P x_o - f_x = 0, \\ &M_P\ddot{y}_o + (R_P + R_R)\dot{y}_o + (K_P + K_R)y_o - R_R\dot{y}_R - K_R y_R - f_y = M_P g \end{aligned} \tag{19}$$

where x_o and y_o are outer race displacements of mass in the x, y direction, respectively, $M_P, K_P and R_P$ are the pedestal mass, stiffness, and damping, respectively, $M_R, K_R and R_R$, are sprung system mass, stiffness, and damping, respectively, and y_R is displacement of sprung mass.
Finally, the sprung mass equation, which is linked in the y direction can be expressed as:

$$M_R\ddot{y}_R + R_R(\dot{y}_R - \dot{y}_o) + K_R(y_R - y_o) = M_R g + \zeta \tag{20}$$

As distribution on the sprung mass, an additional noise source (ζ) is introduced. More details of the bearing dynamic model can be found in our previous study [42].
- **3rd Step: Define and model the faults.** During operation, the bearing can be exposed to surface damage caused by localized fatigue. The main fatigue failure type of bearings is spalling, see Fig. 12, which could happen in different parts of the bearing, including the outer race, the inner race, and the ball part. More details of modeling localized bearing faults can be found in our previous study [42].
- **4th Step: Specify the virtual sensors.** In the rotating machine, the physical model's output is the vertical acceleration of the bearing's outer race (pedestal), which can be interpreted as a

virtual vibration sensor measuring the y-directional velocity of the outer race ($\dot{y}_o$).

References

[1] Xu Z, Saleh JH. Machine learning for reliability engineering and safety applications: Review of current status and future opportunities. Reliab Eng Syst Saf 2021;211:107530.

[2] Azari MS, Flammini F, Santini S. Improving resilience in cyber-physical systems based on transfer learning. In: 2022 IEEE international conference on cyber security and resilience. IEEE; 2022, p. 203–8.

[3] Azari MS, Flammini F, Santini S, Caporuscio M. A systematic literature review on transfer learning for predictive maintenance in industry 4.0. IEEE Access 2023.

[4] Li X, Zhang W, Ma H, Luo Z, Li X. Domain generalization in rotating machinery fault diagnostics using deep neural networks. Neurocomputing 2020;403:409–20.

[5] Zhao C, Zio E, Shen W. Domain generalization for cross-domain fault diagnosis: An application-oriented perspective and a benchmark study. Reliab Eng Syst Saf 2024;109964.

[6] Chattopadhyay P, Balaji Y, Hoffman J. Learning to balance specificity and invariance for in and out of domain generalization. In: Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part IX 16. Springer; 2020, p. 301–18.

[7] Rajabi S, Azari MS, Santini S, Flammini F. Fault diagnosis in industrial rotating equipment based on permutation entropy, signal processing and multi-output neuro-fuzzy classifier. Expert Syst Appl 2022;206:117754.

[8] Moradi R, Cofre-Martel S, Droguett EL, Modarres M, Groth KM. Integration of deep learning and Bayesian networks for condition and operation risk monitoring of complex engineering systems. Reliab Eng Syst Saf 2022;222:108433.

[9] Zhao Z, Li T, An B, Wang S, Ding B, Yan R, et al. Model-driven deep unrolling: Towards interpretable deep learning against noise attacks for intelligent fault diagnosis. ISA Trans 2022;129:644–62.

[10] Jiang L, Lei W, Wang S, Guo S, Li Y. A Deep Convolution Multi-Adversarial adaptation network with Correlation Alignment for fault diagnosis of rotating machinery under different working conditions. Eng Appl Artif Intell 2023;126:107179.

[11] An Z, Li S, Wang J, Xin Y, Xu K. Generalization of deep neural network for bearing fault diagnosis under different working conditions using multiple kernel method. Neurocomputing 2019;352:42–53.

[12] Chen P, Zhao R, He T, Wei K, Yuan J. A novel bearing fault diagnosis method based joint attention adversarial domain adaptation. Reliab Eng Syst Saf 2023;237:109345.

[13] Liu S, Jiang H, Wu Z, Yi Z, Wang R. Intelligent fault diagnosis of rotating machinery using a multi-source domain adaptation network with adversarial discrepancy matching. Reliab Eng Syst Saf 2023;231:109036.

[14] Kim YC, Lee J, Kim T, Baek J, Ko JU, Jung JH, et al. Gradient alignment based partial domain adaptation (GAPDA) using a domain knowledge filter for fault diagnosis of bearing. Reliab Eng Syst Saf 2024;110293.

[15] Yang B, Lee C-G, Lei Y, Li N, Lu N. Deep partial transfer learning network: A method to selectively transfer diagnostic knowledge across related machines. Mech Syst Signal Process 2021;156:107618.

[16] Li W, Gu S, Zhang X, Chen T. Transfer learning for process fault diagnosis: Knowledge transfer from simulation to physical processes. Comput Chem Eng 2020;139:106904.

[17] Fan Z, Xu Q, Jiang C, Ding SX. Deep mixed domain generalization network for intelligent fault diagnosis under unseen conditions. IEEE Trans Ind Electron 2023.

[18] Shi Y, Deng A, Deng M, Xu M, Liu Y, Ding X, et al. Domain augmentation generalization network for real-time fault diagnosis under unseen working conditions. Reliab Eng Syst Saf 2023;235:109188.

[19] Yang Y, Yin J, Zheng H, Li Y, Xu M, Chen Y. Learn generalization feature via convolutional neural network: A fault diagnosis scheme toward unseen operating conditions. IEEE Access 2020;8:91103–15.

[20] Chen L, Li Q, Shen C, Zhu J, Wang D, Xia M. Adversarial domain-invariant generalization: A generic domain-regressive framework for bearing fault diagnosis under unseen conditions. IEEE Trans Ind Inf 2021;18(3):1790–800.

[21] Wang R, Huang W, Lu Y, Zhang X, Wang J, Ding C, et al. A novel domain generalization network with multidomain specific auxiliary classifiers for machinery fault diagnosis under unseen working conditions. Reliab Eng Syst Saf 2023;238:109463.

[22] Wang J, Ren H, Shen C, Huang W, Zhu Z. Multi-scale style generative and adversarial contrastive networks for single domain generalization fault diagnosis. Reliab Eng Syst Saf 2024;243:109879.

[23] Flammini F. Digital twins as run-time predictive models for the resilience of cyber-physical systems: a conceptual framework. Phil Trans R Soc A 2021;379(2207):20200369.

[24] Ma L, Jiang B, Xiao L, Lu N. Digital twin-assisted enhanced meta-transfer learning for rolling bearing fault diagnosis. Mech Syst Signal Process 2023;200:110490.

[25] Dong Y, Li Y, Zheng H, Wang R, Xu M. A new dynamic model and transfer learning based intelligent fault diagnosis framework for rolling element bearings race faults: Solving the small sample problem. ISA Trans 2021.

[26] Jia S, Li Y, Wang X, Sun D, Deng Z. Deep causal factorization network: A novel domain generalization method for cross-machine bearing fault diagnosis. Mech Syst Signal Process 2023;192:110228.

[27] Xia M, Shao H, Williams D, Lu S, Shu L, de Silva CW. Intelligent fault diagnosis of machinery using digital twin-assisted deep transfer learning. Reliab Eng Syst Saf 2021;215:107938.

[28] Mishra R, Choudhary A, Fatima S, Mohanty A, Panigrahi B. A self-adaptive multiple-fault diagnosis system for rolling element bearings. Meas Sci Technol 2022;33(12):125018.

[29] Huo Z, Zhang Y, Sath R, Shu L. Self-adaptive fault diagnosis of roller bearings using infrared thermal images. In: IECON 2017-43rd annual conference of the IEEE industrial electronics society. IEEE; 2017, p. 6113–8.

[30] Weyns D. An Introduction to Self-Adaptive Systems: A Contemporary Software Engineering Perspective. John Wiley & Sons; 2020.

[31] Kephart J, Chess D. The vision of autonomic computing. Computer 2003;36(1):41–50. http://dx.doi.org/10.1109/MC.2003.1160055.

[32] Arcaini P, Riccobene E, Scandurra P. Modeling and analyzing MAPE-K feedback loops for self-adaptation. In: 2015 IEEE/ACM 10th international symposium on software engineering for adaptive and self-managing systems. IEEE; 2015, p. 13–23.

[33] Bozhinoski D, Garzon Oviedo M, Hammoudeh Garcia N, Deshpande H, van der Hoorn G, Tjerngren J, et al. MROS: runtime adaptation for robot control architectures. Adv Robot 2022;1–17.

[34] Malburg L, Hoffmann M, Bergmann R. Applying MAPE-K control loops for adaptive workflow management in smart factories. J Intell Inf Syst 2023;61(1):83–111. http://dx.doi.org/10.1007/s10844-022-00766-w.

[35] Weyns D, Iftikhar MU, Hughes D, Matthys N. Applying architecture-based adaptation to automate the management of internet-of-things. In: Cuesta CE, Garlan D, Pérez J, editors. Software architecture. Cham: Springer International Publishing; 2018, p. 49–67.

[36] Li J, Huang R, He G, Liao Y, Wang Z, Li W. A two-stage transfer adversarial network for intelligent fault diagnosis of rotating machinery with multiple new faults. IEEE/ASME Trans Mechatronics 2020;26(3):1591–601.

[37] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. Adv Neural Inf Process Syst 2014;27.

[38] Georgiou TT. What is a natural notion of distance between power spectral density functions? In: 2007 European control conference. IEEE; 2007, p. 358–61.

[39] Lessmeier C, Kimotho J, Zimmer D, Sextro W. KAt-DataCenter, chair of design and drive technology, paderborn university. 2019, https://mb.uni-paderborn.de/kat/forschung/kat-datacenter/bearing-datacenter.

[40] Case Western Reserve University (CWRU) Bearing Data Center, Case Western Reserve University Available: https://csegroups.case.edu/bearingdatacenter/pages/welcome-casewestern-reserve-university-bearing-data-center-website.

[41] Bechhoefer E. MFPT bearing fault data sets. 2012, http://mfpt.org/fault-data-sets/.

[42] Azari MS, Ricci L, Santini S, et al. An intelligent diagnostic framework based on digital twins and partial transfer learning: methodology and industrial application. 2024, http://dx.doi.org/10.36227/techrxiv.172263061.16209264/v1, TechRxiv, Registration in progress.

[43] Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, et al. Domain-adversarial training of neural networks. J Mach Learn Res 2016;17(59):1–35.

[44] Gazizulin D, Cohen E, Bortman J, Klein R. Critical rotating machinery protection by integration of a “fuse” bearing. Int J Crit Infrastruct Prot 2019;27:100305.