Increasing Robustness of Data-Driven Fault Diagnostics with Knowledge Graphs

Maximilian-Peter Radtke[1], Marco F. Huber[2,3], and Jürgen Bock[1]

[1] *Technische Hochschule Ingolstadt, Ingolstadt, Germany*
maximilian-peter.radtke@thi.de
juergen.bock@thi.de

[2] *Institute of Industrial Manufacturing and Management IFF, University of Stuttgart, 70569 Stuttgart, Germany*

[3] *Fraunhofer Institute of Manufacturing Engineering and Automation IPA, 70569 Stuttgart, Germany*
marco.huber@ieee.org

Abstract

In the realm of prognostics and health management (PHM), it is common to possess not only process data but also domain knowledge, which, if integrated into data-driven algorithms, can aid in solving specific tasks. This paper explores the integration of knowledge graphs (KGs) into deep learning models to develop a more resilient approach capable of handling domain shifts, such as variations in machine operating conditions. We present and assess a KG-enhanced deep learning approach in a representative PHM use case, demonstrating its effectiveness by incorporating domain-invariant knowledge through the KG. Furthermore, we provide guidance for constructing a comprehensive hierarchical KG representation that preserves semantic information while facilitating numerical representation. The experimental results showcase the improved performance and domain shift robustness of the KG-enhanced approach in fault diagnostics.

1. Introduction

Methods for machinery fault diagnostics can be broadly categorized into two types: *knowledge-based* and *data-driven* approaches. Knowledge-based approaches rely on system-specific knowledge to identify and diagnose faults, while data-driven methods utilize previously observed data for classification purposes. Knowledge-based models are easily interpretable and can accurately represent a system when the underlying physical and logical relationships are well understood. However, they require in-depth knowledge about the specific system and are often designed for a single, isolated case, making generalization difficult. On the other hand, data-driven models can be applied to different systems if sufficient data is available. They offer flexibility but are reliant on the availability of data, which can often be expensive to obtain. Furthermore, their performance may decrease significantly when confronted with scenarios that deviate too far from the training data distribution.

While these two categories traditionally had little overlap, there is a growing development of *hybrid approaches* that aim to combine the strengths of both methods and mitigate their respective limitations (Hagmeyer, Zeiler, & Huber, 2022). In a general PHM setting these hybrid approaches are mainly about combining data-driven approaches with physical knowledge. For instance, (Jadhav, Deodhar, Gupta, & Runkana, 2022) use physics informed neural networks (NNs) for monitoring the health of an air preheater, (Deng, Nguyen, Gogu, Morio, & Medjaher, 2022) inform an NN with the stiffness of the bearing it aims to model, and (Chao, Kulkarni, Goebel, & Fink, 2022) extend the feature space with physical properties of the underlying system. However, the knowledge-driven category encompasses a broader range of approaches. These methods rely on the symbolic representation of domain-specific knowledge, such as the knowledge representation-based approach proposed by (Cao, Samet, Zanni-Merk, de Beuvron, & Reich, 2019) or other approaches based on KGs summarized in (Xia, Zheng, Li, Gao, & Wang, 2022).

In the following discussion, we focus on the integration of knowledge representation and data-driven deep learning to develop a more resilient model capable of handling domain shifts, e.g., a machine that functions under different operating conditions. Domain shifts are not only an issue in PHM applications but are a common challenge in deep learning. Models tend to overfit on the domain they are trained on and do not generalize as well to closely related domains. Different approaches have been proposed to deal with this and have been applied to PHM applications. For example, (Zheng et al., 2020) use signal processing techniques guided by a priori knowledge to create a domain invariant feature representation, (Rahat et al., 2022) apply domain adversarial NNs, and (Peng, Liu, & Gryllias, 2022) align the different domains with cyclic spectrum correlation analysis.

An alternative approach is to incorporate domain-invariant knowledge, represented by a KG, into the modeling process. This type of approach has been applied mostly to computer vision tasks, e.g., (Jayathilaka, Mu, & Sattler, 2021) inform their model with n-ball concept embeddings and (Gebru, Hoffman, & Fei-Fei, 2017) use the WordNet KG for improving fine-grained image classification. An extensive overview of this topic is given by (Monka, Halilaj, & Rettinger, 2022).

In contrast to computer vision tasks, a fault diagnostics case typically has fewer classes and general KGs like WordNet are not appropriate since they contain no relevant information for the problem at hand. By adopting the overarching methodology presented by (Monka, Halilaj, Schmid, & Rettinger, 2021), we address common challenges encountered in fault diagnostics scenarios, offering a two-fold contribution:

- Application and evaluation of a deep learning approach enhanced by a KG in a representative PHM use case.
- Guidance for creating a general hierarchical KG, which can be easily represented numerically while preserving its semantic information.

The remainder of the paper is structured as follows. We start by clearly formulating the discussed problem in Sec. 2. Next, we introduce the proposed approach in Sec. 3, describe the experimental setup in Sec. 4 and discuss our results in Sec. 5. The paper closes with conclusions and an outlook on future work. The code used for the experiments is available at https://github.com/AImotion-Bavaria/FaultDiagnosticsKG.

2. PROBLEM FORMULATION

Our goal is to train a classifier for the task of *fault classification* that has a consistent performance across changing domains. A domain $\mathcal{D}$ is composed of a feature space $\mathcal{X}$ and a marginal distribution $P(X)$ over $\mathcal{X}$, where X is a set of instances $X = \{x_1, ..., x_N\}$. A task $\mathcal{T}$ consists of a label space $\mathcal{Y}$ and a decision function f, which is to be learned from the sample data. For the task of fault classification we call $\mathcal{Y}$ the *condition space* of the observed system. We assume that $\mathcal{Y}$ has some kind of structure, which we have knowledge about, and which can be encoded in a KG. We will look at two scenarios, where we have a source domain $\mathcal{D}_S$ for which we have data, e.g., vibration data under certain operating conditions of a machine, and a target domain $\mathcal{D}_T$ for which no data (*scenario 1*) or very little data (*scenario 2*) is available. The goal is to be able to solve one task $\mathcal{T}$ on both domains, i.e., train one decision function $f : \mathcal{X}_S \cup \mathcal{X}_T \to \mathcal{Y}$. The difficulty of this task depends strongly on the difference between the domains' marginal distributions. Both scenarios are special cases of the more general transfer learning framework (Pan & Yang, 2010; Zhuang et al., 2020).

We approach this task by creating a d_E-dimensional feature representation via an encoding NN $\mathrm{Enc} : \mathcal{X} \to \mathbb{R}^{d_E}$, which is robust with respect to domain changes, i.e., different marginal distributions $P(X_S), P(X_T)$, and can serve as an input to the decision function $f(\mathrm{Enc}(x))$.

In scenario 1, with no data available on the target domain, $f(\cdot)$ is trained once on the source domain and directly evaluated on the target domain. In scenario 2, where some data from the target domain is available, $f(\cdot)$ is fine tuned by few-shot learning, i.e., we take a few samples from the target domain to retrain $f(\cdot)$.

3. APPROACH

To train $\mathrm{Enc}(\cdot)$ we apply the approach by (Monka et al., 2021), which combines the supervised contrastive loss with KGs.

3.1. Supervised Contrastive Loss

The underlying concept of contrastive learning involves selecting a sample as an anchor and then bringing positive samples (with the same label as the anchor) closer while pushing negative samples (with different labels) further away in the embedding space. This process trains an NN to create a feature representation of the input. Typically, contrastive learning is applied in a self-supervised setting, where all samples other than the anchor are assumed to be negative. The positive samples are generated by applying augmentation techniques on the anchor sample (Jaiswal, Babu, Zadeh, Banerjee, & Makedon, 2020). However, if we have access to labeled data, we can take advantage of a supervised formulation introduced by (Khosla et al., 2020). Thus, the loss for training is given by

$$\mathcal{L}^{SC} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left({}^{d(z_i, z_p)}/_{\tau}\right)}{\sum_{a \in A(i)} \exp\left({}^{d(z_i, z_a)}/_{\tau}\right)} \,, \quad (1)$$

with $I = \{1, ..., N\}$ the set of indices of samples, $A(i) = I \setminus \{i\}$, the positive samples $P(i) = \{p \in A(i) \mid y_p = y_i\}$, a scalar temperature parameter τ, a distance metric $d(\cdot, \cdot)$, and the output of the network z_k. Empirical evidence suggests, that using $z_k = \mathrm{Enc}(x_k)$ does not give us the optimal feature representation. Rather a projection network $\mathrm{Proj}(\cdot)$ is added to the encoder so that $z_k = \mathrm{Proj}(\mathrm{Enc}(x_k)) \in \mathbb{R}^{d_P}$ (Chen, Kornblith, Norouzi, & Hinton, 2020). According to (Jing, Vincent, LeCun, & Tian, 2022) the projection network prevents the problem of dimensional collapse, which occurs when duplicate information about the representation is encoded between different dimensions. Typically, $\mathrm{Proj}(\cdot)$ consists of a single layer, or at most a very shallow, multilayer perceptron (MLP) and is discarded after training. In accordance to the original paper we choose the cosine similarity as the distance metric $d(\cdot, \cdot)$ and $\tau = 0.1$.

3.2. Knowledge Graphs and Embeddings

To include domain invariant knowledge into this loss function, we need a way of representing this knowledge first. This can be achieved with a KG, which can be defined as "*a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent potentially different relations between these entities*" (Hogan et al., 2021). KGs can be represented as triples of the form (s, p, o) with subjects and objects from a set of entities $s, o \in \mathcal{E}$ and predicates or edges from the set of relations $p \in \mathcal{R}$. An exemplary triple could be (`Berlin`, `capitalOf`, `Germany`), where the predicate `capitalOf` relates the subject `Berlin` with the object `Germany`. The semantics of the triples in a KG are determined by the creator, allowing for flexibility and customization. However, commonly used vocabularies are available to enhance the accessibility and interpretability of KGs, such as the Resource Description Framework Schema (RDFS)[1]. These vocabularies provide a standardized framework for organizing and describing information within KGs. By adopting such vocabularies, KG creators can ensure that their graphs are more easily understood and interoperable with other KGs.

Our objective is to leverage KGs for effectively representing the structure of the condition space $\mathcal{Y}$. Specifically, our focus lies in modelling the hierarchical structure of various system conditions, by modelling all entity relationships using the `rdfs:subClassOf` predicate, which relates a subject to an object entity by stating that all instances of the subject are instances of the object. In a typical scenario, we possess knowledge about general fault types, which can further be divided into more specific sub-faults displaying variations in size or severity. For instance, a bearing can either be healthy or faulty and a faulty condition can further be specified by stating the position and the size of the defect. By utilizing KGs, we aim to capture and illustrate this hierarchical relationship among fault conditions.

KGs are a purely symbolic representation of the domain-invariant knowledge. For use in combination with an NN a numerical representation is necessary, i.e., we need to find a mapping $h : \mathcal{KG} \rightarrow \mathbb{R}^{|\mathcal{E}| \times d_{\mathrm{KG}}}$, where $\mathcal{KG}$ is the space of KGs and d_{KG} is the embedding dimension. If no information about the structure of the KG is available, this is a difficult task and it is unclear if current embedding methods even maintain the KG's semantic information (Jain, Kalo, Balke, & Krestel, 2021). But since we only use the `rdfs:subClassOf` predicate, which is transitive according to the RDFS specification, i.e., if (`A`, `rdfs:subClassOf`, `B`) and (`B`, `rdfs:subClassOf`, `C`) then (`A`, `rdfs:subClassOf`, `C`), we can embed the complete KG in a matrix E of size $|\mathcal{E}| \times |\mathcal{E}|$ without loosing any information. The entries of E are given by $E_{i,j} = 1$ if (e_i, `rdfs:subClassOf`, e_j) holds and 0 otherwise. In doing so, we have an embedding for each condition informed by knowledge about the condition hierarchy. For condition i we write this embedding as $h_{KG,i} = E_i$, with $E_i \in \{0, 1\}^{|\mathcal{E}|}$ being the i-th row of the embedding matrix E.

3.3. Combining Domain Knowledge Graphs and Supervised Contrastive Learning

We use this numerical representation of the knowledge about the condition space to create a more robust feature representation regarding domain changes. For inclusion into our network we use the method by (Monka et al., 2021) and adjust $\mathcal{L}^{SC}$ to

$$\mathcal{L}^{KG} = \sum_{i \in I} \mathcal{L}_i^{KG} \,, \tag{2}$$

$$\mathcal{L}_i^{KG} = \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(d(h_{KG,i}, z_p)/\tau\right)}{\sum_{a \in A(i)} \exp\left(d(h_{KG,i}, z_a)/\tau\right)} \,, \tag{3}$$

where $h_{KG,i}$ is the domain invariant representation of the class label y_i of x_i generated by a KG. Hence, we train $\mathrm{Enc}(\cdot)$ by minimizing $\mathcal{L}^{KG}$ for $\mathrm{Proj}(\mathrm{Enc}(\cdot))$ to create a feature representation, which can be passed to the decision function $f(\mathrm{Enc}(\cdot))$. The general procedure is visualized in Fig. 1.

While this approach was originally proposed for extending $\mathcal{L}^{SC}$, it can readily be applied to a wide set of deep metric learning loss functions, i.e., where the learning goal is the minimization of a distance between vectors.

Note that we include no data augmentation in our approach, which would be the standard approach for contrastive learning and which was also included by (Monka et al., 2021). This is deliberate because we want to study the effect of the influence of the KG in particular. Nevertheless, data augmentation techniques specific to a domain can readily be included, e.g., from (Ding, Zhuang, Ding, & Jia, 2022).

4. EXPERIMENTS

In the following we outline the experimental setup, including datasets, case studies and model configurations, used to assess the performance of the proposed KG-enhanced deep learning approach.

[1] `https://www.w3.org/TR/rdf-schema, accessed 16/05/2023`

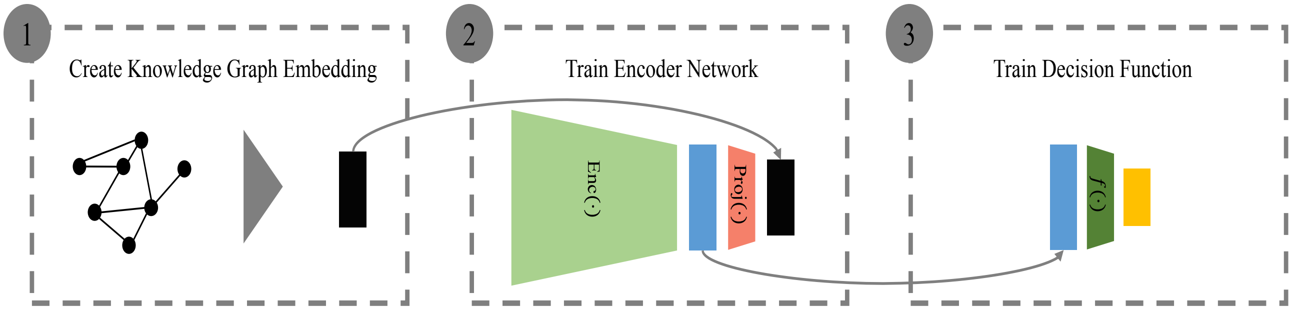

Figure 1. For KG-enhanced deep learning we (1) create a domain invariant KG and embed it into a numerical representation. Then (2) we train an encoding network $\mathrm{Enc}(\cdot)$ based on the KG embedding. Finally (3), we train a decision function $f(\cdot)$ for the classification task based on the feature representation created by the encoding network.

4.1. Data and Preprocessing

Our approach is evaluated on a bearing fault classification task using the widely recognized Case Western Reserve University (CWRU) dataset[2]. The system under observation consists of a reliance electric motor with two bearings of distinct geometries positioned at the drive end (DE) and the fan end (FE) of the motor shaft. Vibration signals were captured for both bearings, with sensors placed near each respective bearing, to record various faults including outer race (OR), inner race (IR), and ball (B), each with different sizes (7, 14, 21). The sample frequency for these recordings was set to 12 kHz. Additionally, vibration signals were also recorded for the healthy state of the system at a higher sample frequency of 48 kHz, which we downsampled to 12 kHz for the purpose of our evaluation. The dataset includes vibration signals for all ten health conditions under different motor loads (0, 1, 2, 3), which impact the shaft speed.

To feed our models, we divided the original signals into non-overlapping sequences of length 512, thereby resulting in roughly 13,000 samples per bearing. No further preprocessing was applied to the data prior to training and evaluation.

4.2. Case Studies

The way the data is recorded enables us to evaluate our approach for the two scenarios described in Sec. 2. With $L = \{0, 1, 2, 3\}$ we denote the set of different motor loads, $B = \{FE, DE\}$ are the two different bearings, $l \in L$ and $b \in B$.

4.2.1. Case Study 1: Different Motor Loads

The first case study, inspired by (Rombach, Michau, & Fink, 2021), simulates scenario 1, where a decision function is evaluated on a completely unseen target domain. We test how well a model, trained on certain motor loads, performs when it is confronted with data recorded under a different load. We train our model on the source domain $\mathcal{D}_{b,L\setminus\{l\}}$, i.e., on data from bearing b for all loads from L except l, and evaluate it on the target domain $\mathcal{D}_{b,l}$, i.e., on data from bearing b for load l. To additionally evaluate how well the model performs on the source domain we perform a 80/20 train/test split on $\mathcal{D}_{b,L\setminus\{l\}}$. In contrast to (Rombach et al., 2021), our approach extends the target domains to include the highest (3) and lowest (0) motor loads. We assume that these extreme load conditions pose greater challenges for generalization compared to situations where the model has access to loads, which are higher and lower than the target domain. By incorporating these additional target domains, we aim to evaluate the model's ability to adapt and generalize across the entire load spectrum.

4.2.2. Case Study 2: Different Bearings

The second case study focuses on simulating scenario 2, where a decision function is fine-tuned through few-shot learning on the target domain. We aim to evaluate the model's performance when presented with vibrations from a different bearing. To simulate this scenario, we train the model on the dataset $\mathcal{D}_{B\setminus b}$ (excluding the target bearing) and evaluate its performance on the dataset $\mathcal{D}_b$ (specific to the target bearing). We acknowledge that this domain shift, caused by the introduction of vibrations from a different bearing, is significantly more challenging compared to a mere change in motor loads. To address this challenge, we employ few-shot learning techniques. We fine-tune our decision function using a limited number of shots, specifically 1, 2, 4, 8, 16, 32, or 64 shots. Each shot represents one sample from each class within the target domain. This adaptation process enables the model to better accommodate and understand the previously unseen bearing. Again, the model is additionally evaluated on the source domain on a 80/20 train test split.

4.3. Bearing Fault Knowledge Graph

The condition space of the bearing fault classification task consists of 10 different conditions and can be effectively represented by a hierarchical KG. The nine different fault conditions are categorized based on their fault size (`Small` - 7, `Medium` - 14, `Large` - 21) and fault type (`InnerRace`,

[2] `https://engineering.case.edu/bearingdatacenter, accessed 16/05/2023`

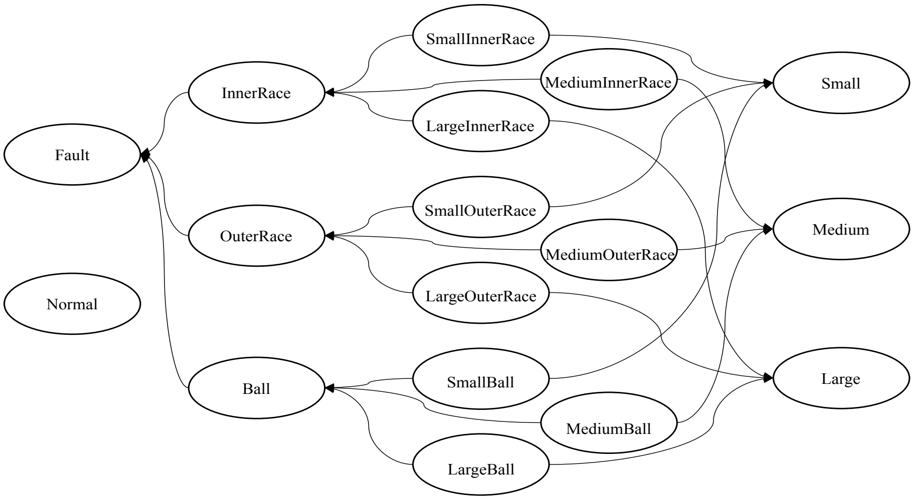

Figure 2. KG representing the hierarchical structure of the condition space for a bearing fault classification task. All arrows represent `rdfs:subClassOf` relationships between entities, where the arrow points in the direction of the object.

`OuterRace`, `Ball`). All fault conditions are subclasses of `Fault`. The healthy condition is related to no other entities. This hierarchical structure establishes an invariant KG that remains consistent across different domains explored in the two case studies. Thus, a triple such as (`LargeInnerRace`, `rdfs:subClassOf`, `InnerRace`) holds true regardless of the specific load or bearing from which the vibration signal originates. The complete KG comprises 21 triples and is depicted in Figure 2. Since we have 17 entities, we can embed the complete KG into a 17 × 17 matrix by utilizing the approach outlined in Section 3.2. Each condition is represented by the respective row of the embedding.

4.4. Model Configurations and Preprocessing

We conduct a comparative analysis of our proposed approach, which utilizes the supervised contrastive loss with a KG embedding (SC+KG), against other established loss functions. Specifically, we compare it to the standard cross-entropy (CE), the conventional supervised contrastive loss from Equation (1) (SC), and the semi-hard implementation of the triplet loss (TL) using the L_2 distance measure (Schroff, Kalenichenko, & Philbin, 2015). The inclusion of TL is motivated by its strong performance in a previous evaluation on a load domain shift case study conducted by (Rombach et al., 2021). It is worth noting that TL can be considered as a special case of SC when only one negative and positive sample is taken into account (Khosla et al., 2020). Consequently, TL can be evaluated in the same manner as SC and SC+KG. Given that TL has access to less information during the learning process, we anticipate that its performance may be comparatively lower than that of SC.

The architecture for the encoding network $\mathrm{Enc}(\cdot)$ is the same for all loss functions and is inspired by (Rombach et al., 2021). It consists of four 1-d convolutional layers (64, 32, 16, 8 kernels) with kernel size 12. After each layer we use the Leaky Rectified Linear Unit (ReLU) as an activation function with a negative slope of 0.5, max pooling with a stride of 2 and dropout with $p = 0.1$. The output is then flattened, fed into a fully connected layer with output dimension $d_E = 50$ and activated by a Leaky ReLU. The output of this layer is our feature representation generated by $\mathrm{Enc}(\cdot)$. For CE this representation is directly handed over to the decision function $f_{\mathrm{CE}}(\cdot)$, which consists of a single linear layer with output dimension 10—the number of classes. We can therefore directly train $f_{\mathrm{CE}}(\mathrm{Enc}(\cdot))$. For the other loss functions ($O = \{\mathrm{SC}, \mathrm{SC{+}KG}, \mathrm{TL}\}$) the output is first fed into a projection network $\mathrm{Proj}_O(\cdot)$ consisting of a linear layer with an output dimension of $d_P = d_{KG} = 17$—the size of the numerical representation of each class induced by the KG. As stated in Sec. 3.1, $\mathrm{Proj}_O(\mathrm{Enc}(\cdot))$ is trained using either SC, SC+KG or TL and afterwards, $\mathrm{Proj}_O(\cdot)$ is discarded. For fault classification, an additional decision function $f_O(\cdot)$ consisting of one linear layer with output dimension 10 is trained using the cross-entropy loss.

The networks $f_{\mathrm{CE}}(\mathrm{Enc}(\cdot))$ and $\mathrm{Proj}_O(\mathrm{Enc}(\cdot))$ were trained for 100 epochs, followed by an additional 20 epochs of training specifically for $f_O(\cdot)$. To mitigate the impact of randomness, all models were trained 10 times using the same 10 random seeds. The Adam optimizer was employed during the training process with learning rate $\gamma = 0.001$.

In the few-shot scenario, the decision functions were fine-tuned for 200 epochs. As the selection of samples for fine-tuning significantly affects the performance, we repeated the fine-tuning process 10 times, each time with different random seeds, to ensure robustness and capture the influence of various sample combinations.

Table 1. Accuracies in percentages for the domain shift between different loads ($\mathcal{D}_{b,L\setminus\{l\}} \rightarrow \mathcal{D}_{b,l}$). The left column of each domain shift depicts the performance on the source domain and the right column on the target domain. Standard deviations are given in brackets for 10 runs and the best value for each domain is printed in bold.

$\mathcal{D}_{\text{DE},L\setminus\{l\}} \rightarrow \mathcal{D}_{\text{DE},l}$	$\mathcal{D}_{\text{DE},L\setminus\{0\}}$	$\mathcal{D}_{\text{DE},0}$	$\mathcal{D}_{\text{DE},L\setminus\{1\}}$	$\mathcal{D}_{\text{DE},1}$	$\mathcal{D}_{\text{DE},L\setminus\{2\}}$	$\mathcal{D}_{\text{DE},2}$	$\mathcal{D}_{\text{DE},L\setminus\{3\}}$	$\mathcal{D}_{\text{DE},3}$
CE	99.8 (0.2)	92.5 (2.6)	99.6 (0.3)	98.6 (1.5)	99.6 (0.3)	99.6 (0.5)	99.7 (0.2)	94.6 (2.7)
TL	**99.9 (0.1)**	91.1 (2.3)	99.6 (0.2)	99.2 (0.3)	**99.7 (0.2)**	**99.9 (0.1)**	**99.8 (0.1)**	95.3 (2.8)
SC	99.8 (0.1)	93.2 (2.2)	99.6 (0.2)	99.4 (0.3)	99.6 (0.2)	99.8 (0.1)	99.7 (0.2)	93.9 (1.8)
SC+KG	**99.9 (0.1)**	**94.9 (1.9)**	**99.7 (0.1)**	**99.7 (0.2)**	**99.7 (0.1)**	**99.9 (0.1)**	**99.8 (0.2)**	**96.8 (1.7)**

$\mathcal{D}_{\text{FE},L\setminus\{l\}} \rightarrow \mathcal{D}_{\text{FE},l}$	$\mathcal{D}_{\text{FE},L\setminus\{0\}}$	$\mathcal{D}_{\text{FE},0}$	$\mathcal{D}_{\text{FE},L\setminus\{1\}}$	$\mathcal{D}_{\text{FE},1}$	$\mathcal{D}_{\text{FE},L\setminus\{2\}}$	$\mathcal{D}_{\text{FE},2}$	$\mathcal{D}_{\text{FE},L\setminus\{3\}}$	$\mathcal{D}_{\text{FE},3}$
CE	99.6 (0.3)	81.3 (2.3)	99.3 (0.4)	98.1 (1.2)	99.5 (0.1)	**99.1 (0.3)**	99.4 (0.3)	90.3 (2.2)
TL	99.6 (0.2)	81.1 (3.3)	**99.7 (0.2)**	98.6 (0.5)	99.5 (0.2)	98.8 (0.4)	**99.6 (0.1)**	92.1 (1.9)
SC	99.6 (0.2)	83.4 (2.3)	99.6 (0.2)	**98.9 (0.3)**	99.5 (0.1)	99.0 (0.4)	99.5 (0.1)	89.2 (2.3)
SC+KG	**99.7 (0.2)**	**83.5 (2.2)**	99.6 (0.1)	**98.9 (0.3)**	**99.7 (0.1)**	**99.1 (0.3)**	**99.6 (0.2)**	**92.8 (1.6)**

5. RESULTS

The results showcase the findings of the two case studies that evaluated the performance of the proposed KG-enhanced deep learning approach in comparison to the baseline models.

5.1. Case Study 1: Different Motor Loads

Table 1 presents the results obtained when dealing with a domain shift to a different load in the fault diagnostics task. All evaluated loss functions demonstrate nearly perfect accuracy on the source domain. The performance on the target domain can be divided into two cases.

In the first case, the target domain is situated between the extreme load ranges of the source domain: for instance the domain shift from $\mathcal{D}_{\text{DE},L\setminus\{2\}}$ to $\mathcal{D}_{\text{DE},2}$. The accuracies of the source domain are quite similar to those of the target domain, indicating that this domain shift is relatively easy. This is expected since the models were trained on data that encompassed both higher and lower loads. To solve the target domain, the model only needs to interpolate between these loads. However, we still observe a performance gain with TL, SC, and SC+KG, compared to CE. Among these methods, SC+KG achieves the best performance, especially when generalizing to load 1.

In the second case, the target load is either higher or lower than anything the model has encountered before. Particularly when generalizing to the lower motor load, i.e., $l = 0$, we observe a significant decline in performance, up to 9 % for the DE bearing and up to 18 % for the FE bearing. In this second case, SC+KG consistently outperforms CE by more than 2 %. SC and TL show less consistency in their performance compared to SC+KG. Interestingly, TL performs better when generalizing to a higher load, almost matching the performance of SC+KG, than when generalizing to a lower load. On the other hand, SC performs better when generalizing to a lower load. Currently, we have no explanation for this observation. In general, we find that handling the motor load domain shift for the FE bearing is more challenging than for the DE bearing.

When examining the standard deviations for both cases, we observe that SC+KG consistently exhibits the lowest values. This is particularly evident for the domain shift to load 1, where the volatility of the results is up to seven times higher for CE compared to SC+KG. From this, we can conclude that SC+KG not only generates more robust representations in the face of domain changes but also in terms of performance.

5.2. Case Study 2: Different Bearings

The results for the domain shift to a different bearing are depicted in Table 2. Due to the nearly perfect performance on the source domain across all loss functions used, these results are not included.

It is evident that directly generalizing to a different bearing yields poor results. When applying zero shots for fine-tuning the decision function, both domain shifts result in accuracies close to random. However, by employing multiple shots for fine-tuning, a substantial performance improvement is observed. Accuracies of up to 85 % are achieved when 64 samples are used to fine-tune the linear classifier.

Regarding the transfer task from the DE bearing to the FE bearing, the knowledge-enhanced approach SC+KG exhibits the best performance, particularly when utilizing a limited number of shots. TL demonstrates comparable performance to SC+KG but experiences a relative decline in performance as more shots are employed. When transferring in the opposite direction, from FE to DE, TL holds a slight advantage over SC+KG, although their performance is nearly identical. For both shifts, SC+KG outperforms CE by approximately 5 % for a low number of shots, and their performances converge as more shots are used.

Additionally, it is worth noting the disparity between vanilla SC and SC+KG. The KG-enhanced method significantly en-

Table 2. Accuracies in percentages for the domain shift between different bearings ($\mathcal{D}_{B\setminus\{b\}} \rightarrow \mathcal{D}_b$) after fine tuning the decision function with few-shot learning. Standard deviations are given in brackets for 10 runs and the best value for each column is printed in bold. The columns indicate the number of samples per class used for fine tuning.

$\mathcal{D}_{\text{DE}} \rightarrow \mathcal{D}_{\text{FE}}$	0	1	2	4	8	16	32	64
CE	13.3 (0.1)	24.5 (4.6)	35.1 (3.7)	43.5 (3.2)	59.9 (2.8)	72.2 (1.3)	79.6 (0.9)	84.8 (0.6)
TL	13.6 (0.1)	28.0 (4.0)	**40.4 (3.9)**	**48.8 (2.7)**	57.6 (2.5)	68.0 (1.6)	76.4 (1.1)	81.6 (0.8)
SC	14.4 (0.1)	24.8 (4.0)	33.2 (3.7)	37.8 (3.3)	53.9 (2.6)	67.8 (1.7)	76.8 (1.0)	82.3 (0.5)
SC+KG	**14.7 (0.1)**	**28.7 (5.1)**	**40.4 (3.4)**	48.7 (3.7)	**61.3 (2.7)**	**72.4 (1.5)**	**80.1 (1.0)**	**85.1 (0.6)**
$\mathcal{D}_{\text{FE}} \rightarrow \mathcal{D}_{\text{DE}}$	0	1	2	4	8	16	32	64
CE	15.4 (0.3)	36.3 (4.5)	45.4 (3.3)	52.2 (2.6)	60.8 (2.2)	71.2 (1.4)	78.8 (1.0)	**84.8 (0.6)**
TL	14.3 (0.1)	**42.2 (4.8)**	**52.5 (4.3)**	**60.7 (3.2)**	**68.5 (1.7)**	**75.1 (1.3)**	**80.0 (1.0)**	84.3 (0.6)
SC	**18.0 (0.1)**	34.4 (4.8)	41.4 (4.3)	46.5 (3.4)	58.5 (2.2)	69.3 (1.3)	76.2 (1.1)	81.3 (0.7)
SC+KG	15.7 (0.1)	41.8 (5.9)	51.4 (4.0)	58.7 (2.7)	67.4 (2.0)	74.8 (1.2)	79.7 (1.0)	83.9 (0.5)

hances accuracy by up to 10 % for a low number of shots, and still achieves an improvement of around 3 % when utilizing 64 samples.

In general, there is considerable variability in the results when employing only a small number of samples for fine-tuning, as indicated by the high values for the standard deviation. This variability is expected since the choice of samples plays a crucial role in model performance. As more samples are used for fine-tuning, the variability diminishes.

6. CONCLUSION

We evaluated the effectiveness of KG-enhanced deep learning in two scenarios: domain generalization to different operating conditions and few-shot learning for fine-tuning the decision function on a different domain. The results demonstrate the robustness and improved performance of KG-enhanced deep learning in creating feature representations that are independent of the source domain. Although alternative loss functions occasionally exhibited slight improvements, they significantly underperformed in other cases. This further highlights the stability of the proposed approach, which achieved robust results across all evaluated scenarios.

It is important to acknowledge that the practical relevance of the few-shot learning example for bearing classification may be limited. In real-world scenarios, when we have only one sample, we often have access to multiple samples of a specific condition. This is because even a signal with a duration of just one second can be divided into multiple samples if the sampling frequency is sufficiently high. However, despite this limitation, the obtained results from the few-shot learning experiments remain promising and serve as a motivation to further refine and apply the approach to other tasks where the acquisition of each additional sample comes with a substantial cost.

We believe that the KG-enhanced approach is especially relevant when a large condition space is considered. Many different kinds of system conditions make it hard to gather data for all operating conditions, further underscoring the importance of a robust feature representation, and provide a rich hierarchical structure that can be represented by a KG. Incorporating more knowledge about the condition space, exploring other metric loss functions and KG embedding techniques, and adapting the approach to a prognostics setting are potential paths for future work to further boost the effectiveness and versatility of KG-enhanced deep learning.

This research highlights the ubiquity of domain shifts, which remain one of the major drawbacks of deep learning applied to fault diagnostics. By addressing this challenge through KG-enhanced deep learning, this work contributes to overcoming the limitations of traditional approaches and demonstrates the significance of considering knowledge-driven methodologies in fault diagnostics and beyond.

REFERENCES

Cao, Q., Samet, A., Zanni-Merk, C., de Beuvron, F. d. B., & Reich, C. (2019). An ontology-based approach for failure classification in predictive maintenance using fuzzy c-means and swrl rules. *Procedia Computer Science*, *159*, 630–639.

Chao, M. A., Kulkarni, C., Goebel, K., & Fink, O. (2022). Fusing physics-based and deep learning models for prognostics. *Reliability Engineering & System Safety*, *217*.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *Proceedings of the International Conference on Machine Learning*.

Deng, W., Nguyen, K. T., Gogu, C., Morio, J., & Medjaher, K. (2022). Physics-informed lightweight temporal convolution networks for fault prognostics associated to bearing stiffness degradation. *Proceedings of the PHM Society European Conference*.

Ding, Y., Zhuang, J., Ding, P., & Jia, M. (2022). Self-supervised pretraining via contrast learning for intelligent incipient fault detection of bearings. *Reliability Engineering & System Safety*, *218*.

Gebru, T., Hoffman, J., & Fei-Fei, L. (2017). Fine-grained recognition in the wild: A multi-task domain adaptation approach. *Proceedings of the IEEE International Conference on Computer Vision*.

Hagmeyer, S., Zeiler, P., & Huber, M. F. (2022). On the integration of fundamental knowledge about degradation processes into data-driven diagnostics and prognostics using theory-guided data science. *Proceedings of the PHM Society European Conference*.

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. d., Gutierrez, C., ... others (2021). Knowledge graphs. *ACM Computing Surveys*, *54*(4), 1–37.

Jadhav, V., Deodhar, A., Gupta, A., & Runkana, V. (2022). Physics informed neural network for health monitoring of an air preheater. *Proceedings of the PHM Society European Conference*.

Jain, N., Kalo, J.-C., Balke, W.-T., & Krestel, R. (2021). Do embeddings actually capture knowledge graph semantics? *Proceedings of the European Semantic Web Conference*.

Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., & Makedon, F. (2020). A survey on contrastive self-supervised learning. *Technologies*, *9*(1).

Jayathilaka, M., Mu, T., & Sattler, U. (2021). Ontology-based n-ball concept embeddings informing few-shot image classification. *Proceedings of the International Conference on Machine Learning and Applications*.

Jing, L., Vincent, P., LeCun, Y., & Tian, Y. (2022). Understanding dimensional collapse in contrastive self-supervised learning. *Proceedings of the International Conference on Learning Representations*.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., ... Krishnan, D. (2020). Supervised contrastive learning. *Proceedings of the Conferences on Neural Information Processing Systems*.

Monka, S., Halilaj, L., & Rettinger, A. (2022). A survey on visual transfer learning using knowledge graphs. *Semantic Web*, *13*(3), 477–510.

Monka, S., Halilaj, L., Schmid, S., & Rettinger, A. (2021). Learning visual models using a knowledge graph as a trainer. *Proceedings of the International Semantic Web Conference*.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Peng, D., Liu, C., & Gryllias, K. (2022). A transfer learning-based rolling bearing fault diagnosis across machines. *Annual Conference of the PHM Society*.

Rahat, M., Mashhadi, P. S., Nowaczyk, S., Rognvaldsson, T., Taheri, A., & Abbasi, A. (2022). Domain adaptation in predicting turbocharger failures using vehicle's sensor measurements. *Proceedings of the PHM Society European Conference*.

Rombach, K., Michau, G., & Fink, O. (2021). Contrastive learning for fault detection and diagnostics in the context of changing operating conditions and novel fault types. *Sensors*, *21*(10).

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Xia, L., Zheng, P., Li, X., Gao, R. X., & Wang, L. (2022). Toward cognitive predictive maintenance: A survey of graph-based approaches. *Journal of Manufacturing Systems*, *64*, 107–120.

Zheng, H., Yang, Y., Yin, J., Li, Y., Wang, R., & Xu, M. (2020). Deep domain generalization combining a priori diagnosis knowledge toward cross-domain fault diagnosis of rolling bearing. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–11.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43–76.

BIOGRAPHIES

Maximilian-Peter Radtke studied business mathematics at the University of Mannheim, Germany and graduated in 2018. After his studies he worked as a data science consultant in various industries for two and a half years before returning to academia. Since 2021 he is part of AIMotion Bavaria at the Technische Hochschule Ingolstadt and the research group AI applications for innovative production and logistic systems. His research interests include the combination of symbolic and sub-symbolic AI approaches and the incorporation of knowledge into deep learning in the area of fault diagnostics and prognostics.

Marco Huber received his diploma, Ph.D., and habilitation degrees in computer science from the Karlsruhe Institute of Technology (KIT), Germany, in 2006, 2009, and 2015, respectively. From June 2009 to May 2011, he was leading the research group Variable Image Acquisition and Processing of the Fraunhofer IOSB, Karlsruhe, Germany. Subsequently, he was Senior Researcher with AGT International, Darmstadt, Germany, until March 2015. From April 2015 to September 2018, he was responsible for product development and data science services of the Katana division at USU Software AG, Karlsruhe, Germany. At the same time he was adjunct professor of computer science with the KIT. Since October 2018 he is full professor with the University of Stuttgart. He further is director of the Department Cyber Cognitive Intelligence (CCI) and of the Department Machine Vision and Signal Processing with Fraunhofer IPA in Stuttgart, Germany. His research interests include machine learning, planning and decision making, machine vision, and robotics.

Jürgen Bock is a computer scientist, who graduated as Diplom-Informatiker from Ulm University, Germany, and as Bachelor of Information Technology with Honours from Griffith University, Brisbane, Australia, in 2006. He began his research career at the FZI Research Center for Information Technology in Karlsruhe, Germany, and received his PhD from the Karlsruhe Institut of Technology (KIT) in 2012. After 2 years as post doc and team leader at the FZI, he joined the corporate research department of KUKA Robotics in Augsburg, Germany as developer and later leader of the team Smart Data and Infrastructure. In 2020 he joined the Technische Hochschule Ingolstadt (THI) as research professor in the area of AI applications in innovative production and logistics systems.