Journal of Manufacturing Systems 71 (2023) 274–285

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

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

Maximizing model generalization for machine condition monitoring with Self-Supervised Learning and Federated Learning

Matthew Russell [a], Peng Wang [a,b,*]

[a] Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY, USA
[b] Department of Mechanical and Aerospace Engineering, University of Kentucky, Lexington, KY, USA

ARTICLE INFO

Keywords:
Condition monitoring
Self-supervised learning
Federated learning
Fault diagnosis
Transfer learning
Emerging faults

ABSTRACT

Deep Learning (DL) can diagnose faults and assess machine health from raw condition monitoring data without manually designed statistical features. However, practical manufacturing applications require robust and repeatable solutions that can be trusted in dynamic environments.

Machine data is often unlabeled and from very few health conditions (e.g., only normal operating data). Furthermore, models often encounter shifts in domain as process parameters change and new categories of faults emerge. Traditional supervised learning may struggle to learn compact, discriminative representations that generalize to these unseen target domains since it depends on having plentiful classes to partition the feature space with decision boundaries. Transfer Learning (TL) with domain adaptation attempts to adapt these models to unlabeled target domains but assumes similar underlying structure that may not be present if new faults emerge. This study proposes focusing on maximizing the feature generality on the source domain and applying TL via weight transfer to copy the model to the target domain. Specifically, Self-Supervised Learning (SSL) with Barlow Twins may produce more discriminative features for monitoring health condition than supervised learning by focusing on semantic properties of the data. Furthermore, Federated Learning (FL) for distributed training may also improve generalization by efficiently expanding the effective size and diversity of training data by sharing information across multiple client machines. Results show that Barlow Twins outperforms supervised learning in an unlabeled target domain with emerging motor faults when the source training data contains very few distinct categories. Incorporating FL may also provide a slight advantage by diffusing knowledge of health conditions between machines. Future work should continue investigating SSL and FL performance in these realistic manufacturing scenarios.

1. Introduction

Smart factories need to detect and diagnose machine faults to prevent costly downtime and repairs. To this end, machine learning can build classification and regression models for condition monitoring and fault diagnosis using statistical patterns discovered in large data sets. Deep Learning (DL) has shifted the paradigm away from manually-designed features (e.g., mean, variance, kurtosis, peak values, etc.) by introducing efficient algorithms for training neural networks with many layers to extract features automatically from raw data (e.g., vibration signals) [1,2]. However, practical manufacturing applications require robust and repeatable solutions that can be trusted in dynamic environments. Since DL models derive their behavior from empirical training data, predictions can be difficult to verify and validate, and large quantities of clean examples are not available covering all possible operating conditions and process parameters. Building such exhaustive training sets is prohibitively time-consuming and cost-intensive—not to mention the privacy concerns if data comes from many different sources. Widespread use of DL for condition monitoring hinges on finding effective alternatives that promote trust and repeatability.

Interest in pursuing trustworthy DL stems from early DL work in manufacturing that established its superiority over traditional approaches like Support Vector Machine (SVM) for analyzing condition monitoring data sets [3]. Despite excellent results on controlled laboratory data sets, many practical considerations hinder widespread adoption of DL within manufacturing. Contrary to image domains that have millions of images from hundreds or thousands of categories [1], fault diagnosis problems often lack the volume and diversity of data required to learn robust feature extraction networks that generalize beyond a single data set, operating condition, or machine [4]. Healthy operating conditions dominate real-world industrial data sets with very

* Corresponding author at: Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY, USA.
E-mail address: edward.wang@uky.edu (P. Wang).

https://doi.org/10.1016/j.jmsy.2023.09.008
Received 27 April 2023; Received in revised form 12 September 2023; Accepted 14 September 2023
Available online 28 September 2023

few – if any – examples of faults [5]. Any limited examples of faults will be overwhelmingly unlabeled. Furthermore, factory environments are dynamic; new types of faults can occur without warning and be confidently misclassified by an outdated model [6–8]. Thus, increased operational trust starts with improving generalization to ensure models behave more predictably when confronted with uncertain process dynamics and incomplete observational knowledge.

Transfer Learning (TL) can alleviate some issues with generalization. TL seeks to repurpose and reuse a model when faced with changing data or tasks (e.g., new faults or process parameters) [9]. These changes affect the statistical properties of the data, shifting it out of the model's valid input domain [10]. TL for domain adaptation transfers a model from a labeled source domain to an unlabeled target domain. However, emerging faults in the target domain may hinder the ability to transfer the source domain model. Additionally, the target domain itself could be unknown or represent a future operating state with no data – even unlabeled data – available at training time. In this case, TL approaches must learn the most generalizable representation possible from the available data. The model can then be transferred to the target domain and used as-is or fine tuned as target domain data becomes available [9]. This technique can bootstrap models for the target domain without assuming an isomorphic relationship to the source domain conditions.

Bootstrapping source models with supervised learning (i.e., labeled data) may be ineffective in practical condition monitoring since few training conditions are available, and labels are often missing. Self-Supervised Learning (SSL) may be more appropriate. SSL techniques create compact clusters of features with similar semantic characteristics [11]. Random augmentations (e.g., random scale, time shift, etc.) implicitly specify what variation the model should expect within a category of signals. For example, if both a randomly flipped signal and the original should map to the same feature, the model learns to ignore flipping. Requiring no labels, SSL facilitates learning data-centric representations from raw, unannotated factory data.

While SSL may better bootstrap condition monitoring models, generalization can be improved further by sharing information among a fleet of machines. Bandwidth constraints may prevent the fleet from continually aggregating data in the cloud, but Federated Learning (FL) can utilize the distributed data efficiently to develop a globally-informed model [12]. Each client machine trains on locally observed data and periodically transmits its model – not the raw data – to a server which combines the updates into a single model. This global model is then distributed to the clients, diffusing information among them. Thus, FL can expand the effective size and diversity data sets by integrating information from multiple clients without inundating communication networks.

Condition monitoring literature lacks a cohesive introduction to SSL and FL for maximizing model generalization. This study outlines how SSL and FL can improve the generalization – and therefore trustworthiness – of DL models on the factory floor via two complementary strategies: SSL extracts informative representations without needing labeled data, and FL expands the effective size and diversity of the data set. Pursuing generalizable models through SSL and FL allows manufacturers to adopt a knowledge-informed approach and securely share information via FL among clients grouped by expert knowledge while simultaneously maximizing the utilization of massively unlabeled data via SSL. The contributions of this study can be summarized as follows:

1. an overview of SSL and related work in manufacturing,
2. an overview of FL and related work in manufacturing,
3. a theoretically motivated framework for combining SSL and FL to improve model generalization, and
4. a case study assessing SSL and FL under emerging faults and changing process parameters using a motor fault data set.

The rest of this paper is organized as follows: Section 2 outlines the theoretical background and related work, Section 3 describes the proposed SSL and FL methods for condition monitoring, Section 4 introduces a motor health condition case study, Section 5 presents and discusses the results, and Section 6 provides concluding remarks.

2. Theoretical background and related work

This work builds on Transfer Learning, Self-Supervised Learning, and Federated Learning.

2.1. Supervised learning and transfer learning

Many factors can limit the applicability and robustness of machine learning models. In manufacturing, changing processing parameters, operating environments, and health conditions can negatively impact performance by shifting the input data distribution outside the expected domain. Transfer Learning (TL) seeks to adapt or reuse models trained in a source domain to a related target domain [9], circumventing the need for large volumes of labeled data for the target task.

2.1.1. Supervised learning

A typical fault diagnosis model can be split into a feature extraction backbone G_θ parameterized by weights θ and classification head F_ϕ with weights ϕ that predicts the probabilities of K classes (e.g., faults) from the extracted features. With labeled data, the model parameters can be optimized with stochastic gradient descent and backpropagation using the cross-entropy loss (i.e., cost) function:

$$\mathcal{L}_{\text{CE}}(X, Y) = -\frac{1}{n} Y^\top \log F_\phi(G_\theta(X)) \tag{1}$$

where $X = [\mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_n]$ is a batch of n input examples, $Y = [\mathbf{y}_1 \mathbf{y}_2 \dots \mathbf{y}_n]$ is corresponding binary label vectors $\mathbf{y} \in \{0, 1\}^K$ with 1 in the index corresponding to the true label and zeros elsewhere, and $F_\phi(G_\theta(X)) = [\hat{\mathbf{y}}_1 \hat{\mathbf{y}}_2 \dots \hat{\mathbf{y}}_n]$ is the set of predicted class probabilities for the batch. Optimizing the weights to maximize classification accuracy teaches the model to draw "decision boundaries" that separate the features from different categories. However, changes in process parameters or operating environment shift the distribution of input data and features from G_θ. These new features no longer align with the decision boundaries learned by the classifier F_ϕ, producing undefined or inconsistent behavior. This damages the generalization of supervised classifiers.

2.1.2. Transfer learning via domain adaptation

Transfer Learning (TL) is one solution to the domain shift problem. For domain adaptation, unlabeled data from a known target domain can regularize the supervised training process so G_θ produces stable, matching distributions of source and target domain features for the classifier F_ϕ. An updated loss function that includes unlabeled target domain data is used during training:

$$\mathcal{L}_{\text{DA}}(X_s, Y_s, X_t) = \mathcal{L}_{\text{CE}}(X_s, Y_s) + \lambda D(G_\theta(X_s), G_\theta(X_t)) \tag{2}$$

where X_s is the batch of source domain inputs, Y_s is the batch of source domain labels, X_t is the batch of unlabeled target domain inputs, and $D(\cdot, \cdot)$ is a function measuring the distribution discrepancy between source domain features $G_\theta(X_s)$ and target domain features $G_\theta(X_t)$ [13]. The λ factor controls the strength of feature regularization. Since the feature extractor G_θ produces a consistent distribution of features from both the source and target domains, the fault classifier F_ϕ is more likely to generate accurate predictions for the target domain.

A popular implementation of $D(\cdot, \cdot)$ in manufacturing is Maximum Mean Discrepancy (MMD). Using MMD to ensure similarity between source and target features, [14] demonstrated TL of bearing and gearbox vibration data across different loads and shaft speeds. With flexible kernel implementations, MMD can be combined with a polynomial or Cauchy kernel as shown on laboratory fault data sets [15,16]. Applying

MMD at multiple levels in a deep feature extractor can also provide performance gains for lab-to-real transfer for locomotive bearing fault diagnosis and classification and localization of bearing faults [17,18].

Rather than relying on an explicit metric, another widely used approach is Domain Adversarial Neural Network (DANN) which replaces the $D(\cdot,\cdot)$ loss term with another neural network D_ψ that learns to discriminate source and target features [19]. By training the feature extractor G_θ to confuse the domain discriminator D_ψ, the feature extractor learns to generate matching features for source and target domain data. DANN can facilitate TL across different bearing data sets with a 1D CNN feature extractor [20]. Interestingly, combining both MMD and DANN may be beneficial and has also been demonstrated for TL across data sets [21].

2.1.3. Transfer learning via weight transfer

Domain adaptation may encounter difficulties when new faults emerge. If the target domain contains emerging faults, encouraging source and target features to match may be detrimental. Furthermore, the classifier itself must be reconfigured to detect the additional fault(s). Thus, instead of domain adaptation, TL under emerging faults shifts to maximizing the generalization of feature representations learned on the source domain. If the representation is general enough, network weights can be transferred to the target domain to separate emerging faults and previously known faults. That is, given labeled source and/or unlabeled target domain data, TL via weight transfer seeks to pretrain a representation that remains discriminative for future emerging faults. In image processing, weight transfer allows applications to reuse low-level, general features learned by networks trained on massive image data sets [22]. The size and diversity of the training data enables these pretrained networks to produce highly discriminative features for emerging categories of images. Starting from these pretrained weights can produce useful feature representations for solving problems in domains like medical imaging where data is too scarce to train reliable image classifiers from scratch [23].

Manufacturing researchers have leveraged these pretrained image networks creatively by transforming condition monitoring data sets into images. While the high-level tasks differ, pretrained networks extract useful low-level information about lines and shapes in the images [1]. If vibration data is transformed into 2D images via the Continuous Wavelet Transform (CWT), these pretrained image networks can provide out-of-the-box features for training fault classifiers when labeled manufacturing data is limited [24,25]. They can even accelerate domain adaptation by providing the initial feature representation before applying a technique like MMD [13]. Outside of pretrained image networks, [26] demonstrated that TL via weight transfer can improve predictions of a target aircraft engine's degradation by training a degradation model on source engines, transferring the weights to the target engine, and then fine-tuning on the target's first few degradation steps. However, in many cases TL via weight transfer remains difficult for manufacturing because of the lack of labeled data required to pretrain highly general feature extractors.

2.2. Self-supervised learning

Self-Supervised Learning (SSL) uses unlabeled data to train feature extraction networks that can be transferred to downstream tasks. Broadly speaking, SSL lets the data "supervise itself" through pretext tasks or invariance-based methods to learn a useful encoding of the input examples. SSL could be transformational in manufacturing where labeled data is scarce and unlabeled data is plentiful.

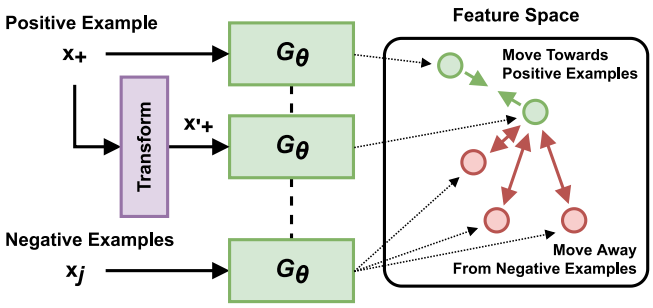


Fig. 1. SSL techniques seek to move augmented features towards members of the same pseudoclass while increasing separation from other pseudoclasses.

2.2.1. Pretext task SSL

Pretext task SSL trains models on a related problem using auto-generated labels. Examples of pretext tasks include predicting image rotations [27], the relative position of patches within an image [28], or the next word in a natural language sequence (e.g., GPT-*n* models from OpenAI) [29]. Manufacturing and health monitoring research has explored various adaptations of this approach. Some studies re-brand traditional unsupervised techniques as "self-supervised". For example, an embedding learned from only normal data via Kernel Principal Component Analysis (PCA) helped detect faults in an industrial metal etching process and was described as self-supervised [30]. Similarly, [31] trained a deep autoencoder as a "self-supervised" auxiliary task for bearing fault classification, while [32] adopted a similar approach for anomaly detection in washing machines. Work by [33] predicted the orientation of randomly rotated laser powder bed fusion process images from additive manufacturing and characterized this as a pretext task. However, since the downstream task was also orientation prediction, this resembles pretraining with data augmentation rather than a distinct pretext goal. True pretext task SSL for downstream fault diagnosis mines features from unlabeled data via distinct pretraining tasks that do not depend on fault information. For example, a model could learn useful features by predicting statistical properties of unlabeled input signals (e.g., mean, variance, skew, and kurtosis) [34]. Both [35,36] randomly distorted input signals and trained a model to identify the applied distortion. All three approaches produced features useful for bearing fault diagnosis. Thus, without requiring manual labels, pretext task SSL can bootstrap models for future health monitoring tasks.

2.2.2. Invariance-based SSL

Instead of using pretext tasks, invariance-based SSL applies random transformations to a "seed" example from the data set, creating family of examples belonging to the same "pseudoclass". The feature extraction network is then trained to homogenize features from all augmented examples in the pseudoclass [11]. A contrastive loss function encourages each pseudoclass to be both compact and well-separated from others [37]. Through this process, the network learns to ignore the randomized attributes and focus on semantically meaningful ways to cluster the inputs data (see Fig. 1).

Contrastive approaches to Invariance-based SSL depend on having plentiful "negative" examples of other pseudoclasses to ensure good clustering. For example, consider the InfoNCE loss function, where $\mathbf{x}'_+$ is an augmented version (i.e., same pseudoclass) of a positive reference example $\mathbf{x}_+$ [38,39]:

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{s(G_\theta(\mathbf{x}'_+), G_\theta(\mathbf{x}_+))}{\sum_{j=1}^{n} s(G_\theta(\mathbf{x}'_+), G_\theta(\mathbf{x}_j))} \tag{3}$$

where n is the size of the batch that includes a positive example $\mathbf{x}_+$ and $n-1$ negative examples (i.e., other pseudoclasses), and $s(\cdot,\cdot)$ is a similarity metric. Increasing the number of negative examples increases

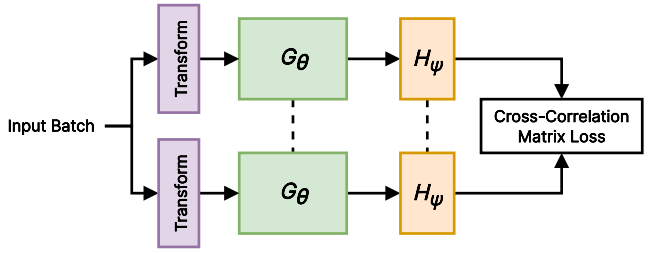


Fig. 2. Barlow Twins encourages feature projections to be correlated within each pseudoclass and independent from each other to reduce redundancy in the representation.

the lower bound on the mutual information (similarity) between the features of the positive sample $G_\theta(\mathbf{x}_+)$ and those of its augmentation $G_\theta(\mathbf{x}'_+)$ [38]. This will encourage compact feature clusters. However, efficiently training with enough negative examples can be nontrivial since the batch size is limited [40]. Momentum Contrast (MoCo) [39] increased the number of negative examples by accumulating features across multiple batches. The encoder trained with contrastive loss to separate the current batch from this larger group of negative example features. A "momentum encoder" embedded the previous examples into the latent space was updated through a running average to ensure the representations of negative examples from multiple previous batches remained stable.

MoCo prompted many conceptually related developments. A Simple Framework for Contrastive Learning of Visual Representations (SimCLR) [41] and Bootstrap Your Own Latent (BYOL) [42] both proposed modifications of the MoCo-style architecture that could perform well with fewer or no negative examples. SimCLR made the important contribution of a "projection head" network that mapped features to a larger-dimension space before applying contrastive loss, protecting the features themselves from being too aggressively homogenized. Work by [43] proposed an even more straightforward approach known as Simple Siamese Representation Learning (SimSiam). SimSiam learned to consolidate feature projections from two augmentations while preventing gradients from one of the projections from updating the encoder. This effectively held one projection stationary while moving the other towards this anchor. This proved effective even without large batches, plentiful negative examples, or momentum networks. Bypassing issues with contrastive loss altogether, Barlow Twins used a cross-correlation loss that learned correlated features among pseudoclass examples while discouraging redundancy among the feature dimensions (see Fig. 2) [44]. Subsequently, Variance-Invariance-Covariance Regularization (VICReg) introduced a generalization of Barlow Twins with a slightly more complex loss function [45]. These methods proved increasingly useful for computer vision problems.

Within manufacturing, invariance-based SSL from computer vision can be leveraged by first converting 1D sensing data into 2D images. With 2D images of unlabeled vibration data, SimCLR can find discriminative fault features for rotating machinery using image augmentations like rotations, crops, and affine transforms [46]. Utilizing BYOL, [47] extracted bearing fault features after converting vibration data to images with methods including Short-Time Fourier Transform (STFT) and Continuous Wavelet Transform (CWT). However, applying image domain techniques to vibration data might lack a robust, physically meaningful interpretation. Therefore, an important step for adapting invariance-based SSL to condition monitoring is designing appropriate random augmentations for raw time series data (e.g., vibration and electrical current) that guide training towards features with rich fault information.

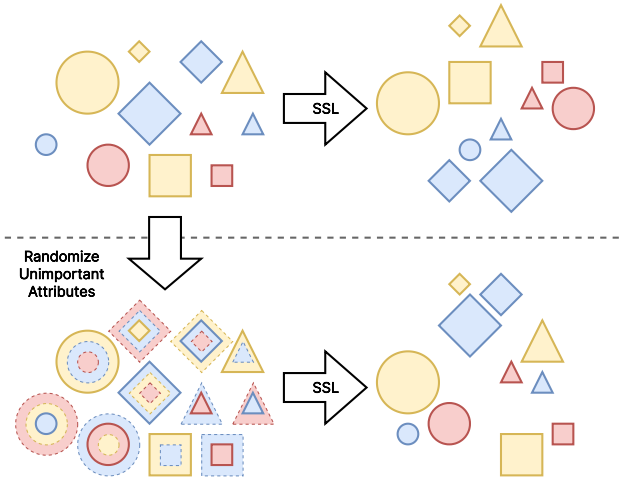


Fig. 3. By randomizing semantically meaningless attributes, augmentations force SSL to identify pseudoclasses through the remaining, semantically meaningful characteristics. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.2.3. Designing time series data augmentations

The random augmentations used by invariance-based SSL must be carefully selected to avoid destroying important semantic information. Input semantics often emerge from complex underlying relationships; high-level, semantic labels (e.g., bearing inner race fault) cannot be reduced to a simple feature analysis (e.g., normalized vibration amplitude exceeding 0.6) nor should this be expected. The difficulty in uncovering these nonobvious correlations motivates the use of DL. Therefore, if an input attribute is semantically meaningful, extracting and manipulating the attribute tends to be very difficult (e.g., algorithmically transforming vibrations from a bearing inner race fault to a healthy vibration signal). The contrapositive is also true: if an attribute is not difficult to manipulate, it will likely not be semantically meaningful (to an extent). Thus, effective random augmentations need not be complex to homogenize representations of semantically-related examples (see Fig. 3). Existing augmentation-based SSL work with images supports this theory by using simple transforms like translation, crop, flip, rotation, contrast, blur, and color distortion for state-of-the-art results [43–45]. Each domain is different [48], and designing equivalent augmentations for 1D time series data unlocks the potential of invariance-based SSL for raw sensing signals.

Several studies have explored possible time series augmentations. Since time series examples are related temporally (unlike images), [49] generated pseudoclasses for invariance-based SSL from pairs of consecutive instances from the vibration signal in addition to time and amplitude distortions of single instances. Gaussian noise, amplitude scaling, stretching, masking, and time shifting were used with MoCo to pretrain a feature extractor for detect incipient faults in bearing histories [50]. Adopting BYOL, [51] used truncation (i.e., masking a contiguous region), lowpass filtering, Gaussian noise, geometric scaling, and downsampling to learn representations from raw, unlabeled vibration data for bearing fault diagnosis. Results indicated that truncation and downsampling were particularly useful. A similar study utilizing SimSiam was conducted by [52] with truncation, lowpass filtering, Gaussian noise, and time reversal. Using a motor condition data set, [53] implemented Barlow Twins on multichannel vibration and current signals with random time shifting, truncation, scaling, and vertical flipping. The random time shift was crucial for extracting good features for the motor fault diagnosis task. These studies demonstrate effective data augmentations when applying invariance-based SSL to 1D signals.

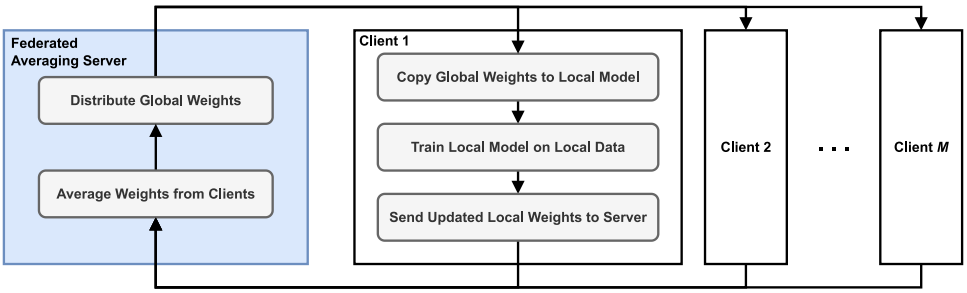


Fig. 4. Overview of Federated Learning using FedAvg.

Algorithm 1: The FedAvg FL algorithm [12]

Input : Number of rounds N; number of clients M; client steps per round n;
Output: Trained global model weights w^N
$w^0 \leftarrow$ Random initial model weights;
for $i \leftarrow 1$ **to** N **do**
  for $j \leftarrow 1$ **to** M **do**
    $w_j^i \leftarrow w^{i-1}$; // Copy global to client
    $a_j \leftarrow 0$;
    for $k \leftarrow 1$ **to** n **do**
      // Train one minibatch on client
      $\mathcal{M} \leftarrow$ sample minibatch;
      $\mathcal{L} \leftarrow \text{ComputeLoss}(\mathcal{M}; w_j^i)$ $w_j^i \leftarrow w_j^i - \eta \nabla \mathcal{L}$;
      $a_j \leftarrow a_j + |M|$;
    end
  end
  $w^i \leftarrow \frac{1}{\sum_{l=1}^{M} a_l} \sum_{j=1}^{M} a_j w_j^i$; // Update global
end

2.3. Federated learning

Federated Learning (FL) facilitates distributed training of predictive deep learning models on private user data via the FedAvg algorithm [12]. To maintain user privacy, network training is performed on the user's device—only the updated model weights and parameters are sent to the cloud. In the FedAvg algorithm, the network weights are averaged together to create the global model without needing to send any client data to the cloud. This allows clients to retain private control over their data while still collaborating to train a more generalizable model. Algorithm 1 outlines FedAvg, starting with a randomized global model w^0 and performing N rounds of federation. The global model for round $i = 1, 2, \ldots, N$ is distributed to M clients who update the model using n local minibatches of data. Each client $j = 1, 2, \ldots, M$ then transmits the updated model w_j^i for round i back to the server. Each client also transmits total amount of training data a_j used by client j. Once all the updates are received for round i, the server computes the global model via weighted average:

$$w^i = \frac{1}{\sum_{l=1}^{M} a_l} \sum_{j=1}^{M} a_j w_j^i \tag{4}$$

The weighting coefficients a_j ensure that the global update is biased towards client models that trained on more data, which are likely to produce a more stable step than models trained on only a few examples. The global model is then redistributed to all the clients for the next round of FL (see Fig. 4).

2.3.1. FL for condition monitoring and fault diagnosis

An immediately apparent benefit of FL for manufacturing is the ability to train on multiple data sets without exposing sensitive factory data to the server. Motivated by this privacy perspective, [54] proposed FL for building a fault diagnosis model from isolated data sets, although the method assumes all clients see matching faults. Client models with low validation performance are ignored when aggregating the global model to improve robustness. A peer-to-peer adaptation of FL showed improvements over local training at each node for detecting wind turbine and bearing faults [55]. [56] also investigated FL for bearing fault diagnosis while proposing a vertical FL algorithm based on gradient tree boosting to accommodate clients with different feature subsets. For Remaining Useful Life (RUL) applications, [57] implemented FL for collaborative training of transformer models on degradation data from simulated turbofan aircraft engines.

2.3.2. Multi-party and single-party incentives for FL

Beyond privacy, FL offers benefits to both coalitions of multiple manufacturers and within a single, distributed manufacturer. In additive manufacturing, [58] found that FL improves defect image segmentation over locally trained client models and showed that performance gains can both incentivize manufacturers to join existing federations and incentivize these federations to welcome new clients. Work by [59] further supports FL's ability to improve model performance versus locally trained models while preserving privacy among aircraft manufacturers. Even if manufacturers decline federations with competitors to avoid possible model poisoning [60], FL offers substantial benefits for communication-efficient training on distributed data owned by a single manufacturing entity, reducing the network traffic needed to maximize utilization of distributed sensing. However, in both the multi-party and single-party paradigms, FL implementations must handle discrepancies between clients while still maximally leveraging a collaborative approach.

2.3.3. FL for heterogeneous clients

In practical applications, clients could have different tasks or distributions of data, making basic FedAvg suboptimal for each member but still desirable for privacy benefits. Initializing FL clients with a pretrained global feature extractor can reduce the required training time on individualized downstream tasks [61]. However, the case studies only tested this for image domain tasks. Similarly, a personalized FL approach can locally optimize feature extractors and classifiers while penalizing shifts between the local classifier weights and the globally optimized weights [62]. This permits the clients to share information without a hard constraint that fixes weights among them. Surprisingly, if the clients observe different faults, [63] demonstrated that FL can share classifier information across rotating machinery clients even if they have unbalanced or non-i.i.d classes. Injecting noise and creating fake pseudoclasses within each client can also help with globally aligning classes between models [64]. Conversely, if the client input distributions differ significantly, a single global model might not be successful. [65] opted to cluster gradient updates from members and perform FL separately within each subgroup. Experiments validated the algorithm on benchmark data and a custom bearing fault data set. However, these studies in heterogeneous FL critically stop short

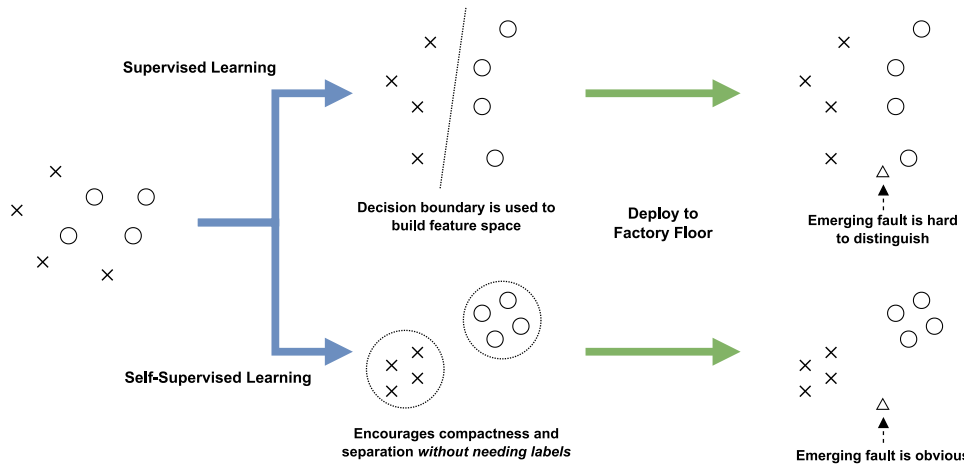


Fig. 5. SSL encourages compactness and separation of pseudoclasses while supervised representations are dependent on decision boundaries.

of addressing the problem of massively unlabeled data at each client. Furthermore, when the number of observed classes is extremely limited, relying supervised learning could hinder the discriminativeness of learn representations.

3. Proposed methods for maximizing model generalization

Although supervised learning on massively diverse data sets may produce generalizable features [66], it may struggle when class (i.e., fault/condition) diversity is limited in two ways by (1) producing less compact clusters, and (2) allowing noise or systematic biases to dominant feature extraction. A simple classification objective constructs the feature space and decision boundaries without explicitly encouraging compact clusters (see Fig. 5). With limited training classes, the model has few decision boundaries with which to partition the feature space. This could produce loosely structured features, increasing the likelihood that features from future emerging faults will overlap those from previous health conditions. While adding compactness objectives may help, fewer classes also means the model has fewer observations from varied environmental conditions and process parameters. Since DL implementations are free to learn features themselves, a supervised model could resort to systematic biases to separate data rather than the more complex underlying fault signals as intended. Combining data from distributed machines could mitigate these issues by increasing class diversity, but aggregating high-velocity sensing streams could be difficult given bandwidth constraints. Furthermore, most raw data will be unlabeled regardless, making large-scale supervised learning impossible. The proposed method instead adopts SSL to support unlabeled data and improve the feature space structure and FL to expand the effective data set size without inundating communication networks or introducing privacy concerns (see Fig. 6). Together, these techniques learn a more discriminative feature space that generalizes to new operating conditions and emerging faults.

3.1. Barlow twins

Replacing supervised learning with SSL introduces the knowledge-informed assumption that although emerging faults or new operating conditions have not been observed, this time series data from the target domain will have similar building blocks and salient characteristics – e.g., frequency content – that discriminates them. To extract these salient indicators instead of unwanted biases, SSL relies on expert-designed random data augmentations that indicate the expected variation with the signals. Barlow Twins SSL seeks to tightly cluster feature projections from different augmentations of the same observation by maximizing the cross-correlation between projections. This ensures that examples falling within the expected signal variation are grouped closely together. The augmentations themselves should be informed by knowledge of condition monitoring signals to randomize unimportant signal attributes while preserving the semantic class [51]. Extending the proposed augmentations from [53], Algorithm 2 outlines the random transformations used with Barlow Twins in the proposed methods for condition monitoring. The examples are randomly shifted (jittered) in time, scaled, and masked. Given input batch X of n examples, feature extraction backbone G_θ, projector H_ψ, Barlow Twins first computes the projections of two augmented versions X' and X'' of the input batch (according to Algorithm 2) and their corresponding projections $Z' = H_\psi(G_\theta(X'))$ and $Z'' = H_\psi(G_\theta(X''))$. Then both sets of projections

Algorithm 2: Random augmentations for Barlow Twins in PyTorch style

```
# x: 1D input tensor with shape (B, C, L)
def randomly_augment(x):
  # Random jitter
  jitter = random.randrange(x.shape[-1])
  x = torch.cat(
    (x[:, :, jitter:], x[:, :, :jitter]),
    dim=-1,
  )
  # Random scale
  vmax = (
    x.abs()
    .reshape(x.shape[0], -1)
    .max(dim=-1, keepdim=True)[0]
  )
  max_scale = vmax.reciprocal()
  min_scale = 0.1
  scales = (
    torch.rand_like(max_scale)
    * (max_scale - min_scale)
    + min_scale
  )
  x = x * scales.unsqueeze(-1)
  # Random mask
  mask_size = 64
  mask_start = random.randrange(
    x.shape[-1] - mask_size
  )
  mask_end = mask_start + mask_size
  x[:, :, mask_start:mask_end] = 0.0
return x
```

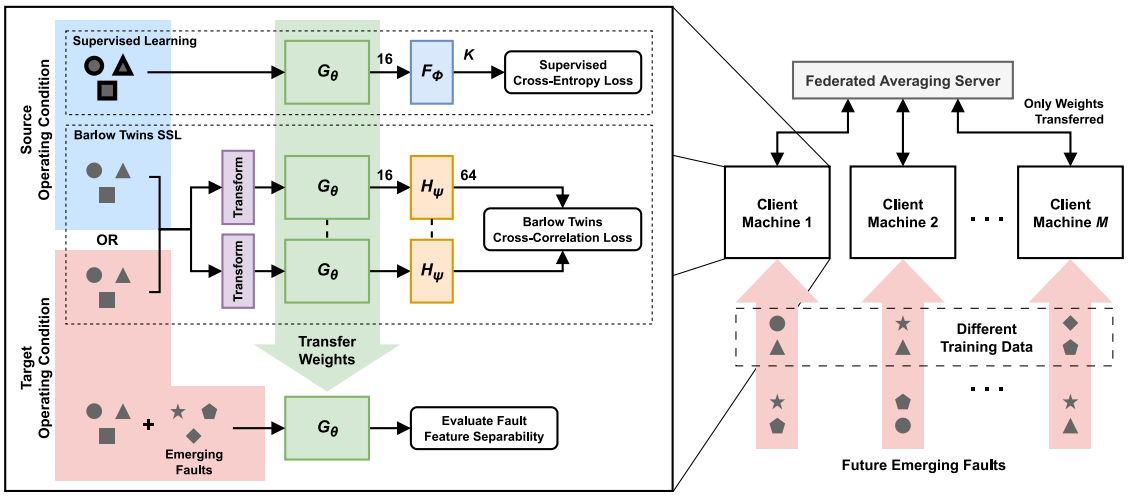


Fig. 6. Proposed methods for comparing the discriminability of emerging faults when transferring weights from a supervised or self-supervised 1D CNN feature extraction backbone. Federated Learning can then be used to share information efficiently among multiple client machines.

are normalized across the batch:

$$\mu_i = \frac{1}{n}\sum_{k=1}^{n} Z_{ik}$$
$$\sigma_i^2 = \frac{1}{n}\sum_{k=1}^{n} (Z_{ik} - \mu_i)^2 \tag{5}$$
$$\hat{Z}_{ij} = (Z_{ij} - \mu_i)/\sigma_i$$

Next, the cross-correlation matrix R is computed and normalized by the batch size:

$$R = \hat{Z}'\hat{Z}''^{\top}/n \tag{6}$$

Finally, the loss function can be calculated using R:

$$\mathcal{L}_{\mathrm{BT}}(R) = \mathrm{tr}\left((R-I)^2\right) + \lambda \sum_i \sum_{j\neq i} R_{ji} \tag{7}$$

where λ controls the strength of the independence constraint. The first term encourages the diagonal elements to be one, meaning that individual features are highly correlated (aligned) across the batch, meaning that instances within the expected variation—as defined by the applied random augmentations–will map to similar feature projections (i.e., cluster together). The second term drives off-diagonal elements to zero so each feature is independent from the rest. This improves the representational capacity by ensuring multiple features do not encode the same information. With this loss function, the Barlow Twins feature extractor and projection head can be trained with standard stochastic gradient descent and backpropagation methods. Fig. 7 shows the architecture of the 1D CNN backbone G_θ for extracting features from condition monitoring data and the Barlow Twins projection head H_ψ.

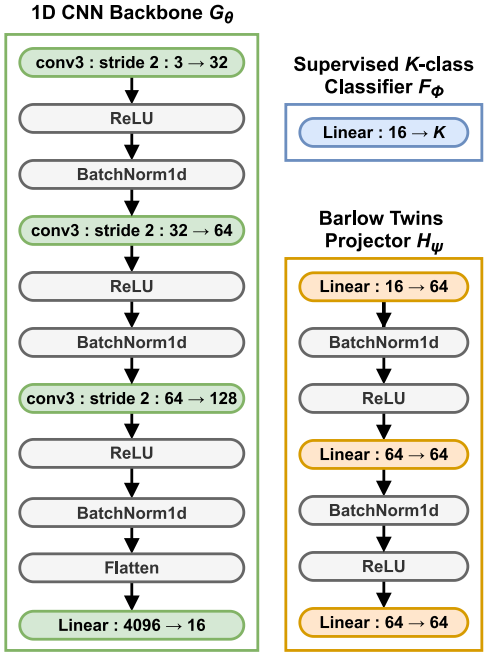


Fig. 7. The architectures for the 1D CNN backbone feature extractor G_θ, supervised K-class classifier F_ϕ, a and Barlow Twins projection head H_ψ.

3.2. Federated learning for information sharing

Most factory floors will have multiple similar machines that will each experience different health conditions throughout operation. Data from a single machine may contain very few distinct conditions, but network constraints may prevent each machine from streaming all its sensing data to the cloud to construct a unified data set. The machines themselves may not be geographically colocated or may belong to separate manufacturers without data-sharing agreements. To circumvent these hindrances, the model can be trained with `FedAvg` (see Algorithm 1). Each client machine retains complete ownership of its data while indirectly gaining knowledge about new health conditions through model averaging on the FL server. This indirect information sharing between clients via the global model can be viewed as a form of TL. When each client receives an updated global model, they benefit from the observations and knowledge of the other clients. Thus, even if a client lacks training experience with a given health condition, if another client *has* trained with that condition, the FL algorithm will diffuse this experience back to the uninformed client (see Fig. 8). Thus, FL may offer TL advantages among the clients, improving the generalization of each one to future fault conditions. Moreover, The client machines only send updated models to the FL server once per round, significantly reducing the volume and velocity of data transmitted to the cloud. By combining FL with SSL, DL can operate in realistic condition monitoring scenarios with unlabeled, distributed training data while reducing network communication and maintaining manufacturer privacy.

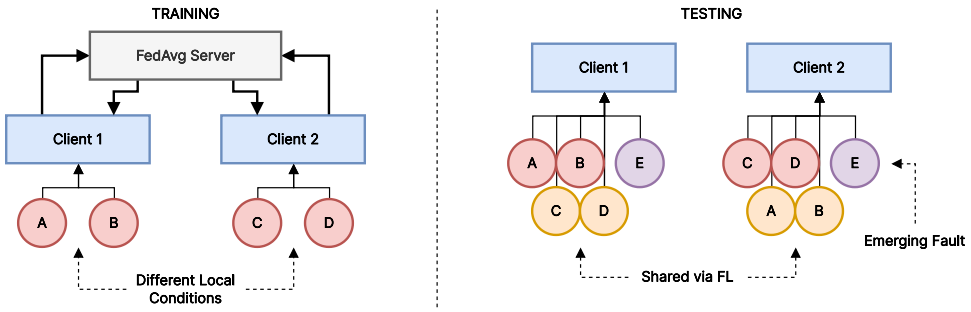


Fig. 8. Each client experiences different conditions, and averaging model weights diffuses this knowledge to other clients, maximizing the diversity of the data set and improving performance on emerging faults.

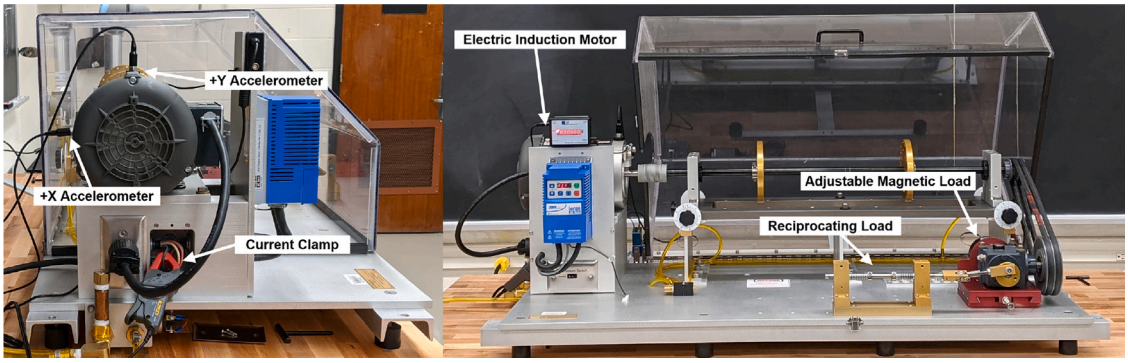


Fig. 9. The SpectraQuest Machinery Fault Simulator used to collect the motor health condition data set.

4. Experiments

Two case studies investigate the proposed claims. The first compares the generalizability of representations after pretraining with supervised learning or SSL on varying numbers of distinct classes. The second examines the impact of distributed training with FL on model performance under emerging faults.

4.1. Motor condition data set

Both case studies use a motor fault condition data set collected from the SpectraQuest Machinery Fault Simulator (MFS) in Fig. 9. With a 12 kHz sampling rate, two accelerometers mounted orthogonally capture vibration data, and a current clamp measures electrical current signals. Sixty seconds of steady-state data is gathered for eight motor health conditions: normal (N), faulted bearings (FB), bowed rotor (BoR), broken rotor (BrR), misaligned rotor (MR), unbalanced rotor (UR), phase loss (PL), and unbalanced voltage (UV). Each of the conditions is run at 2000 RPM and 3000 RPM with loads of 0.06 N m and 0.7 N m for a total of 32 unique combinations of health conditions and process parameters. For simplicity, each unique combination can be identified with xy where x is 2 or 3 to specify the RPM parameter, and y is "H" or "L" to specify a high or low load parameter (e.g., 3L refers to 3000 RPM with load of 0.06 N m). The signals are then normalized to $[-1, 1]$ and split into 256-point windows for the DL experiments.

4.2. Transfer learning experiments

The first set of experiments tests the claim that SSL is a more effective TL pretraining method. The experimental design reflects the following assumptions:

1. labeled training data is available from a source set of process parameters,
2. unlabeled training data is available from a target set of process parameters, and
3. the pretrained model may encounter new fault types once deployed.

This scenario leads to three comparison methods:

- **Supervised (Source)**: supervised training on the labeled source domain data
- **Barlow Twins (Source)**: self-supervised training on the source domain data (ignoring labels)
- **Barlow Twins (Target)**: self-supervised training on the unlabeled target domain data

All three methods use the same 1D CNN feature extraction backbone G shown in Fig. 7. The supervised network adds the K-class classifier F_ϕ to the backbone, while Barlow Twins adds the projection head H_ψ. The networks F_ϕ and G_θ are then optimized using stochastic gradient descent and backpropagation with cross-entropy loss from (1). The Barlow Twins model produces projections $Z' = H_\psi(G_\theta(X'))$ and $Z'' = H_\psi(G_\theta(X''))$ from input batch augmentations X' and X'' (see Algorithm 2), and the training loss is computed from (5)–(7) with $\lambda = 0.01$. Both the supervised and self-supervised models are trained for 1000 epochs with an Adam optimizer and learning rate of 0.0005.

To assess the quality and generalizability of each method's representation, the frozen features of each pretrained network are used to train a privileged linear evaluation classifier with access to labeled target domain data from all eight health conditions (the *evaluation data set*), following conventions in the literature for evaluating SSL models [44]. Access to privileged label information prevents this classifier from being trained and deployed in practice, but it follows the accepted standard for assessing the separability of the underlying feature representations. The evaluation classifier is trained for 75 epochs on the frozen features, and the test set accuracy is used to judge the representation quality.

Table 1
Transfer learning health condition sets.

# of conditions	Condition classes
2	{N, PL} {PL, BoR} {BrR, UV} {UR, UV} {FB, UV}
4	{N, BrR, UR, UV} {PL, BrR, MR, UV} {FB, PL, BoR, UV} {FB, BrR, MR, UR} {BoR, BrR, MR, UR}
6	{N, FB, PL, BrR, MR, UR} {N, PL, BoR, MR, UR, UV} {N, PL, BoR, BrR, MR, UV} {N, FB, BoR, BrR, MR, UV} {N, PL, BoR, BrR, MR, UR}

Table 2
Federated learning health condition sets.

ID	Client 1	Client 2
1	{BoR, MR}	{BrR, UR}
2	{FB, UR}	{BrR, UV}
3	{BoR, N}	{BrR, FB}
4	{BrR, UV}	{UR, N}
5	{FB, MR}	{BoR, UV}

To simulate the occurrence of new, unseen faults, the source and target domain training data sets are limited to two, four, or six randomly selected health conditions. Since the evaluation data set contains all eight conditions, this corresponds to encountering six, four, or two previously unseen classes after pretraining, respectively.

To capture variation caused by the source/target domain selection, training health conditions, and model initialization, 450 experiments are conducted, 150 for each of the three comparative methods. The 150 runs come from all combinations of two source/target domain pairs (3L→2H or 2H→3L), 15 unique health condition configurations for the source/target training data, and five random seeds (0 through 4). The 15 combinations of training health conditions consist of five randomly sampled sets for each of two, four, and six health conditions (see Table 1). All experiments use an NVIDIA V100 GPU with 32 GB of RAM for hardware acceleration.

4.3. Federated learning experiments

The FL experiments determine whether sharing model information between clients with disjoint sets of training conditions will improve the distinguishability of future emerging faults. To evaluate this, two clients are each assigned two randomly selected motor health conditions. Each client has local training data for its two conditions from all process parameters combinations (i.e., 2L, 2H, 3L and 3H). The FL server provides both clients with an initial global model with random weights. In each round of FL, the clients train their local model on their unique set of two health conditions and then return the updated model to the server. The server averages the weights and redistributes the new model to the clients in preparation for the next round of FL (see Algorithm 1).

FL experiments are run for 1000 rounds, and each client trains for 20 local batches in each round. When performing supervised learning, each client updates the weights using cross-entropy loss from (1). For Barlow Twins training, each client uses the cross-correlation loss from (5)–(7). Both supervised learning and Barlow Twins use the same network architectures for TL shown in Fig. 6 and are trained with an Adam optimizer and learning rate of 0.0002.

Each of the four possible model configurations – supervised learning and Barlow Twins each with and without FL – is trained with five random seeds (0 through 4) to gauge variation caused by random initialization. Five unique sets of training conditions are tested to marginalize effects of individual health conditions (see Table 2). All combinations of the four methods, five seeds, and five condition sets lead to a total of 100 FL experiments. All experiments use an NVIDIA V100 GPU for hardware acceleration. Similar to TL, both clients are evaluated using the accuracy of a privileged linear classifier trained on the frozen feature extraction network to classify all eight conditions. The classifier is trained for 75 epochs after FL is complete.

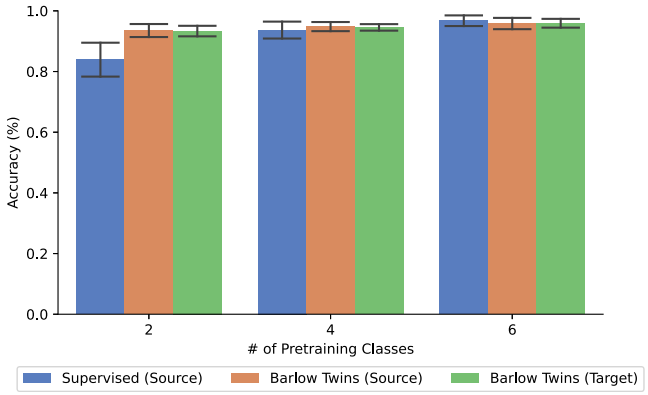


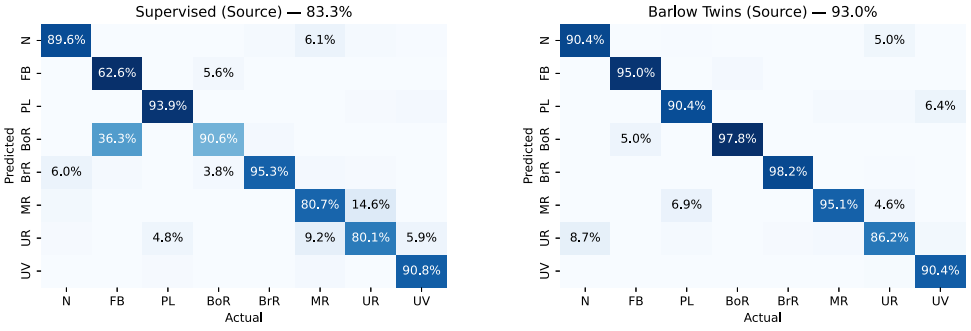
Fig. 10. Target domain accuracy of the weight transfer methods on all eight motors condition versus number of faults in the training domain.

5. Results and discussion

The results indicate that Barlow Twins produces more generalizable and transferable representations than supervised learning, and that FL for information sharing may further improve performance.

5.1. Transfer learning results

Table 3 and Fig. 10 present the key TL results comparing supervised learning on labeled source process parameters, Barlow Twins on unlabeled source process parameters, and Barlow Twins on unlabeled target process parameters. The accuracy metrics are computed from the test split of the evaluation data set containing all eight conditions under the target process parameters. Even when just two conditions are available for training, Barlow Twins generates a separable representation capable of 93.5% accuracy when shown all eight health conditions. In the same scenario, supervised learning is limited to 83.9% accuracy. Fig. 11 shows representative confusion matrices that highlight the improvements of SSL over supervised learning. For example, supervised learning struggles to distinguish the misaligned rotor (MR) and unbalanced rotor (UR) conditions while using Barlow Twins boosts the accuracy within these categories by 15 and 6 points, respectively. The SSL approach still confuses some classes (e.g., N↔UR and UR→{MR,N}) possibly because the random augmentations could not fully span the expected variation of these classes. That is, data labeled as normal varied more than the expected variation captured by the random jitter, scaling, and masking from Algorithm 2. As a result, Barlow Twins clustered some of these examples closer to UR, leading the evaluation classifier to miscategorize them. Similarly, some PL examples are classified as MR using Barlow Twins representations, indicating that these PL instances experience some variation that clustered them closer to MR. Future work can investigate whether these examples truly are miscategorized or actually resemble members of the confused class. Barlow Twins can also utilize unlabeled target domain data to further improve the representation – Barlow Twins (Target) in Table 3 – while supervised learning cannot use this data due to the lack of labels. Interestingly, Barlow Twins (Target) does not show a



(a) Supervised (Source) (b) Barlow Twins (Source)

Fig. 11. Representative confusion matrices showing the advantage of using Barlow Twins over supervised learning when transferring models to new process parameters (3L→2H) with six emerging conditions.

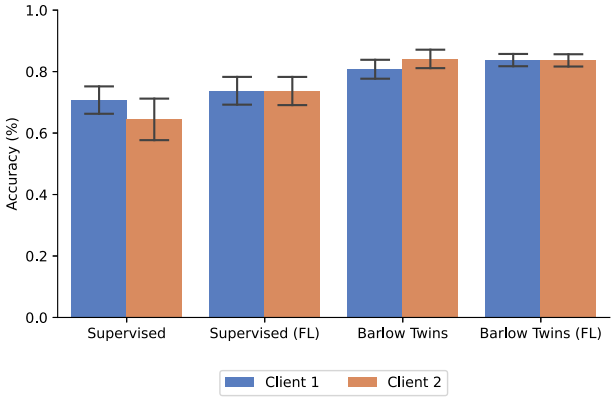


Fig. 12. Client evaluation accuracies on all health conditions.

Table 3
Transfer learning evaluation accuracy results (%).

Method	# of training health conditions		
	2	4	6
Supervised (Source)	83.9 ± 5.6	93.7 ± 2.8	**96.7 ± 1.8**
Barlow Twins (Source)	**93.5 ± 2.1**	**94.8 ± 1.5**	95.8 ± 1.9
Barlow Twins (Target)	93.3 ± 1.7	94.5 ± 1.1	95.9 ± 1.4

Table 4
Federated learning accuracy results (%).

Method	Client 1	Client 2	Overall
Supervised	70.7 ± 4.5	64.4 ± 6.8	67.6 ± 6.5
Supervised (FL)	73.8 ± 4.5	73.7 ± 4.6	73.7 ± 4.5
Barlow Twins	80.8 ± 3.1	**84.1 ± 3.0**	82.4 ± 3.5
Barlow Twins (FL)	**83.7 ± 2.0**	83.6 ± 2.0	**83.7 ± 2.0**

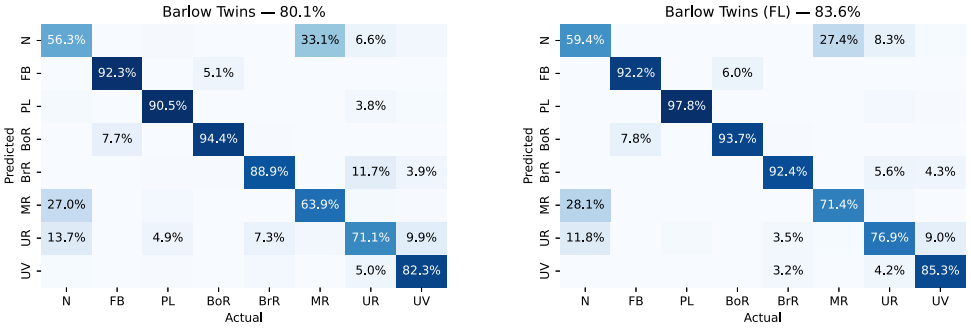
clear improvement over Barlow Twins (Source) indicating that SSL is effective for learning generalizable features from the motor condition monitoring source domain data.

As more conditions are included in training, the performance convergence of supervised learning and Barlow Twins can be explained according to the optimization objective of each approach. Supervised learning seeks to split the data along decision boundaries for the classifier. While this may ensure the training classes are distinguishable, it does not guarantee compactness of the feature clusters. Thus, it is suspected that features from new, emerging faults could overlap with those from faults seen in training. In contrast, Barlow Twins encourages similar input instances to have correlated and closely matching features. This emphasis on feature similarity produces tight clusters that reduce the likelihood of new fault features overlapping with existing clusters. When the number of training conditions increases, the additional decision boundaries created by supervised learning naturally improve feature cluster compactness, bringing its evaluation accuracy closer to that of Barlow Twins. However, because manufacturing applications will have limited class diversity compared to the possible number of emerging faults, these results show the general superiority of SSL-based representations over those transferred from supervised learning in uncertain operating environments.

5.2. Federated learning results

Table 4 and Fig. 12 present the FL results. Supervised learning shows an noticeable increase in discriminability of emerging faults when FL is included. Without FL, the overall evaluation accuracy between the clients is only 67.6%. When FL is included, information about the health conditions is shared indirectly through the `FedAvg` server, boosting the overall accuracy to 73.7%. Since both clients share a global model during FL, they have nearly identical accuracy. When trained without FL, the supervised learning clients show a 6-point discrepancy.

Barlow Twins outperforms all supervised learning methods even when FL is excluded. The separately-trained clients reach an overall evaluation accuracy of 82.4%. Once FL combined with Barlow Twins, performance increases to 83.7%, the highest overall accuracy among all methods. As in the supervised case, FL also reduces the discrepancy between the clients, reducing the accuracy difference from 3.3 points to 0.1 point. The representative confusion matrices in Fig. 13 show in the improvement in Client 1 when FL is included. Phase loss (PL) accuracy increases from 90.5% to 97.8%, and misaligned rotor (MR) accuracy increases from 63.9% to 71.4%. The differences in accuracy with respect to the TL-only results may be a result of training with all sets of process parameters instead of a single source domain set. The Barlow Twins data augmentations might be effective for one or two process parameter sets, they require additional development to capture the class variation expected across all the process parameter combinations. For example, the TL experiments might more easily distinguish N vs. MR and UR because the transfer occurred between 2H↔3L naturally leading to more distant clusters than when data contains only a single process parameter change. While these are directions for future work, these preliminary results demonstrate how indirect information sharing through the `FedAvg` server may be able to boost discriminability of emerging faults, if the individual clients see a limited number of distinct health conditions. By merging models trained on different subsets of health conditions, FL may increase the diversity of the training data set, improving the generalization of the learned features.



(a) Without FL (b) With FL

Fig. 13. Representative confusion matrices showing the benefits of including FL for Barlow Twins Client 1. Client 1 was trained on {BoR, N}, while Client 2 (not shown) was trained on {BrR, FB}.

5.3. Limitations and future directions

While the results are promising, the case study focuses on motor condition monitoring. Additional experiments are necessary to validate the approach more fully against a variety of manufacturing problems and data sets. In addition, while SSL facilitates learning discriminative representations without labeled data, downstream classification tasks still require an additional step to either cluster features automatically into presumed class groups or integrate a human-in-the-loop solution in which an operator can tag a limited number of features with labels. Future work should also characterize when SSL and FL approaches struggle with manufacturing data. Understanding possible shortcomings and failure modes will enable practitioners to rapidly implement the right technology for a given problem.

6. Conclusion

Given growing developments in SSL, this study compares the generalization of feature representations learned via SSL versus those learned via supervised methods. In weight transfer experiments, a feature extractor trained with Barlow Twins outperformed a supervised classifier when transferring to an operating environment with different process parameters that contained emerging faults. With only two health conditions for training, the features learned by Barlow Twins from the source domain produced an evaluation classifier accuracy 9.6 points higher than that of the representation learned by supervised training on labeled source domain data. To further improve performance, knowledge of distributed but similar SSL client models can inform an FL architecture that shares fault experience while respecting privacy concerns. Thus, manufacturing applications with large unlabeled data sets can use SSL and FL to learn generalizable representations for emerging faults even without diverse, labeled data. With enhanced emerging fault detection across conditions, models will be better equipped for the factory floor and improve the trustworthiness and reliability of practical condition monitoring deployments.

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.

Acknowledgments

This work is supported by the National Science Foundation under Grant No. 2015889. We would thank the University of Kentucky Center for Computational Sciences and Information Technology Services for their support and use of the Lipscomb Compute Cluster and associated research computing resources.

References

[1] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems, Vol. 25. 2012, p. 1097–105.
[2] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436–44.
[3] Wen L, Li X, Gao L, Zhang Y. A new convolutional neural network-based data-driven fault diagnosis method. IEEE Trans Ind Electron 2017;65:5990–8.
[4] Li Z, Zhang W, Ding Q, Sun J-Q. Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation. J Intell Manuf 2020;31:433–52.
[5] Sun S, Wang T, Yang H, Chu F. Adversarial representation learning for intelligent condition monitoring of complex machinery. IEEE Trans Ind Electron 2022;70(5):5255–65.
[6] Yu X, Zhao Z, Zhang X, Zhang Q, Liu Y, Sun C, Chen X. Deep-learning-based open set fault diagnosis by extreme value theory. IEEE Trans Ind Inf 2021;18(1):185–96.
[7] 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 2021;26(3):1591–601.
[8] Fu Y, Cao H, Cheng X, Ding J. Broad auto-encoder for machinery intelligent fault diagnosis with incremental fault samples and fault modes. Mech Syst Signal Process 2022;178:109353.
[9] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng 2010;22(10):1345–59.
[10] Kouw WM, Loog M. An introduction to domain adaptation and transfer learning. 2018, arXiv:1812.11806.
[11] Dosovitskiy A, Springenberg JT, Riedmiller M, Brox T. Discriminative unsupervised feature learning with convolutional neural networks. In: Advances in neural information processing systems, Vol. 27. 2014, p. 766–74.
[12] McMahan HB, Moor E, Ramage D, Hampson S, y Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th international conference on artifical intelligence and statistics (AISTATS). 2017.
[13] Wang P, Gao RX. Transfer learning for enhanced machine fault diagnosis in manufacturing. CIRP Ann Manuf Technol 2020;69:413–6.
[14] Lu W, Liang B, Cheng Y, Meng D, Yang J, Zhang T. Deep model based domain adaptation for fault diagnosis. IEEE Trans Ind Electron 2017;64(3):2296–305.
[15] Yang B, Lei Y, Jia F, Li N, Du Z. A polynomial kernel induced distance metric to improve deep transfer learning for fault diagnosis of machines. IEEE Trans Ind Electron 2019;67(11):9747–57.
[16] Cao H, Shao H, Zhong X, Deng Q, Yang X, Xuan J. Unsupervised domain-share CNN for machine fault transfer diagnosis from steady speeds to time-varying speeds. J Manuf Syst 2022;62:186–98.
[17] Yang B, Lei Y, Jia F, Xing S. An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings. Mech Syst Signal Process 2019;122:692–706.
[18] Su K, Liu J, Xiong H. A multi-level adaptation scheme for hierarchical bearing fault diagnosis under variable working conditions. J Manuf Syst 2022;64:251–60.
[19] Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V. Domain-adversarial training of neural networks. J Mach Learn Res 2016;17:1–35.
[20] Li X, Zhang W, Ding Q, Li X. Diagnosing rotating machines with weakly supervised data using transfer learning. IEEE Trans Ind Electron 2019;16(3):1688–97.
[21] Guo L, Lei Y, Xing S, Yan T, Li N. Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data. IEEE Trans Ind Electron 2018;66(9):7316–25.

[22] Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks? In: Advanced in neural information processing systems, Vol. 27. 2014, p. 3320–8.
[23] Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, Xiong H, He Q. A comprehensive survey on transfer learning. Proc IEEE 2020;109(1):43–76.
[24] Shao S, McAleer S, Yan R, Baldi P. Highly accurate machine fault diagnosis using deep transfer learning. IEEE Trans Ind Inf 2018;15(4):2446–55.
[25] Wang X, Shen C, Xia M, Wang D, Zhu J, Zhu Z. Multi-scale deep intra-class transfer learning for bearing fault diagnosis. Reliab Eng Syst Saf 2020;202:107050.
[26] He Z, Shao H, Ding Z, Jiang H, Cheng J. Modified deep autoencoder driven by multisource parameters for fault transfer prognosis of aeroengine. IEEE Trans Ind Electron 2021;69(1):845–55.
[27] Gidaris S, Singh P, Komodakis N. Unsupervised representation learning by predicting image rotations. In: International conference on learning representations (ICLR). 2018.
[28] Doersch C, Gupta A, Efros AA. Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE international conference on computer vision (ICCV). 2015.
[29] Radford A, Narasimhan K, Salimans T, Sutskever I. Improving language understanding by generative pre-training. 2018, https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.
[30] Wang T, Qiao M, Zhang M, Yang Y, Snoussi H. Data-driven prognostic method based on self-supervised learning approaches for fault detection. J Intell Manuf 2018;31:1611–9.
[31] Zhang W, Chen D, Kong Y. Self-supervised joint learning fault diagnosis method based on three-channel vibration images. Sensors 2021;21:4774.
[32] Shul Y, Yi W, Choi J, Kang D-S, Choi J-W. Noise-based self-supervised anomaly detection in washing machines using a deep neural network with operational information. Mech Syst Signal Process 2023;189:110102.
[33] Kim J, Yang Z, Ko H, Cho H, Lu Y. Deep learning-based data registration of melt-pool-monitoring images for laser power bed fusion additive manufacturing. J Manuf Syst 2023;68:117–29.
[34] Zhang T, Chen J, He S, Zhou Z. Prior knowledge-augmented self-supervised feature learning for few-shot intelligent fault diagnosis of machines. IEEE Trans Ind Electron 2022;69(10):10573–84.
[35] Wang H, Liu Z, Ge Y, Peng D. Self-supervised signal representation learning for machinery fault diganosis under limited annotation data. Knowl-Based Syst 2022;239:107978.
[36] Nie G, Zhang Z, Shao M, Jiao Z, Li Y, Li L. A novel study on a generalized model based on self-supervised learning and sparse filtering for intelligent bearing fault diagnosis. Sensors 2023;23(4):1858.
[37] Hadsell R, Chopra S, LeCun Y. Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06). 2006, p. 1063–6919.
[38] van den Oord A, Li Y, Vinyals O. Representation learning with contrastive predictive coding. 2018, arXiv:1807.03748.
[39] He K, Fan H, Wu Y, Xie S, Girshick R. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR). 2020, p. 9729–38.
[40] Hermans A, Beyer L, Leibe B. In defense of the triplet loss for person re-identification. 2017, arXiv:1703.07737.
[41] Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th international conference on machine learning. Proceedings of machine learning research, vol. vol. 119, 2020, p. 1597–607.
[42] Grill J-B, Strub F, Altché F, Tallec C, Richemond PH, Buchatskaya E, Doersch C, Pires BA, Guo ZD, Azar MG, Piot B, Kavukcuoglu K, Munos R, Valko M. Bootstrap your own latent - a new approach to self-supervised learning. In: Advances in neural information processing systems, Vol. 33. 2020.
[43] Chen X, He K. Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR). 2021, p. 15750–8.
[44] Zbontar J, Jing L, Misra I, LeCun Y, Deny S. Barlow twins: Self-supervised learning via redundancy reduction. In: Proceedings of the 38th international conference on machine learning. Proceedings of machine learning research, vol. vol. 139, 2021, p. 12310–20.
[45] Bardes A, Ponce J, LeCun Y. VICReg: Variance-invariance-covariance regularization for self-sueprvised learning. In: International conference on learning representations (ICLR). 2022.
[46] Wei M, Liu Y, Zhang T, Wang Z, Zhu J. Fault diagnosis of rotating machinery based on improved self-supervised learning method and very few labeled samples. Sensors 2022;22(1):192.
[47] Zhang W, Chen D, Xiao Y, Yin H. Semi-supervised contrast learning based on multi-scale attention and multi-target contrast learning for bearing fault diagnosis. IEEE Trans Ind Inf 2023.
[48] Balestriero R, Ibrahim M, Sobal V, Morcos A, Shekhar S, Goldstein T, Bordes F, Bardes A, Mialon G, Tian Y, Schwarzschild A, Wilson AG, Geiping J, Garrido Q, Fernandez P, Bar A, Pirsiavash H, LeCun Y, Goldblum M. A cookbook of self-supervised learning. 2023, arXiv:2304.12210.
[49] Hu C, Qu J, Sun C, Yan R, Chen X. Inter-instance and intra-temporal self-supervised learning with few labeled data for fault diagnosis. IEEE Trans Ind Inf 2023.
[50] Ding Y, Zhuang J, Ding P, Jia M. Self-supervised pretraining via contrast learning for intelligent incipient fault detection of bearings. Reliab Eng Syst Saf 2022;218, Part A:108126.
[51] Peng T, Shen C, Sun S, Wang D. Fault feature extractor based on bootstrap your own latent and data augmentation algorithm for unlabeled vibration signals. IEEE Trans Ind Electron 2022;69(9):9547–55.
[52] Wan W, Chen J, Zhou Z, Shi Z. Self-supervised simple siamese framework for fault diagnosis of rotating machinery with unlabeled samples. IEEE Trans Neural Netw Learn Syst 2023.
[53] Russell M, Wang P, Liu S, Jawahir IS. Mixed-up experience replay for adaptive online condition monitoring. IEEE Trans Ind Electron 2023;1–8.
[54] Zhang W, Li X, Ma H, Luo Z, Li X. Federated learning for machinery fault diganosis with dynamic validation and self-supervision. Knowl-Based Syst 2021;213:106679.
[55] Wang H, Liu C, Jiang D, Jiang Z. Collaborative deep learning framework for fault diagnosis in distributed complex systems. Mech Syst Signal Process 2021;156:107650.
[56] Xia L, Zheng P, Li J, Tang W, Zhang X. Privacy-preserving gradient boosting tree: Vertical federated learning for collaborative bearing fault diagnosis. IET Collaborat Intell Manuf 2022;4(3):208–19.
[57] Du NH, Long NH, Ha KN, Hoang NG, Huong TT, Tran KP. Trans-Lighter: A light-weight federated learning-based architecture for Remaining Useful Lifetime prediction. Comput Ind 2023;148:103888.
[58] Mehta M, Shao C. Federated learning-based semantic segmentation for pixel-wise defect detection in additive manufacturing. J Manuf Syst 2022;64:197–210.
[59] Deng T, Li Y, Liu X, Wang L. Federated learning-based collaborative manufacturing for complex parts. J Intell Manuf 2022.
[60] Ding L, Wu J, Li C, Jolfaei A, Zheng X. SCA-LFD: Side-channel analysis-based load forecasting disturbance in the energy internet. IEEE Trans Ind Electron 2023;70(3):3199–208.
[61] Wang KI-K, Zhou X, Liang W, Yan Z, She J. Federated transfer learning based cross-domain prediction for smart manufacturing. IEEE Trans Ind Inf 2021;18(6):4088–96.
[62] Shi N, Kontar RA. Personalized federated learning via domain adaptation with an application to distributed 3D printing. Technometrics 2022;65(3):328–39.
[63] Mehta M, Chen S, Tang H, Shao C. A federated learning approach to mixed fault diagnosis in rotating machinery. J Manuf Syst 2023;68:687–94.
[64] Li X, Zhang C, Li X, Zhang W. Federated transfer learning in fault diagnosis under data privacy with target self-adaptation. J Manuf Syst 2023;68:523–35.
[65] Mehta M, Shao C. A greedy agglomerative framework for clustered federated learning. IEEE Trans Ind Inf 2023;Early Access:1–12.
[66] Tian Y, Wang Y, Krishnan D, Tenenbaum JB, Isola P. Rethinking few-shot image classification: A good embedding is all you need? In: Computer vision – ECCV 2020. 2020, p. 266–82.