Supervised Contrastive Learning for Electric Motor Bearing Fault Detection

Zhang, Hengrui; Wang, Bingnan

TR2024-120 September 04, 2024

Abstract

Various faults can cause electric machine failures, causing downtime and asset losses. Fault detection technologies are highly desirable in the industry to predict and prevent such failures. Recent advances in machine learning have enabled data- driven models that identify faults from signals monitored in the motors. However, those signals could be complex and the features that indicate faults are subtle. Therefore, effective methods for extracting informative features relevant to faults from signals are desired. In this paper, we explore the use of contrastive learning in the detection of bearing faults from phase current signals. We develop a model architecture consisting of two parts, a feature extractor and a classifier, where the feature extractor is pre-trained using supervised contrastive learning. Tested on the Paderborn University bearing fault dataset, our model attains a high fault classification accuracy of 87%, which outperforms the conventional machine learning models. We also perform ablation tests to demonstrate the importance of contrastive learning- based training in this model. By investigating the classification results and extracted features of the models, we further verify the effectiveness of contrastive learning in extracting features that distinguish different classes. We anticipate that contrastive learning can lay the foundation of more accurate fault detection models and be extended to other practical fault detection tasks.

International Conference on Electrical Machines (ICEM) 2024

Supervised Contrastive Learning for Electric Motor Bearing Fault Detection

Hengrui Zhang
Mechanical Engineering
Northwestern University
Evanston, IL 60208, USA
hrzhang@u.northwestern.edu

Bingnan Wang
Mitsubishi Electric Research Laboratories
Cambridge, MA 02139, USA
bwang@merl.com

***Abstract*—Various faults can cause electric machine failures, causing downtime and asset losses. Fault detection technologies are highly desirable in the industry to predict and prevent such failures. Recent advances in machine learning have enabled data-driven models that identify faults from signals monitored in the motors. However, those signals could be complex and the features that indicate faults are subtle. Therefore, effective methods for extracting informative features relevant to faults from signals are desired. In this paper, we explore the use of *contrastive learning* in the detection of bearing faults from phase current signals. We develop a model architecture consisting of two parts, a feature extractor and a classifier, where the feature extractor is pre-trained using supervised contrastive learning. Tested on the Paderborn University bearing fault dataset, our model attains a high fault classification accuracy of 87%, which outperforms the conventional machine learning models. We also perform ablation tests to demonstrate the importance of contrastive learning-based training in this model. By investigating the classification results and extracted features of the models, we further verify the effectiveness of contrastive learning in extracting features that distinguish different classes. We anticipate that contrastive learning can lay the foundation of more accurate fault detection models and be extended to other practical fault detection tasks.**

***Index Terms*—fault detection, electric machines, bearing fault, machine learning, contrastive learning**

I. Introduction

Electric motors are crucial components in various industrial and commercial applications, such as pumps, fans, conveyors, and HVAC systems. The seamless operation of these motors is essential for productivity, efficiency, and safety in various sectors. However, like all mechanical and electrical components, electric motors are susceptible to faults and failures over time, leading to costly downtime and operational disruptions. Therefore, the need for reliable and timely fault detection in electric motors has never been more critical. This pursuit not only ensures the longevity and optimal performance of these vital machines but also contributes significantly to energy efficiency and sustainability goals.

Advances in sensor technologies and data analysis techniques have facilitated accurate and efficient motor fault detection. During the operation of a motor, multiple forms of signals can be collected and monitored, including phase currents, vibration, acoustics, and temperature change. If there is prior knowledge available about how faults may change specific features of signals, faults can be detected by simply applying some filters or rules. Unfortunately, in most cases, the effect of motor faults can be complicated, and such knowledge is not available. Without requiring domain knowledge, fault can be identified via *anomaly detection*: signals that show distinct features compared to other parts (assuming the motor is healthy at most times) indicate the presence of faults. However, anomaly detection merely finds outliers based on patterns in the signals, but cannot reveal the type or severity of the fault, and the detected signals often need human verification.

Among many types of motor faults, bearing faults are the most common cause of motor failures [1]. Accurate bearing fault detection has been a research focus in the industry. Basic physical models have been established to identify the connection between bearing faults and measurable signals, including vibration [2], acoustic noise [3], and stator current [4]. As a type of mechanical fault, bearing defects impact the mechanical response of the motor system, which can be directly reflected in vibration signals and high-frequency acoustic noises. However, bearing fault detection accuracy based on vibration or acoustic signals can be unreliable, due to various factors, such as the often large background noises in a factory environment, and signal variations depending on sensor locations [5], [6]. The motor current signature analysis (MCSA) approach, on the other hand, detects motor faults based on stator current signals [7]–[9]. It has the advantage of cost-saving with no additional sensors required. Physical models for MCSA can be established for well-defined single-point bearing faults, such as inner race fault, outer race fault, and ball fault [10], [11]. Characteristic bearing fault frequencies can be identified in motor current signals. However, in naturally occurring bearing faults, the signals can be much more complicated and challenging to model and identify with MCSA.

On the other hand, data-driven approaches enable more general and accurate electric motor diagnosis. Machine learning (ML) models are capable of pattern recognition from high-dimensional, complex data. With the simulated and experimental data available for various fault types, ML could treat fault detection in a supervised classification fashion, predicting

This work was done when Hengrui Zhang was with Mitsubishi Electric Research Laboratories.

fault types and severity given observed signals. A plethora of ML models have been developed for motor fault detection, including conventional statistical ML as well as deep learning methods [12]. Most of the existing works use vibration signals as input for ML models, since faults usually exert significant change in the motor's vibration frequency, and thus detecting fault is relatively easy. With MCSA, the information related to bearing faults is much more subtle and difficult to detect compared with conventional vibration analysis. Many ML models work well on vibration signals but often fail on motor current signals for fault detection purposes [12]. Therefore, it is essential to be able to extract fault-related features in motor current signals for the development of data-driven approaches based on MCSA.

The representation of data plays a key role in data-driven studies, including ML. For an ML model to effectively classify signals, the input signal (in our case, currents) should be presented in an appropriate format. In the context of classification, the appropriateness of input format is associated with classes (i.e., fault types). Ideally, the input data (1) captures the difference between classes, and (2) discards the common features that all classes share. It should also be compatible with the model in terms of size, structure, etc.

To prepare ML-ready data from currents, several *feature engineering* methods have been applied. One straightforward way is to directly use the signal as time series data. The length and resolution of such time series are carefully selected to ensure that the data captures relevant information and has a reasonable size. Another widely adopted feature engineering technique is Fourier transform (FT). Because of the effect of bearing fault on frequency, the frequency spectra obtained from FT are expected to be relevant based on domain knowledge. However, such feature engineering techniques do not ensure the separability of classes, and classification remains challenging.

To address the challenge, we seek more effective ways to extract critical information from current signals. We choose contrastive learning (CL), an effective representation learning method, to facilitate bearing fault diagnosis. We show that with supervised contrastive learning, effective bearing fault detection can be achieved using motor current signals. The remaining parts of this paper are organized as follows: Section II describes the experiment dataset for the study and the data analysis. Section III introduces the CL method and its related applications. Sections IV–V present the CL-based model development and results. Section VI discusses future work and concludes the paper.

II. Dataset and Initial Analysis

A. Dataset Overview

In this work, we use the bearing fault dataset from Paderborn University [13] as a testbed for our ML models. This dataset contains the synchronous measurement of motor current and vibration signals obtained from the test rig shown in Figure 1, which consists of the following modules: an electric motor (1), a torque-measurement shaft (2), a rolling bearing test module (3), a flywheel (4) and a load motor (5). Bearings under study are mounted in the external bearing test module, which is directly coupled to the drive motor via a shaft.

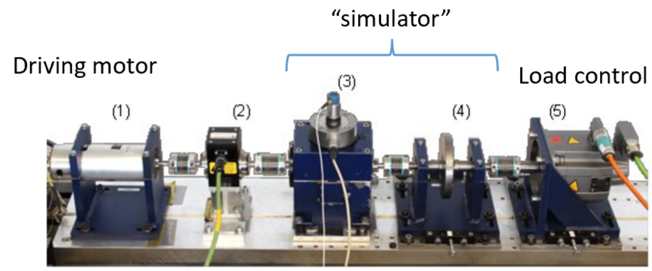

Fig. 1. Experimental setup of the Paderborn University dataset [13]. Component (1) is the driving motor where the currents are measured. Components (3–4) are a "simulator" of motor, on which the bearing faults are manually created. The load is controllable via component (5).

The dataset serves as a great benchmark for bearing fault diagnosis and classification, as it includes data measured and various bearing fault conditions, under various operating conditions. It provides signals measured for bearings with 4 different fault conditions: healthy, inner race fault, outer race fault, and mixed fault. These include both artificial (created by machining of trenches, drilling, or engraving) and near natural (through accelerated aging tests) faults. For each fault type, 8 different severity levels are included, making 32 *health conditions* in total. Each bearing is studied under 4 different operating conditions (characterized by torque, rotational speed, and radial force). 20 repetitive measurements are done for every combination of bearing and operating conditions. Each measurement lasts 4 seconds, with a sampling rate of 64 kHz, producing a time series of length 256,000. Currents of 2 phases and vibration in 1 direction are measured. In this work, we focus on the effectiveness of ML models for bearing fault detection using current signals only. Therefore only the phase current data are used in the following sections. In addition, for simplicity, we focus on detecting bearing faults at a single location, without considering mixed faults (damage on both inner and outer races).

B. Data Analysis

For both sanity checks and examination of feature engineering techniques, we perform some initial analyses of the data. Figure 2 (a) shows segments of one phase current in a short time period (0.05 s), obtained in one example measurement of each class. Visually, they look similar (except for a phase shift, which is irrelevant to fault detection). ML tests also suggest that it is difficult to identify their differences as related to faults. In Figure 2 (b), we show their frequency spectra, obtained via FT and plotted in log-scale. In the frequency domain, the features usually lie in *characteristic frequencies*, reflected by peaks in the spectra. However, the three classes exhibit almost identical sets of peaks, despite changes in the relative heights. So, the frequency spectra might contain features that distinguish fault types, but they are too subtle to be found visibly and described clearly. Thus we utilize contrastive learning based ML for the distinction of those visually similar signals and bearing fault detection and classification.

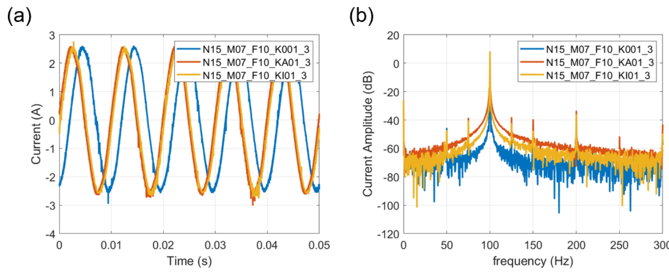

Fig. 2. Comparison of phase current signals for healthy motor (blue), inner race fault (yellow), and outer race fault (red), in the time domain (a) and frequency domain (b).

III. Contrastive Learning

Contrastive learning (CL) is a paradigm of ML that focuses on differentiating between similar and dissimilar instances in the data. This method is applicable in both supervised and unsupervised settings. The core mechanism of CL involves drawing contrasts between paired instances, a process that facilitates intricate feature extraction and the learning of robust representations. Crucially, CL leverages relational information between samples, as opposed to isolating individual sample characteristics. The following subsections will delve into the specifics of CL, exploring its mechanics, its applications, and the foundational principles that underpin its role within the broader context of ML.

A. Basic concepts

Consider a set of data with corresponding classes $\{(\boldsymbol{x}_i, y_i)\}$, where the class labels y_i can be either explicitly known or implicit. The samples $\boldsymbol{x}$ may take arbitrary forms, from simple vectors to complex images or spectra. The goal of CL is to learn a representation space (often referred to as "latent space"), where samples of the same class are close, while samples of different classes are far apart, as Figure 3 shows.

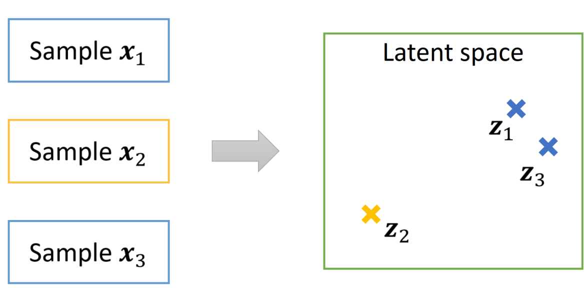

Fig. 3. Illustration of the goal of contrastive learning. The model maps a sample $\boldsymbol{x}$ to a latent representation $\boldsymbol{z}$. In the latent space, $\boldsymbol{z}_1$ and $\boldsymbol{z}_3$ which represent samples of the same class are close, whereas $\boldsymbol{z}_2$, representing a sample of a different class, are far from them.

Rather than a specific model architecture, CL defines a general framework and a way of training ML models, distinct from its counterparts such as supervised, unsupervised, and reinforcement learning. The model $f : \boldsymbol{x} \to \boldsymbol{z}$ that maps samples to latent representations can be designed according to the samples' format. The key component of CL is a contrastive loss, built upon a similarity (or distance) metric for $\boldsymbol{z}$: minimizing the loss function enlarges the similarity within the same class and shrinks across classes. A typical example is the loss function presented in [14]:

$$\mathcal{L} = -\sum_{i\in I} \log \frac{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_{j(i)})/\tau}{\sum_{a\in A(i)} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_a)/\tau}, \tag{1}$$

which is designed for the self-supervised setting, where all samples are assumed to come from different classes. For an "anchor index" i, $\boldsymbol{z}_i \cdot \boldsymbol{z}_{j(i)}$ is the cosine similarity between it and *positive* samples $j(i)$ obtained using data augmentation from i; $\boldsymbol{z}_i \cdot \boldsymbol{z}_a$ is the cosine similarity between it and *negative* samples in the set $A(i) \equiv I \setminus \{i\}$. A "temperature" parameter τ controls the strength of the penalty on similarity with negative samples. Various forms of contrastive losses employ other similarity metrics and mathematical transformations to improve the behavior of the function. An in-depth examination of the contrastive losses can be found at [15].

B. Supervised contrastive learning

When class labels are available, CL can utilize the information provided by labels to facilitate the organization of latent space. The work [14] develops a modified loss function for this purpose:

$$\mathcal{L} = \sum_{i\in I} \frac{-1}{|P(i)|} \sum_{p\in P(i)} \log \frac{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_p)/\tau}{\sum_{a\in A(i)} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_a)/\tau}, \tag{2}$$

where $P(i)$ and $A(i)$ are sets of positive and negative samples, respectively. The idea is similar to its self-supervised counterpart.

Even though complete knowledge of class labels allows training a classification model via a conventional supervised approach (e.g., by minimizing cross-entropy loss or hinge loss), supervised contrastive learning (SCL) has its unique advantages. Most importantly, contrastive learning focuses on learning *discriminative* representations by pushing apart dissimilar data points, thus extracting the distinctive features of every class. Such features are often more robust and less susceptible to minor variations (or noise) in input data. Besides, this provides transferability: the learned representations encapsulate meaningful and diverse characteristics of the data and can be transferable to other tasks or domains. Contrastive learning-based pre-training is also likely to mitigate the risk of overfitting, especially when the amount of data is limited.

C. Related applications

There have been some efforts to apply CL to motor fault detection. Ding et al. [16] use self-supervised CL for pre-training followed by semi-supervised fine-tuning on partially annotated data. The data are run-to-failure vibration signals collected from accelerated experiments. Signals are collected by accelerometers in two directions, at a sampling rate of 25.6 kHz. The model is built upon the momentum contrast method [17] and a ResNet-50 backbone model. It takes time series data as input and outputs a binary classification of normal/fault. Some other deep learning models trained in a supervised way are used as the baseline.

Another related work [18] differs from this one in method and settings: using the vibration data from the Case Western Reserve University (CWRU) dataset, the work aims at a more specific classification of multiple bearing fault types (inner race, outer race, ball damage) and severity. The representation learning part uses a 4-layer convolutional neural network (CNN) as backbone, and trains in a self-supervised CL fashion using the semi-hard triplet loss [19]. Vibration signals' frequency spectra are input to the CL-based representation learning model, and the extracted "features" are used as inputs for downstream classification (SVM) and clustering models. The CL model is compared against other feature extractors, such as autoencoders. A special setting considered in this work is fault detection under "novel conditions", i.e., testing the model on operating conditions that are not in the training set.

There are also some works focusing on specific aspects. Zhang et al. [20] use supervised CL to improve fault detection using imbalanced data, e.g., abundant data for normal conditions and scarce data for faults. Another work [21] focuses on *domain adaptation*, i.e., improving the generalizability of models trained on one domain (for example, artificial faults) in the target domain (ideally, real faults). Both works use CWRU's vibration dataset.

In a broader sense, some related works apply CL to the diagnosis (1) of other machinery and (2) using other methods. As an example of (1), Wang et al. [22] uses self-supervised CL to detect likely-fault vibration signals in marine engines. As for (2), Yang et al. [23], [24] explore the use of CL in graph-based fault diagnosis. Beyond the scope of fault diagnosis, researchers (mainly in the ML community) have been tailoring CL methods for high-frequency time series data, which is arguably the most important data type for health monitoring. Zhang et al. [25] develop a CL-based pre-training method for time series; Chen et al. [26] uses CL to attain few-shot learning for high-frequency time series. The methods therein could be adopted for electric motor diagnosis.

While most existing fault diagnosis methods rely on vibration signals, in this work we apply supervised contrastive learning to motor stator current signals measured in the Paderborn Unversity dataset, and show that effective bearing fault detection can be achieved.

IV. Model Development

We develop a fault detection model consisting of two parts: (1) a *feature extractor* that maps input data to an *embedding* space, and (2) a *classifier* (one fully connected layer) that maps an embedding vector to class probabilities. Figure 4 shows the architecture of our model.

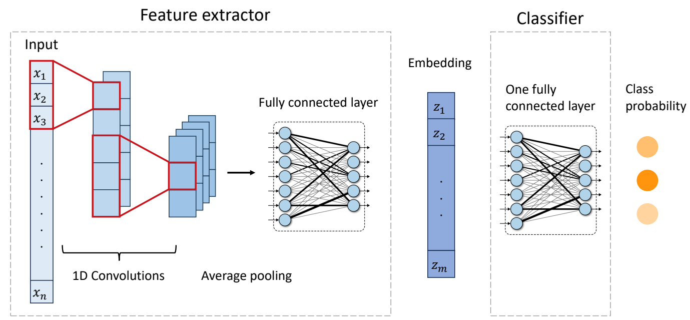

Fig. 4. Illustration of our motor fault detection model.

The input data for motor fault detection (time series or frequency spectra) are usually long one-dimensional (1D) sequences. Due to the cost of experimentation, the amount of data is usually limited, so, the model should not have too many parameters. Considering these, we choose the 1D convolutional neural network (CNN) as the backbone of the feature extractor model because of its expressiveness and parameter efficiency. The input data goes through multiple 1D convolutional (Conv1D) layers, with length reduced and the number of channels increased after each layer. Every convolutional layer contains max pooling, leaky ReLU activation, and batch normalization. Then, an average pooling is performed (along the length direction), followed by a fully connected (FC) layer that outputs an m-dimensional embedding vector. The embedding is rescaled to have a unit norm.

The model is trained in two stages. First, the "feature extractor" part is trained by minimizing the supervised contrastive loss (Equation 2). Afterward, the weights are recorded, and the whole model (feature extractor + classifier) is trained together in a conventional classification setting, i.e., by minimizing the cross-entropy loss. In this second stage, the weights of the feature extractor are allowed to change. This can be viewed as fine-tuning. At the end, the whole trained model is evaluated on the test set.

To stabilize training and prevent overfitting, multiple strategies are utilized. First, we train the model using the AdamW optimizer [27] with a "weight decay", which applies L_2 regularization to the weights. Second, the learning rate is controlled by a scheduler that reduces it on the plateau, specifically, if the validation error has not improved more than a small threshold for 10 epochs, the learning rate is reduced by a factor of 10. Third, we use early stopping: after a preset number of epochs ("patience"), if the validation error has not improved for 10 epochs, training is terminated and the model state exhibiting the lowest error is recorded. We have also tested dropout, however, it is found to hinder training in our case.

The model contains several hyperparameters. Some of them are associated with model architecture, such as the number and specification of convolutional layers. We have tested the effects of these hyperparameters and combinations of them, and choose optimal ones to build our model. When the choice of certain hyperparameters makes little difference, we use values that lead to a simpler model (e.g., two layers instead of three; 32 filters instead of 64). Table I shows the model specifications with the optimal hyperparameters we find. Other hyperparameters control the training process, such as "patience" for early stopping and strength of weight decay. These are tuned each time when a model is trained on different data.

TABLE I
MODEL SPECIFICATIONS: HYPERPARAMETERS FOR EVERY LAYER; SHAPE FOR INPUT DATA AND THE EMBEDDING SPACE.

Layers	Specifications
Input	Shape: (2 channels, signal length)
Conv1D Layer 1	Kernel size: 3, stride: 2, padding: 1, 32 filters
Conv1D Layer 2	Kernel size: 3, stride: 2, padding: 1, 64 filters
Pooling Layer	1D adaptive average pooling
FC Layer 1	Dimensions: (64, 32)
Embeddings	Dimension: 32
FC Layer 2	Dimensions: (32, 3)

V. RESULTS AND DISCUSSIONS

We test the capability of the SCL-enhanced motor fault detection model using the Paderborn dataset. The experiments have three purposes:

1) Find out what is a suitable representation of input data for SCL.
2) Ablation test on the effectiveness of SCL compared to conventional learning schemes.
3) Benchmark of the SCL-trained deep learning model against conventional ML models.

The data preparation and experimental design are described as follows.

A. Experimental Design

As SCL is a general framework designed to extract information from data, we use common representations of current signals in time and frequency domains, without too much feature engineering. For the time domain, we divide the raw data (64 kHz, 4 s, 256,000 time steps) into segments with lengths of 64 ms (4096 time steps), keeping the original sampling rate. For the frequency domain, we first take segments of length 1 s, then downsample the signals to a rate of 2 kHz and perform Fourier transform, yielding frequency spectra from 0 to 1000 Hz, with a frequency resolution of 1 Hz. The downsampling is done by taking one time step out of every 32 (i.e., skipping 31 time steps). By shifting the downsample selection (i.e., 0-32-64-... to 1-33-65-...), 32 distinct data points are taken from one 1-second segment. Fourier transforms are done for each phase current, respectively. The frequency spectra are obtained using the fast Fourier transform (FFT) algorithm [28], implemented in the `numpy` and `scipy` packages. The real part of intensities are taken and converted to decibels (dB). The final amounts of data points are around 100,000 for the time domain and around 205,000 for the frequency domain.

For the experiments, we split the dataset into three parts: 50% for training, 20% for validation, and 30% for testing. The validation dataset is used to monitor the training process, and the loss function value thereon is used in learning rate adjustment and early stopping. The final accuracy and other metrics are evaluated on the test set. Figure 5 illustrates the preprocessing and split of datasets.

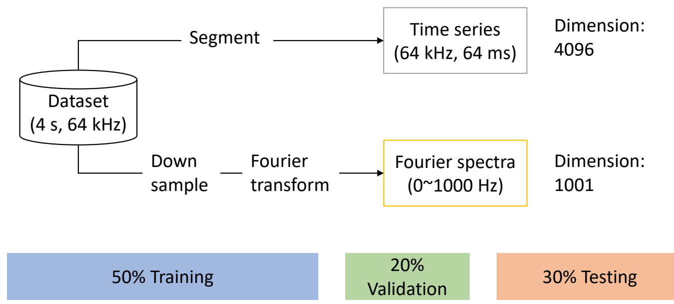

Fig. 5. Preprocessing and split of datasets.

We also used some other settings to mimic more realistic application scenarios. One is to train (and validate) on data of some operating conditions and test on other conditions. The other is to train on data of artificially created faults and test on faults from accelerated experiments. However, the results are not satisfactory. The main reason is that operating conditions (especially load) and the way faults are created may have a significant impact on the phase currents. Another potential reason is that deep learning models are sensitive to the distribution shift of data, they require the training and testing datasets to be retrieved from the same distribution. To extend the model to these scenarios, a practical way is to fine-tune the model using (a small amount of) data from the target operating conditions after pre-training the model.

B. Model Performances

We first train the model with SCL pretraining, as described in Section IV, using time-domain and frequency-domain data, respectively. As both the data splitting and model training processes contain stochasticity, we repeat training multiple times using different random settings and compare the average testing accuracies. Testing accuracy is defined as the fraction of correctly classified samples in the testing set. The average accuracy is around 87% using frequency data, and 77% using time series data, which shows the advantage of frequency spectra as input. Next, we choose frequency spectra as the input format and perform an ablation study. We train the model without SCL pretraining, i.e., directly train the whole model (Figure 4) by minimizing cross-entropy loss, keeping all other regularization and early-stopping settings. This is to assess how SCL pretraining benefits the model. The accuracy without SCL pretraining is 73%, which confirms that pretraining with SCL leads to better accuracy. Table II lists the results of these experiments.

As a benchmark study, we then train several conventional statistical ML models to classify frequency spectra into dif-

TABLE II
ACCURACIES OF MODELS TRAINED ON DIFFERENT INPUT FORMATS, USING DIFFERENT TRAINING SCHEMES.

Training Scheme	Input Format	Accuracy
With SCL	Time series	77%
	Frequency spectra	87%
Without SCL		73%

ferent health conditions, using the same way of defining training and testing sets. Models we test include logistic regression (linear classifier), support vector machine (SVM), tree ensembles, and k-nearest neighbors (kNN). All models are trained using the implementation of `scikit-learn` [29]. For SVM, the dimension of training data is first reduced using principal component analysis (PCA), and radial basis function (RBF) is used as the kernel function. The gradient boosting classifier model takes too long to train because of the data size, hence, it is excluded from the comparison. Accuracies of other models are reported in Table III. Out of the tested models, random forest attains the best accuracy, higher than a deep learning model without SCL pretraining. This is potentially related to the tabular nature of frequency spectra [30]. The SCL-trained deep learning model outperforms random forest, which again demonstrates the effectiveness of SCL.

TABLE III
ACCURACIES OF STATISTICAL ML MODELS TRAINED ON FREQUENCY-DOMAIN DATA.

Model	Accuracy	Note
Random forest	81%	Default settings
k-nearest neighbors	69%	$k = 10$
Logistic regression	68%	L_2 regularization
PCA–SVM	72%	RBF kernel

In addition, the SCL approach can be applied to any model architecture, such as the lightweight CNN used here, at negligible computational overhead, which is desirable for industrial applications.

C. Interpretation

Accuracy just provides an overall summary of model performances. To further investigate the performance difference between models and their origins, we show the confusion matrix in Figure 6. In these matrices, the diagonal values are percentages of correct classifications, and off-diagonal values reflect the misclassification. We can observe that the deep learning model without SCL tends to make two mistakes: (1) misclassify inner race fault as outer race fault, and (2) misclassify healthy as faults. The random forest model does not suffer as much from mistake (1), but it still tends to misclassify healthy as inner race fault. The SCL-based deep learning model reduces both of these two mistakes, which accounts for its better overall accuracy.

As is introduced in Section III, the power of contrastive learning is in extracting features that distinguish classes. For

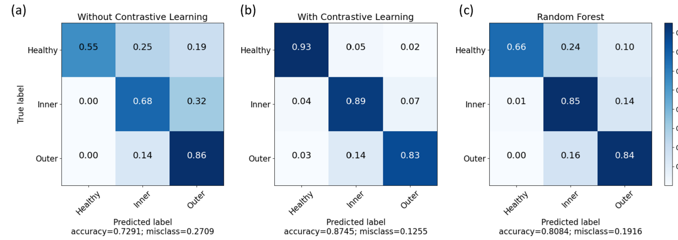

Fig. 6. Confusion matrices for (a) deep learning model without SCL, (b) deep learning model with SCL, and (c) random forest. In each matrix, the rows are true labels and the columns are predicted labels. The percentages are calculated based on the number of true labels (sum along rows), i.e., *recalls*.

the deep learning model, we consider the embeddings generated by the "feature extractor" as the features. To investigate the features, we use t-distributed stochastic neighbor embedding (t-SNE) to reduce them to 2 dimensions and visualize them in Figure 7. As the dataset is large (size $\sim 10^6$), only 1,000 samples are shown in each plot. In the top left part of Figure 7 (a), many green dots (inner race fault) overlap with purple and yellow ones (healthy), which indicates that inner race fault is not well distinguished from other two classes in the feature space learned by this model without SCL. Whereas in Figure 7 (b), green and purple dots are better separated. The yellow dots still overlap with others, which explains the relatively low recall (83%) of outer race fault. Yet, the feature space forms three clearer clusters, indicating that the distribution of features reflects the classes. This demonstrates that contrastive learning helps extract relevant features to help the classification of healthy and different fault types.

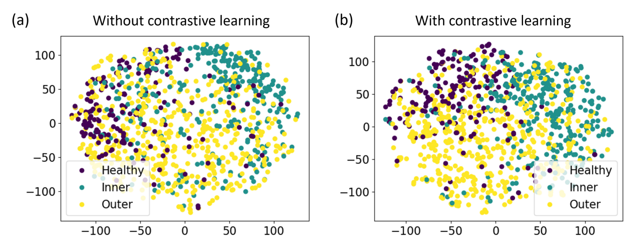

Fig. 7. Visualization of extracted features, (a) without SCL and (b) with SCL. Colors indicate different labels, i.e., health/fault conditions.

Finally, we discuss limitations and future work directions. Due to its supervised nature, the SCL approach requires labeled datasets, and its generalizability depends on data quality. Moreover, in-depth interpretation could help identify informative characteristics in the signal, thus offering insights for fault detection technique development.

VI. CONCLUSIONS

In conclusion, we investigated the effectiveness of supervised contrastive learning in developing data-driven models for bearing fault diagnosis. While most existing machine learning approaches are based on vibration signals, we develop the fault detection method using stator current signals, which are generally considered to be more challenging to classify. We applied a model architecture with a feature extractor pre-trained using supervised contrastive learning, and a classifier. We showed that our model not only achieves a higher fault classification accuracy than the model without contractive learning, but also outperforms conventional machine learning models. In future work, we will implement the method on other experimental datasets to further validate its effectiveness. As a general framework, we expect that our proposed fault detection method using contrastive learning can be generally applied to other fault detection tasks.

ACKNOWLEDGMENT

The authors thank Dr. Zhizhou Zhang and Dr. Chungwei Lin for helpful discussions.

REFERENCES

[1] Motor Reliability Working Group. Report of large motor reliability survey of industrial and commercial installations, part i. *IEEE Transactions on Industry Applications*, IA-21(4):853–864, 1985.

[2] Jinane Harmouche, Claude Delpha, and Demba Diallo. Improved fault diagnosis of ball bearings based on the global spectrum of vibration signals. *IEEE Transactions on Energy Conversion*, 30(1):376–383, 2015.

[3] Myeongsu Kang, Jaeyoung Kim, and Jong-Myon Kim. An fpga-based multicore system for real-time bearing fault diagnosis using ultrasampling rate ae signals. *IEEE Transactions on Industrial Electronics*, 62(4):2319–2329, 2015.

[4] M. El Hachemi Benbouzid. A review of induction motors signature analysis as a medium for faults detection. *IEEE Transactions on Industrial Electronics*, 47(5):984–993, 2000.

[5] A. Roque, J.M.F. Calado, and J.M. Ruiz. Vibration analysis versus current signature analysis. *IFAC Proceedings Volumes*, 45(20):794–799, 2012.

[6] T. Praveenkumar, Saimurugan.M Muthusamy, and Ramachandran K I. Comparison of vibration, sound and motor current signature analysis for detection of gear box faults. *International Journal of Prognostics and Health Management*, 8, 06 2017.

[7] R.R. Schoen, T.G. Habetler, F. Kamran, and R.G. Bartfield. Motor bearing damage detection using stator current monitoring. *IEEE Transactions on Industry Applications*, 31(6):1274–1279, 1995.

[8] Martin Blodt, Pierre Granjon, Bertrand Raison, and Gilles Rostaing. Models for bearing damage detection in induction motors using stator current monitoring. *IEEE Transactions on Industrial Electronics*, 55(4):1813–1822, 2008.

[9] Wei Zhou, Thomas G. Habetler, and Ronald G. Harley. Bearing fault detection via stator current noise cancellation and statistical control. *IEEE Transactions on Industrial Electronics*, 55(12):4260–4269, 2008.

[10] Shen Zhang, Bingnan Wang, Makoto Kanemaru, Chungwei Lin, Dehong Liu, Koon Hoo Teo, and Thomas G. Habetler. Quantification of rolling-element bearing fault severity of induction machines. In *International Electric Machines and Drives Conference (IEMDC)*, pages 44–50, 2019.

[11] Shen Zhang, Bingnan Wang, Makoto Kanemaru, Chungwei Lin, Dehong Liu, Masahito Miyoshi, Koon Hoo Teo, and Thomas G. Habetler. Model-based analysis and quantification of bearing faults in induction machines. *IEEE Transactions on Industry Applications*, 56(3):2158–2170, 2020.

[12] Shen Zhang, Shibo Zhang, Bingnan Wang, and Thomas G. Habetler. Deep learning algorithms for bearing fault diagnostics—a comprehensive review. *IEEE Access*, 8:29857–29881, 2020.

[13] Christian Lessmeier, James Kuria Kimotho, Detmar Zimmer, and Walter Sextro. Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. *PHM Society European Conference*, 3(1), 2016.

[14] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020.

[15] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Computer Vision and Pattern Recognition*, pages 2495–2504. IEEE, 2021.

[16] Yifei Ding, Jichao Zhuang, Peng Ding, and Minping Jia. Self-supervised pretraining via contrast learning for intelligent incipient fault detection of bearings. *Reliability Engineering & System Safety*, 218:108126, 2021.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[18] Katharina Rombach, Gabriel Michau, and Olga Fink. Contrastive learning for fault detection and diagnostics in the context of changing operating conditions and novel fault types. *Sensors*, 21(10):3550, 2021.

[19] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[20] Jiyang Zhang, Jianxiao Zou, Zhiheng Su, Jianxiong Tang, Yuhao Kang, Hongbing Xu, Zhiliang Liu, and Shicai Fan. A class-aware supervised contrastive learning framework for imbalanced fault diagnosis. *Knowledge-Based Systems*, 252:109437, 2022.

[21] Yongchao Zhang, Zhaohui Ren, Shihua Zhou, Ke Feng, Kun Yu, and Zheng Liu. Supervised contrastive learning-based domain adaptation network for intelligent unsupervised fault diagnosis of rolling bearing. *IEEE/ASME Transactions on Mechatronics*, 27(6):5371–5380, 2022.

[22] Ruihan Wang, Hui Chen, and Cong Guan. A self-supervised contrastive learning framework with the nearest neighbors matching for the fault diagnosis of marine machinery. *Ocean Engineering*, 270:113437, 2023.

[23] Chaoying Yang, Jie Liu, Qi Xu, and Kaibo Zhou. A generalized graph contrastive learning framework for few-shot machine fault diagnosis. *IEEE Transactions on Industrial Informatics*, pages 1–10, 2023.

[24] Chaoying Yang, Jie Liu, Kaibo Zhou, and Xingxing Jiang. Semisupervised machine fault diagnosis fusing unsupervised graph contrastive learning. *IEEE Transactions on Industrial Informatics*, 19(8):8644–8653, 2023.

[25] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. In *Neural Information Processing Systems*, 2022.

[26] Xi Chen, Cheng Ge, Ming Wang, and Jin Wang. Supervised contrastive few-shot learning for high-frequency time series. In *AAAI Conference on Artificial Intelligence*, volume 37, pages 7069–7077, 2023.

[27] Ilya Loshchilo and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[28] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

[29] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

[30] Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Neural Information Processing Systems*, volume 35, pages 507–520. Curran Associates, Inc., 2022.

BIBLIOGRAPHIES

Hengrui Zhang is a Ph.D. candidate in Mechanical Engineering at Northwestern University, USA. He received B.S. in Materials Science and Engineering from Shanghai Jiao Tong University, China in 2020. He has a broad interest in AI for Science, with a focus on materials discovery and design.

Bingnan Wang (M'12-SM'15) received his B.S. degree from Fudan University, Shanghai, China, in 2003, and Ph.D. degree from Iowa State University, Ames, IA, USA, in 2009, both in Physics. He has been with Mitsubishi Electric Research Laboratories (MERL), located in Cambridge, Massachusetts since then, and is now a Senior Principal Research Scientist. His current research focuses on electric machine analysis, design optimization, fault diagnosis with both physics modeling and machine learning techniques.