UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Rodrigo Kobashikawa Rosa

A Multilabel Approach to Machine Learning-Based Bearing Fault Diagnosis using the CWRU Dataset

Florianópolis
2024

Rodrigo Kobashikawa Rosa

A Multilabel Approach to Machine Learning-Based Bearing Fault Diagnosis using the CWRU Dataset

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica para a obtenção do título de mestre em Engenharia Elétrica.
Orientador: Prof. Danilo Silva, PhD

Florianópolis
2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Rosa, Rodrigo Kobashikawa
A Multilabel Approach to Machine Learning-Based Bearing Fault Diagnosis using the CWRU Dataset / Rodrigo Kobashikawa Rosa ; orientador, Danilo Silva, 2024.
69 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Elétrica, Florianópolis, 2024.

Inclui referências.

1. Engenharia Elétrica. 2. Bearing fault diagnosis. 3. Multi-label classification. 4. Deep neural networks. 5. Data leakage. I. Silva, Danilo. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

Rodrigo Kobashikawa Rosa

A Multilabel Approach to Machine Learning-Based Bearing Fault Diagnosis using the CWRU Dataset

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Júlio Apolinário Cordioli, Dr.
Universidade Federal de Santa Catarina

Prof. Márcio Holsbach Costa, Dr.
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Engenharia Elétrica.

Documento assinado digitalmente
Telles Brunelli Lazzarin
Data: 25/11/2024 15:54:58-0300
CPF: ***.400.719-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Telles Brunelli Lazzarin, Dr
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Documento assinado digitalmente
Danilo Silva
Data: 25/11/2024 12:31:15-0300
CPF: ***.557.924-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Danilo Silva, PhD
Orientador

Florianópolis, 2024.

Esse trabalho é dedicado aos meus queridos pais.

ACKNOWLEDGEMENTS

Agradeço profundamente aos meus pais pelo apoio incondicional em todas as etapas da minha vida. Eles são meus dois grandes pilares, que estiveram ao meu lado em todas as conquistas e nos momentos mais difíceis.

Agradeço também à minha namorada, que esteve ao meu lado durante todo o meu mestrado, acompanhando e apoiando tanto nos momentos de alegria quanto nos desafios que enfrentei ao longo desses anos.

Gostaria de expressar minha gratidão aos amigos e colegas de projeto, com quem compartilhei momentos agradáveis de conversas, trocas de conhecimento e apoio mútuo em todas as etapas do trabalho. Acredito sinceramente que todos se tornaram grandes profissionais.

Agradeço de coração ao meu orientador, Danilo Silva, pela confiança, pelas oportunidades e pelos ensinamentos que me proporcionou desde a minha graduação até o mestrado. Muito do que sou hoje, tanto profissionalmente quanto pessoalmente, devo aos seus conselhos e orientações.

Por fim, mas não menos importante, gostaria de agradecer à UFSC pelo ensino de excelência, e à Dynamox, juntamente com a FEESC, pelo apoio a esta pesquisa. Agradeço também ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), que apoiou meu mestrado por meio do processo número 164299/2021-1.

People think of education as something they can finish.
- Isaac Asimov

RESUMO

Nos ambientes industriais, a operação confiável de máquinas rotativas é extremamente importante para diversos processos. Qualquer falha não detectada pode levar a paradas de máquinas, perdas financeiras e riscos à segurança. Os rolamentos são um dos componentes mais comuns e também a causa mais frequente de falhas em máquinas rotativas. Portanto, dada a sua importância, a detecção e o diagnóstico precoce de falhas em rolamentos têm sido amplamente estudados no contexto da manutenção preditiva. Este trabalho propõe uma nova abordagem para modelar o problema do diagnóstico de falhas em rolamentos utilizando aprendizado de máquina, empregando o conjunto de dados de falhas em rolamentos da Case Western Reserve University (CWRU). Apesar do conjunto de dados ser considerado uma referência padrão para testar novos algoritmos, a divisão típica do conjunto de dados sofre de vazamento de dados, conforme mostrado por Hendriks et al. (2022) e Abburi et al. (2023), levando a artigos reportarem resultados excessivamente otimistas. Embora a divisão proposta por eles mitigue significativamente esse problema, ela não o elimina completamente. Além disso, a tarefa de classificação multiclasse proposta por eles ainda pode levar a um cenário irrealista, ao excluir a possibilidade de mais de um tipo de falha ocorrer ao mesmo tempo ou em diferentes localizações. Conforme defendido neste trabalho, uma formulação multirrótulo (detectando a presença de cada tipo de falha em cada localização) pode resolver ambos os problemas, levando a um cenário mais próximo da realidade. Além disso, essa abordagem mitiga o pesado desequilíbrio de classes no conjunto de dados da CWRU, onde casos de falha aparecem com muito mais frequência do que casos saudáveis, embora o oposto seja mais provável na prática. Uma formulação multirrótulo também permite uma avaliação mais precisa utilizando métricas independentes de prevalência para classificação binária, como a curva ROC. Finalmente, este trabalho propõe uma divisão mais realista do conjunto de dados, que permite maior diversidade no conjunto de treinamento, mantendo a divisão livre de vazamento de dados. Os resultados mostram que essa nova divisão pode melhorar significativamente o desempenho, ao mesmo tempo em que possibilita uma análise de erro mais detalhada. Como aplicação da nossa abordagem, é realizada uma análise comparativa utilizando vários modelos de aprendizado profundo do estado da arte aplicados a representações de sinais 1D e 2D nos domínios do tempo e/ou frequência.

Palavras-chave: Diagnóstico de falhas de rolamento. Classificação multirrótulo. Análise comparativa. Redes Neurais Profundas. Vazamento de dados.

RESUMO ESTENDIDO

Introdução

A detecção e o diagnóstico de falhas mecânicas em máquinas rotativas são essenciais para garantir a continuidade dos processos produtivos e reduzir custos operacionais. Rolamentos de elementos rolantes são componentes críticos e frequentemente a principal fonte de falhas. Sensores de vibração oferecem uma abordagem eficaz para a detecção de falhas, mas a análise manual dos dados não é escalável. Assim, técnicas automatizadas baseadas em aprendizado de máquina, especialmente no aprendizado profundo, têm sido amplamente empregadas para melhorar a eficiência na identificação precoce de falhas.

O conjunto de dados CWRU tem sido amplamente utilizado como referência para avaliar novas arquiteturas de redes neurais para detecção e diagnóstico de falhas. Ele inclui sinais de vibração coletados de rolamentos saudáveis e defeituosos sob diferentes condições de carga de motor. No entanto, o conjunto de dados apresenta um desbalanceamento entre condições saudáveis e defeituosas, contendo muito mais amostras defeituosas (na realidade também existe um desbalanceamento, porém ao contrário, com mais amostras saudáveis). Além disso, muitos estudos utilizando-o com modelos de aprendizado de máquina apresentam problemas de vazamento de dados, onde o mesmo rolamento defeituoso é utilizado durante o treinamento e o teste, comprometendo a capacidade dos modelos de generalizar para novos dados e trazendo resultados sobre-otimistas.

Embora divisões alternativas do conjunto de dados tenham sido propostas para mitigar o vazamento de dados, limitações ainda existem, como a presença de uma única configuração saudável e a incapacidade de lidar com múltiplas falhas simultâneas. Além disso, o uso de acelerômetros sincronizados, como proposto em alguns estudos, é improvável em cenários industriais reais. Assim, a aplicabilidade prática dessas técnicas ainda enfrenta desafios, exigindo aprimoramentos metodológicos para garantir a generalização e robustez dos modelos em condições reais.

Este trabalho apresenta uma nova abordagem para a detecção de falhas mecânicas em rolamentos por meio da classificação multirrótulo, onde o modelo identifica a presença ou ausência de cada uma das falhas (trilha interna, trilha externa e do elemento rolante) nos lados do ventilador (*fan end*) e do acionamento (*drive end*) do motor. Essa estratégia acaba com o problema de vazamento de dados da configuração saudável, garantindo que sinais da configuração de rolamentos saudáveis estejam apenas no conjunto de teste, e possibilita a representação realista de cenários com múltiplos tipos de falhas. A abordagem também mitiga o desequilíbrio de classes, permitindo que cada classe negativa inclua outros tipos de falhas, resultando em avaliações mais precisas usando métricas independentes da prevalência, como a curva ROC e a área sob a curva (AUROC).

Objetivos

Esta dissertação tem como objetivo desenvolver um sistema robusto de detecção e diagnóstico de falhas baseado em aprendizado de máquina, proporcionando uma avaliação realista e confiável. O objetivo principal é melhorar o desempenho do modelo e permitir a comparação entre diferentes arquiteturas e características de entrada.

Os objetivos específicos são:

1. Propor uma abordagem para resolver o problema de vazamento de dados;
2. Apresentar uma nova divisão do conjunto de dados CWRU para aumentar a diversidade de informações dos rolamentos no treinamento;
3. Fornecer um procedimento de otimização de hiperparâmetros;
4. Explorar diferentes abordagens de treinamento para melhorar o desempenho;
5. Realizar uma análise de erros nas previsões do modelo;
6. Conduzir uma comparação do desempenho de várias arquiteturas e representações de sinais.

Metodologia

Este trabalho utilizou o conjunto de dados de falhas em rolamentos CWRU, que inclui experimentos com rolamentos saudáveis e vários rolamentos com falhas induzidas artificialmente. As falhas foram criadas por meio de usinagem por descarga eletrostática, resultando em falhas pontuais de vários diâmetros na trilha interna, trilha externa e elementos rolantes. Os dados foram coletados de forma síncrona a partir de dois acelerômetros em duas posições do motor, sob diferentes níveis de carga e para diferentes tamanhos de defeito.

O problema da detecção e diagnóstico de falhas em rolamentos foi abordado por meio de técnicas de classificação multirrótulo. Isso envolveu a construção de detectores específicos para cada tipo de falha, permitindo a verificação independente de múltiplos tipos de falhas simultaneamente. Ao tratar o diagnóstico de falhas no *drive end* e *fan end* como problemas independentes, a metodologia permitiu o uso eficaz dos sinais das duas localidades para aprimorar a precisão da classificação. As amostras foram rotuladas positivamente para um determinado tipo de falha com base na localização específica onde o sinal foi adquirido, resultando em uma representação mais realista das ocorrências de falhas.

Para evitar a contaminação dos dados e aumentar a eficácia do treinamento do modelo, foi proposta uma nova divisão do conjunto de dados. Rolamentos defeituosos de diferentes tamanhos foram agrupados (contanto que o mesmo rolamento não estivesse presente nos treino e teste), permitindo configurações aleatórias de tipos de falhas, localizações e cargas entre os subconjuntos de treinamento e teste. Isso garantiu que os modelos fossem treinados em uma variedade de condições, refletindo cenários do mundo real em que múltiplos tipos e tamanhos de falhas poderiam ocorrer simultaneamente.

Para a seleção do modelo e avaliação de desempenho, foi empregado o Método de Validação Cruzada Dupla (CVM-CV). Essa abordagem envolveu otimização de hiperparâmetros em uma primeira etapa de validação cruzada, seguida da reavaliação em diferentes divisões de treinamento-teste, minimizando o viés na estimativa de desempenho. A arquitetura ResNet foi utilizada como principal modelo devido à sua eficácia em aplicações similares, com vários hiperparâmetros otimizados por meio de busca em grade para aprimorar o desempenho do modelo. Além disso, foram realizados experimentos com várias arquiteturas do estado da arte de visão computacional e para detecção de falhas foram exploradas e também com diferentes representações de sinais de entrada, permitindo uma avaliação abrangente da eficácia do modelo no diagnóstico de falhas em rolamentos.

Resultados e Discussão
Utilizando o modelo ResNet18 com a metodologia proposta, foi obtido uma métrica AUROC de 0,911$\pm$0,038. A análise de erros revelou que o modelo teve dificuldades para diferenciar falhas internas e externas, assim como para detectar certas condições específicas de falhas. Essas dificuldades, no entanto, podem ser explicadas por características do conjunto de dados CWRU, como já descrito em estudos anteriores a respeito do conjunto de dados, o que confere maior credibilidade aos resultados obtidos.

Para avaliar a nossa metodologia, foram preparados diversos experimentos de remoção. Exploramos a remoção de diversas abordagens de treinamento da nossa metodologia, incluindo a contribuição da nova divisão do conjunto de dados, a expansão da proporção de treino/teste de 1:2 para 2:1 e o aumento da diversidade das configurações de falhas. Além disso, dividimos a detecção de falhas por tipo e localização em dois problemas distintos, utilizando um único modelo, o que dobrou o conjunto de treinamento. Com os experimentos, observou-se que nossa metodologia fez com que o AUROC macro médio aumentasse de 0,781 para 0,911, demonstrando a eficácia das abordagens propostas.

Realizamos também um estudo comparativo com diferentes arquiteturas de modelos e representações de sinais. Nenhuma arquitetura superou os resultados da ResNet aplicada a imagens de espectrograma. No entanto, o WDCNN aplicado ao cepstrum de potência apresentou resultados competitivos, especialmente considerando seus menores requisitos computacionais. Adicionalmente, observamos que muitos trabalhos anteriores não especificam seus procedimentos de otimização de hiperparâmetros, o que pode levar a avaliações enviesadas. Nosso estudo contribui para mitigar esse viés, propondo um procedimento rigoroso de seleção e avaliação de modelos.

Considerações Finais
O conjunto de dados CWRU continua sendo um recurso fundamental para a pesquisa em detecção e diagnóstico de falhas em rolamentos. Recomendamos que futuros trabalhos que utilizem esse conjunto de dados adotem a metodologia proposta neste estudo, pois ela oferece uma avaliação mais realista e confiável dos modelos de aprendizado de máquina, minimizando problemas de vazamento de dados e oferecendo uma base sólida para o desenvolvimento de soluções aplicáveis em cenários industriais.

Palavras-chave: Diagnóstico de falhas de rolamento. Classificação multirrótulo. Análise comparativa. Redes Neurais Profundas. Vazamento de dados.

ABSTRACT

In today's industrial environments, the reliable operation of rotating machinery is extremely important for various processes. Any unnoticed failure can lead to machine downtime, financial loss, and safety hazards. Rolling bearings are one of the most common components and also the most frequent cause of failure in rotating machinery. Therefore, given their importance, the early detection and diagnosis of rolling bearing faults have been widely studied in the context of predictive maintenance. This work proposes a novel approach for modeling the problem of rolling bearing fault diagnosis with machine learning using the Case Western Reserve University (CWRU) bearing fault dataset. Although the dataset is considered a standard reference for testing new algorithms, the typical dataset division suffers from data leakage, as shown by Hendriks et al. (2022) and Abburi et al. (2023), leading to papers reporting over-optimistic results. While their proposed division significantly mitigates this issue, it does not eliminate it entirely. Moreover, their proposed multi-class classification task can still lead to an unrealistic scenario by excluding the possibility of more than one fault type occurring at the same or different locations. As advocated in this paper, a multi-label formulation (detecting the presence of each type of fault for each location) can solve both issues, leading to a scenario closer to reality. Additionally, this approach mitigates the heavy class imbalance of the CWRU dataset, where faulty cases appear much more frequently than healthy cases, even though the opposite is more likely to occur in practice. A multi-label formulation also enables a more precise evaluation using prevalence-independent metrics for binary classification, such as the ROC curve. Finally, this paper proposes a more realistic dataset division that allows for more diversity in the training dataset while keeping the division free from data leakage. The results show that this new division can significantly improve performance while enabling a fine-grained error analysis. As an application of our approach, a comparative benchmark is performed using several state-of-the-art deep learning models applied to 1D and 2D signal representations in time and/or frequency domains.

Keywords: Bearing fault diagnosis. Multi-label classification. Benchmarking. Deep neural networks. Data leakage.

LIST OF TABLES

LIST OF FIGURES

CONTENTS

1 INTRODUCTION

Detecting and diagnosing mechanical failures in industrial machines is crucial for maintaining a continuous production process. Within the domain of rotating machinery, rolling element bearings are numerous, representing most of the bearings while being critical components and often the primary source of mechanical failures (ZHANG et al., 2020). Therefore, adopting automated techniques is strategic for early fault detection, enabling maintenance scheduling and reducing downtime and operational costs.

In this sense, vibration sensors offer a promising data-driven approach to detecting bearing faults, significantly aiding in the precise and timely identification of potential issues in rotating machinery. Nevertheless, with a large number of machines monitored by vibration sensors, manual analysis by specialists is not scalable. Alternatively, quick analysis of large data volumes is achievable through automated fault analysis techniques, ensuring potential problems are detected and addressed promptly. Among these techniques, machine learning methods have been successfully applied (LEI et al., 2020; ZHANG; LI; DING, 2019), with deep learning algorithms (NEUPANE; SEOK, 2020) achieving state-of-the-art performance. However, employing and benchmarking these techniques requires a careful methodology (KAPOOR; NARAYANAN, 2023), problem formulation, and dataset preparation to ensure they generalize to real-world scenarios.

The CWRU bearing fault dataset has become a widely used benchmark dataset for new network architectures (NEUPANE; SEOK, 2020). It comprises vibration signals collected during experiments on a test bench featuring rolling bearings within an electric motor. The dataset encompasses bearings with single-point faults of varying sizes in the inner race, outer race, and rolling element (ball), as well as healthy bearings. Measurements were obtained from both the drive end and fan end locations, with healthy and faulty bearings installed under four operational motor load conditions for each fault type. As a result, the experiments feature several bearing configurations for faulty bearings and just one for healthy bearings. This disproportion leads to a heavy skew towards abnormality in the dataset, which is a somewhat unrealistic reflection of the more prevalent healthy condition observed in real-world scenarios.

While abundant in the literature, it is unclear if fault classification results obtained with the CWRU dataset using machine learning models truly generalize to different settings. The typical CWRU dataset division seen in the literature has the segments from the same signal being randomly split between train and test sets. This creates a situation known as *data leakage* (KAPOOR; NARAYANAN, 2023) that can cause the machine learning model to memorize a signature of each specific signal, compromising the model's generalization to new unseen signals. It is especially troublesome for deep learning models due to their capacity to easily overfit, resulting in an over-optimistic evaluation that would fail to be reproduced in practice. Another typical CWRU division, mostly used in the context of domain adaptation, is the division of the dataset by motor load. In this division, the aforementioned segment-level data leakage is no longer present because all segments from the same signal remain together in either the train or

the test set. However, both Hendriks, Dumond & Knox (2022) and Abburi et al. (2023) identified another form of data leakage present in this division. In the CWRU dataset, the same faulty bearings are reused for experiments at different loads so that multiple measurements are made for each bearing. As a result, if the train and test datasets are divided by load, measurements at different loads are created from the same faulty bearings, resulting in the same bearing appearing in both the train and test datasets. This situation is also a form of data leakage, as the bearing information is leaked between train and test sets. Indeed, both papers show that it leads to over-optimistic results compared to divisions where this leakage does not occur.

Although their methodology has effectively solved most of the data leakage problem and presented a more practical scenario than previous studies, some disadvantages remain. The multi-class formulation present in both works comes with several limitations, the most prominent being its inability to operate entirely without data leakage due to the presence of only one healthy bearing configuration in the dataset, making it impossible to split the dataset and prevent measurements from the same healthy bearing configuration (experiment with only healthy bearings at the drive and fan end) from appearing in both train and test datasets. Additionally, since classes are defined based on the CWRU bearing configurations, where, at most, a single fault is present on each configuration, the model cannot account for multiple faults happening at once. Moreover, the highly imbalanced healthy class renders accuracy an unrealistic evaluation metric. Finally, in the Hendriks, Dumond & Knox (2022) paper, the authors rely on the use of two synchronous accelerometers to identify the fault location, which is an unlikely assumption for real-case scenarios and unnecessary for fault detection and diagnosis.

1.1 OBJECTIVES

This dissertation aims to develop a robust machine learning-based fault detection and diagnosis system, providing a realistic and reliable evaluation framework with the ultimate goal of enhancing the model's performance and enabling a comparison of multiple model architectures and input features.

The specific objectives of this study are:

1. Propose a problem formulation approach to solve the data leakage problem;
2. Present a new dataset division on the CWRU bearing dataset to increase bearing information diversity in the training set;
3. Provide a hyperparameter optimization procedure under the proposed framework;
4. Explore different training approaches to enhance model performance;
5. Generate insights through error analysis of the model's predictions;
6. Conduct a comparative benchmark using several model architectures and signal representations.

1.2 CONTRIBUTIONS

In this work, we propose a novel approach for modeling the problem using binary multi-label classification. Specifically, for each location (fan end and drive end), a model must detect the presence or absence of each type of fault (inner, outer and ball). With this approach, it is possible to remove the data leakage issue by ensuring that all signals from the healthy bearing configuration appear only in the test set while also leading to a more realistic scenario by having the possibility of more than one fault type occurring at each motor bearing. The multi-label approach also mitigates the healthy bearing class imbalance by transforming the problem into a separate binary classification for each of the three fault types, making the negative class of each fault-type detector comprise the other fault types instead of only the healthy class. Consequently, the approach enables a more precise evaluation by using prevalence-independent metrics such as the Receiver-operating characteristic (ROC) curve and the Area Under the ROC curve (AUROC). Moreover, by transforming the problem into two separate problems (one at each possible fault location) and, for each problem, evaluating only a single signal (the signal acquired at that specific location), it is possible to eliminate the need for synchronous signals. Lastly, with this multi-label approach, a different dataset division is made possible that maximizes the diversity of fault types and sizes during training and testing while keeping it without data leakage.

Experiments under our proposed methodology were conducted on the ResNet18 architecture applied to raw spectrogram images. A rigorous performance evaluation is presented, along with an error analysis of the model's outputs, as well as ablation experiments to evaluate the gains of individual components of our proposed training approach and dataset division. Additionally, we conducted a comparative benchmark using several state-of-the-art deep learning models applied to different signal representations to identify the most suitable architecture. Finally, we also consider a simpler problem of detecting when a fault (of any type) is present at a specific location, for which performance can be further improved.

1.3 RELATED WORK

To address the bearing information from leaking between splits, Hendriks et al. (HENDRIKS; DUMOND; KNOX, 2022) proposed a new dataset split. Using the same idea of the typical division where the sets are split by load, the authors noted that if the split was made by fault size, the bearing information leakage no longer occurred for the faulty bearing measurements. Considering the problem of fault classification into seven classes (healthy state plus inner/outer/ball faults at either fan or drive end), their proposed approach significantly reduced the inflated performance metric of state-of-the-art deep convolutional networks, which achieved 95% classification accuracy in the division by load and now achieve a more realistic performance of 53% accuracy in the division by fault size.

Abburi et al. (ABBURI et al., 2023), on the other hand, proposed a different dataset

split to mitigate bearing information from leaking. In their proposal division, they assign only drive-end fault measurements for the training set, fan-end fault measurements of sizes 7 and 14 mils for the validation set, and fan-end measurements of size 21 mils for the test sets. The experiments used traditional machine learning models, such as Random Forest, Naive Bayes, and Support Vector Machine, on a multi-class problem formulation with three fault classes (inner race, outer race, ball) and the healthy condition. The results have shown that the bearing split had worse results across all metrics reported (accuracy, precision, macro F1-score, recall) when compared to the random split where the bearing information leakage occurred. For the Naive Bayes model, the accuracy dropped from 85.8% to 69.5%.

However, as mentioned earlier, none of these multi-class formulations are able to avoid bearing data leakage from the healthy class, which we avoid using a multi-label approach.

Although a few previous works using the CWRU dataset for fault detection and diagnosis have also adopted a multi-label approach, they all employ different formulations that ignore the specific problems addressed in this dissertation. Shen et al. (SHEN et al., 2020) consider two multi-class labels representing fault type and fault size in the CWRU dataset. Since a single label is used for fault type diagnosis, this is still a multi-class problem, where fault types are assumed to be mutually exclusive. Similarly, Yu et al. (YU et al., 2021) consider three multi-class labels corresponding to fault size, fault type and motor speed, except that the healthy condition is ignored. Again this is a multi-class problem for fault type diagnosis, while the fault detection problem is not treated. The closest to our formulation are those of Chen et al. (CHEN et al., 2019) and Jin et al. (JIN et al., 2021), which consider binary labels for the three fault types plus the healthy state (as well as labels for each fault type and size combination in Chen et al. (2019)). However, the inclusion of a healthy state label requires using healthy data for training (besides testing), which our approach does not require. Moreover, these works only consider signals acquired at a single location (either drive end or fan end) corresponding to faults occurring at that same location; thus, the fault localization problem is not treated.

More importantly, these works differ from ours in their division of the dataset into train and test splits, which fails to avoid data leakage (ABBURI et al., 2023), either at the segmentation level or at the bearing level. These leakages predominantly cause overoptimistic results with nearly 100% accuracy, and therefore, their performance cannot be properly evaluated. Another point not present in any of these studies is the use of the fan end signals as negative samples when diagnosing faults at the drive end and vice-versa. To the best of our knowledge, our work is the only study that has applied a multi-label approach to the CWRU dataset to address problems such as bearing data leakage and healthy class imbalance, ultimately transforming the problem formulation into one that is better aligned with real-world conditions while ensuring proper evaluation of the model.

1.4 DISSERTATION STRUCTURE

The remainder of this dissertation is organized as follows: Chapter 2 provides a theoretical background around rolling bearings and machine learning, focusing on the important topics needed to understand the presented methods. Afterward, Chapter 3 introduces and discusses the proposed methodology. The experiments and results are presented in Chapter 4, along with ablations and error analysis sections. Finally, Chapter 5 discusses the results obtained and concludes the dissertation.

2 THEORETICAL BACKGROUND

This chapter aims to introduce and revisit the main concepts used throughout this study regarding vibration-based condition monitoring, fault signals on rolling bearings and machine learning concepts.

2.1 VIBRATION-BASED CONDITION MONITORING

In industrial settings, it is critical to ensure the reliability and efficiency of machinery. Originally, machines were made to 'run to break' to maximize the operating time between shutdowns. However, this also means that fault breakdowns can occasionally be catastrophic, with serious safety risks, causing production loss due to production line stoppage and a high repair cost. For those causes, preventive maintenance was introduced as scheduled maintenance intervals shorter than the expected 'time between failures' with a minimal likelihood of failure between repairs. While catastrophic failures are significantly reduced, the disadvantages come as most machines could run longer for a factor of 2-3 times higher, causing too much maintenance to be carried out and resulting in an excessive number of replacement components consumed (RANDALL, 2021).

Predictive maintenance strategies, such as condition-based monitoring, solve those problems by regularly monitoring the machines' condition to detect early signs of degradation or impending failure, allowing maintenance to be performed at the optimum time. In condition-based monitoring, vibration analysis is one of the main ways of obtaining information on working machinery in production, having many advantages compared to other methods. It reacts immediately to change, can be used for permanent and intermittent monitoring, can point to the faulty component more easily than other techniques, and many powerful signal processing techniques can be applied to vibration signals (RANDALL, 2021). With the advent of Industry 4.0, (LASI et al., 2014), the use of sensors connected to the internet on machines of all industries increased drastically, including accelerometer sensors actively collecting vibration data for permanent monitoring of assets. The enormous amount of data collected made developing various machine learning models possible.

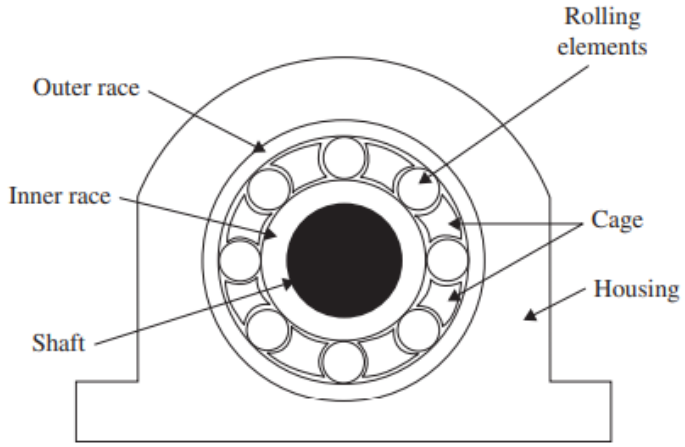
2.2 ROLLING ELEMENT BEARINGS

Rolling bearings maintain the motion between static and moving parts in a rotating machine. They are among the most used elements in machines, and if we analyze the most frequent reasons for machine problems, they rank high as one of the most frequent causes of failure. For this reason, rolling bearing failures have been widely studied, and several diagnostic methods using signal processing or data-driven techniques have been developed.

Rolling bearings are composed of four main components. These include the inner race, where the shaft drives; the outer race, typically positioned in a hole or housing; the rolling

elements, commonly placed between the inner and outer races; and the cage, which is used to keep the rolling elements separated equally (AHMED; NANDI, 2020). Figure 1 illustrates a typical rolling bearing and its components.

Figure 1 – A typical roller bearing.



Source: Ahmed & Nandi (2020)

Defects on bearings may occur for many reasons, such as fatigue, incorrect lubrication, contamination and corrosion, and incorrect bearing installation (HARRIS, 2001; NANDI; TOLIYAT; LI, 2005). When a fault happens in a rolling bearing, a series of periodically repeated impulses are generated at the bearing fundamental fault frequency. Depending on the damaged part, a different frequency can be used to diagnose the fault. For the faults in the outer race, the ball pass frequency at the outer race (BPFO) is used; for the inner race faults, the ball pass frequency at the inner race (BPFI); for the rolling element faults, the ball spin frequency (BSF); and finally, for the cage faults, it is used the fundamental train frequency (FTF) (RANDALL, 2021). The equations for each frequency are given by:

$$\text{BPFO} = \frac{nf_r}{2}\left(1 - \frac{d}{D}cos\phi\right), \tag{2.1}$$

$$\text{BPFI} = \frac{nf_r}{2}\left(1 + \frac{d}{D}cos\phi\right), \tag{2.2}$$

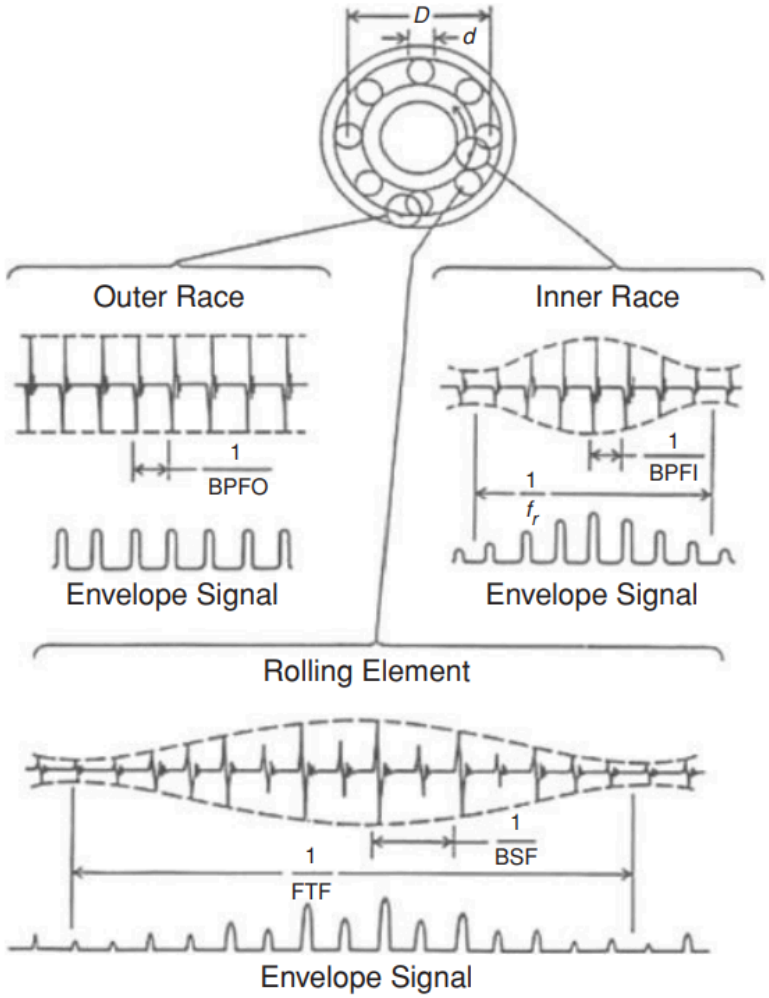
$$\text{BSF} = \frac{f_r}{2}\left(1 - \frac{d}{D}cos\phi\right), \tag{2.3}$$

$$\text{FTF} = \frac{D}{2d}\left[1 - \left(\frac{d}{D}cos\phi\right)^2\right], \tag{2.4}$$

where f_r is the shaft speed, n is the number of rolling elements, and ϕ is the angle of the load from the radial plane.

For many years, envelope analysis has been recognized as the standard method for bearing diagnostics. This technique involves bandpass filtering a signal in a high-frequency band, in which the fault impulses are amplified by the structural resonances. The filtered signal is amplitude demodulated to create the envelope signal. Then, the spectrum is taken, which contains diagnostic information related to the repetition frequency (ball pass frequency or ball spin frequency) and the modulation of the frequencies that the fault is passing through the load zone or moving with respect to the measurement point (RANDALL, 2021). Examples of the signals of each component along the envelope signal can be seen in Figure 2.

Figure 2 – Typical signals and envelope signals from local faults in rolling element bearings.



Source: Randall (2021)

The cepstrum is another signal processing technique that may be applied to bearing fault diagnosis. It was originally proposed by Bogert (1963) and defined as the power spectrum of the logarithm of the power spectrum,

$$C_p = \left| \mathscr{F} \left\{ log \left(\left| \mathscr{F} f(t) \right|^2 \right) \right\} \right|^2 \tag{2.5}$$

also called as a *spectrum of a spectrum* due to it involving the Fourier transform of a spectrum.

Cepstrum analysis may be used especially for signals whose faults contain families of harmonics and sidebands, such as gear and bearing faults, as it is able to represent the spacings in the spectrum. However, there is a caveat. In the case of bearing faults, discrete harmonics of the fault frequencies are not typical for slow-speed machines, only on high-speed machines where resonances excited by the fault represent a relatively low harmonic order of the ball pass frequencies involved (RANDALL, 2021). For this reason, envelope analysis tends to be the preferred technique, given that it works well in both cases.

2.3 MACHINE LEARNING

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that utilizes statistical algorithms that can learn from data. A popular definition of what is ML goes as follows:

> A computer program is said to learn from experience E with respect to some class of tasks T, and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. (MITCHELL, 1997)

From this definition, there are many possible *experiences* (E), *tasks* (T), and *performance measures* (P) that can be used to describe all the possibilities within the machine learning scope. A brief description and examples will be presented to understand better what they can be and how they connect to a machine learning algorithm.

2.3.1 Supervised learning

Machine learning algorithms can be categorized as supervised, unsupervised, or reinforcement learning depending on what kind of *experience* they have during the learning process.

Supervised learning is the most common form of ML, where the learning algorithm is given a *task* to learn a mapping f from inputs $\boldsymbol{x} \in \boldsymbol{X}$ to outputs $\boldsymbol{y} \in \boldsymbol{Y}$. The inputs $\boldsymbol{x}$—also called features—are often a fixed-dimensional vector of numbers or categories, the output y is known as label or target, and when presented as a set of n input-output pairs, they are known as the training set (the *experience* given to the task to learn). The *performance measure* depends on the type of output we want to predict (MURPHY, 2022).

The term "supervised learning" originates from the view of the target output $\boldsymbol{y}$ being provided by a teacher who teaches the machine learning system what it has to do (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.3.2 Classification

In a classification problem, there is a set of C discrete number of classes $\boldsymbol{Y} = \{1, 2, ..., C\}$ where the algorithm's goal is to train a function that maps each input $\boldsymbol{x}$ into one of the classes.

An example of a classification task is object recognition, where the input is an image (usually described as a set of pixel brightness values), and the output is a numeric code identifying the object in the image (GOODFELLOW; BENGIO; COURVILLE, 2016).

There are three main types of classification tasks:

- Binary classification: classifying instances into one of two classes;
- Multi-class classification: classifying instances into one of three or more classes;
- Multi-label classification: variant of multi-class classification where multiple nonexclusive labels can be assigned to each instance.

In the context of this work, the multi-label classification task is to correctly associate the vibration signal to the appropriate bearing fault class labels.

2.3.3 Evaluation metrics

A quantitative measure of its performance is needed to evaluate a machine learning algorithm's abilities. However, selecting a proper performance measure depends highly on the task being executed and must be carefully chosen. For tasks such as classification, it is common to see the use of accuracy or some sort of error rate quantification metric (GOODFELLOW; BENGIO; COURVILLE, 2016). An example of such a quantitative measure of error rate is the confusion matrix.

The confusion matrix, also known as the error matrix, is a table that allows visualization of the model's performance. It is usually represented as the columns being the samples in a predicted class and the rows being the instances in an actual class. The diagonal of the matrix represents all instances correctly classified, while the other cells in the table represent the incorrect predictions.

In a binary classification task, a confusion matrix is represented as shown in Table 1. True Positives (TP) indicate the number of correct predictions for the positive class, while True Negatives (TN) represent the correctly predicted negative instances. On the other hand, False Positives (FP) and False Negatives (FN) indicate when the model confuses the classes. False Positives occur when negative-class instances are incorrectly identified as positive instances, whereas False Negatives occur when actual positive instances are incorrectly identified as negative instances.

Table 1 – Confusion matrix example for the binary classification case.

		Predicted condition	
		Positive	Negative
Target condition	Positive	TP	FN
	Negative	FP	FN

Source: Author, 2024

Please bear in mind that the confusion matrix can be used not only for binary classification but also for multi-class classification problems. In this case, as shown in Table 2, the diagonal of the table shows the correctly predicted positive outcomes for each class, while the other cells show the false positives, false negatives, and true negatives based on the chosen class for analysis.

Table 2 – Confusion matrix example for the multi-class classification case.

		Predicted condition			
		Class 1	Class 2	Class 3	Class4
Actual condition	Class 1	TP			
	Class 2		TP		
	Class 3			TP	
	Class 4				TP

Source: Author, 2024

A confusion matrix can also yield other metrics by using the Positive instances (P), Negative instances (N), TP, TN, FP, and FN. Common ones such as True Positive Rate (TPR), also known as recall, and False Positive Rate (FPR) can also be used to plot another visualization of the model's performance in the form of the receiver operating characteristic (ROC) curve. The ROC curve is simply the plot of TPR against FPR at each threshold of a classification model (POWERS, 2020). The formula to obtain the TPR and FPR can be seen in Equations 2.6 and 2.7.

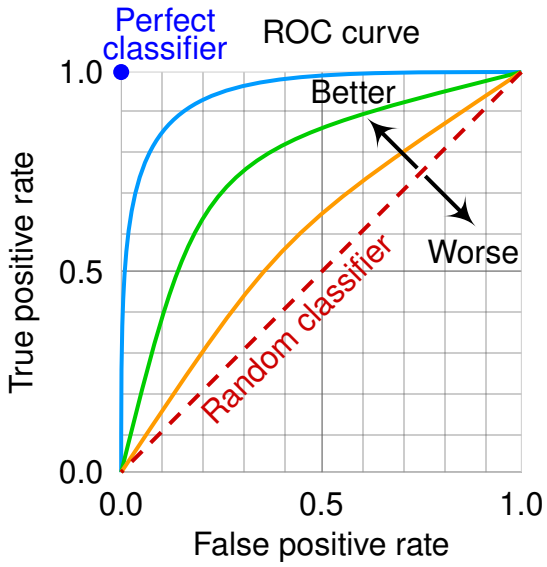
$$\mathrm{TPR} = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FN})} \tag{2.6}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{(\mathrm{FP} + \mathrm{TN})} \tag{2.7}$$

The ROC curve provides a tool for selecting the operating point (threshold) that optimizes the model regarding TPR and FPR. A perfect classifier can be seen in Figure 13 at the top left corner when the TPR is 1 and the FPR is 0. It is not always possible to maximize TPR and minimize FPR at the same time due to practical limitations in tasks that have restrictions on the amount of false negatives or false positives. Thus, one option is to fix either the FPR or TPR to meet the problem specification needs and perform experiments on different models or features to obtain the best result.

When comparing classifiers, we may want to reduce the performance metric to a single scalar that represents expected performance. A common method to do it with the ROC curve is to calculate the area under the curve, also called AUC. It has an important statistical property where the AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance (FAWCETT, 2006). Since the Area Under the ROC curve (AUROC) is a portion of the area of a unit square, the possible values for the AUROC range between 0 and 1, where 0.5 is equivalent to the area of random guessing, and 1 is the area of a perfect classifier.

Figure 3 – Receiver operating characteristic (ROC) curve.



Source: By cmglee, MartinThoma, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=109730045

2.3.4 Model evaluation and model selection

When training machine learning models, we are concerned with assessing the machine learning algorithm's performance on unseen data, as this reflects how well it will perform in real-world applications. The simplest way to do this is to evaluate its performance by splitting the dataset into a training set and a test set. The training set is used to fit a model, and the test set to predict the labels and evaluate its performance. It shall be noted that we do not want to train and evaluate the model on the same training dataset since this introduces an optimistic bias due to overfitting (RASCHKA, 2018).

However, in machine learning, a lot of experimentation and tuning of the different hyperparameters is required. This results in multiple different models where we are interested in selecting the single best-performing model. To do this, we need a method to estimate and compare the performance of these models. We cannot reuse the same holdout test set multiple times during the hyperparameter tuning process since it will introduce a bias in the final performance estimate, resulting in overly optimistic estimates of the generalization performance by leaking information. To avoid this problem, it is common to split the dataset into a train, validation, and test set. Now, the process revolves around using the training and validation set for hyperparameter tuning and model selection and leaving the test set for model evaluation (RASCHKA, 2018).

This method of splitting the dataset into train, validation, and test sets for hyperparameter tuning and model selection is not the only method and is often not the best approach for the task. There are multiple methods available in the literature (RASCHKA, 2018; TSAMARDINOS; RAKHSHANI; LAGANI, 2015), each with its advantages and trade-offs, and the selection of the method used depends on the task at hand, dataset size and other characteristics that

have to be carefully observed.

The K-Fold Cross Validation is one of the other methods available and perhaps the most common method for model evaluation and selection for small and medium sample sizes. The procedure involves dividing the dataset into K training and validation folds, using the entire dataset so that each sample has the opportunity to be tested. This minimizes the impact of data variability that may occur with the usual train-validate-test split when choosing a single split at random. The performance of each fold is estimated, and the final estimation is the result of averaging the performance of all K iterations. Figure 4 illustrates an example of a 5-fold cross validation procedure.

Figure 4 – Illustration of the K-Fold Cross Validation method.

Source: Author, 2024

It is important to note that although we aim to accurately estimate a model's future performance, biased performance estimates are acceptable for model selection if the bias affects all models equally. When we compare different models or algorithms to select the best-performing one, we only need to know their relative performance.

2.4 DEEP NEURAL NETWORKS

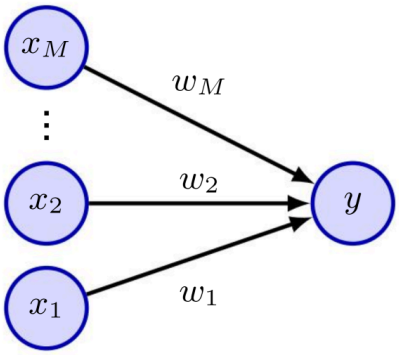
Neural network models were originally inspired by studies of information processing in the brains of humans and other mammals, where the basic processing units of the brain are called neurons. Just like the neurons in the human brain, the proposal of the first computational model of a neuron by McCulloch & Pitts (1943) is what brought forth the basis of computational approaches to learning, known as Artificial Neural Networks (ANNs).

The perceptron is a simple neural network introduced by Rosenblatt (1961) representing a linear combination of M inputs, which are then transformed using a nonlinear function called the activation function. A mathematical description of the perceptron can be seen in Equations 2.8 and 2.9 and also in diagram form in Figure 5. In the equations, w represents the weights, x represents the inputs, and the activation function $f(\cdot)$ is defined as the Heaviside step function.

$$a = \sum_{i=1}^{M} w_i x_i \tag{2.8}$$

$$y = f(a) \tag{2.9}$$

Figure 5 – A simple neural network representing a single neuron.



Source: (BISHOP; BISHOP, 2023)

The perceptron is a single-layer neural network since it only has one learnable layer. Its capability to learn from data in a brain-like manner was impressive, which brought attention to more studies in this area. However, Minsky & Papert (1969) showed limitations of the perceptron with formal proofs and presented that it could not solve the XOR problem (a classical problem to predict the output of XOR logic gates given two binary inputs). These limitations were one of the reasons for the first AI winter during the 1970s and early 1980s (BISHOP; BISHOP, 2023).

To solve the issues of training neural networks to learn more complex problems, more than one learnable layer of neurons was needed. To enable these changes to the training procedure, the solution came from the use of differential calculus and gradient-based optimization methods. This involved changing the step function to a continuous differentiable activation function and introducing differentiable error functions to evaluate the learnable parameters on how well the model predicts the target variables in the training set (BISHOP; BISHOP, 2023). The process of evaluating the derivatives of the error function that made a breakthrough in the field of neural networks is the backpropagation algorithm (RUMELHART; HINTON; WILLIAMS, 1986).

The development of the backpropagation algorithm and gradient-based optimization has allowed neural networks to solve many practical problems. The use of networks with multiple layers of learnable parameters has led to the sub-field of machine learning known as Deep Neural Networks (DNNs).

The effectiveness of training deep neural networks depends heavily on the proper tuning of the model's hyperparameters. Hyperparameters such as batch size, learning rate, and the number of epochs are crucial in determining how well the network converges to an optimal

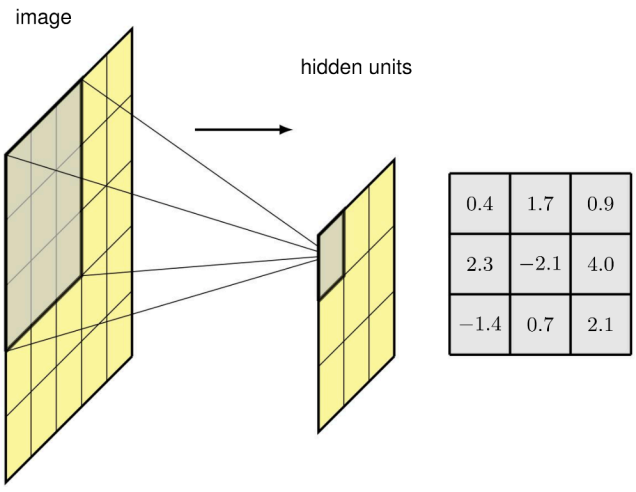
solution. Batch size affects how much data is processed during gradient computation, while the learning rate controls the size of steps taken during optimization. The number of epochs dictates how many times the entire dataset is used to update the model parameters.

2.4.1 Convolutional Neural Networks

Consider image classification as an application example of a neural network. A standard Multi-Layer Perceptron (MLP) requires a large number of parameters to process image data due to its high-dimensional nature. Imagine an image with a width W, a height H, and 3 channels corresponding to the red, green, and blue colors (considering RGB pixels). For each hidden unit of the first hidden layer of the neural network, there would be $W \times H \times 3$ weights, making it huge. On top of that, the neural network must be able to generalize what it has learned in one location to all possible locations in the image without needing to see examples in the training set at every possible location (BISHOP; BISHOP, 2023).

To address these problems, LeCun et al. (1998) proposed Convolutional Neural Networks (CNNs) in their paper, applying to the task of handwritten digit recognition in the form of the LeNet architecture. Unlike MLPs, CNNs use convolutional layers that preserve the spatial structure of the data. The convolutional layer is the primary building block of a CNN, which applies learnable filters (or kernels) to the input data to detect local patterns. This operation mimics the way human vision works by identifying simple features, such as edges, at early layers and progressively more complex patterns at deeper layers. These filters are shared across the input space, allowing the model to learn position-invariant features. Figure 6 shows an example of a unit in a hidden layer of a network that receives input from pixels in a 3x3 patch of the image. The weights associated with this hidden unit can be visualized as a small 3x3 matrix, also called a kernel.

Figure 6 – Illustration of a 3x3 kernel with weights associated with a 3x3 patch of an image as input.



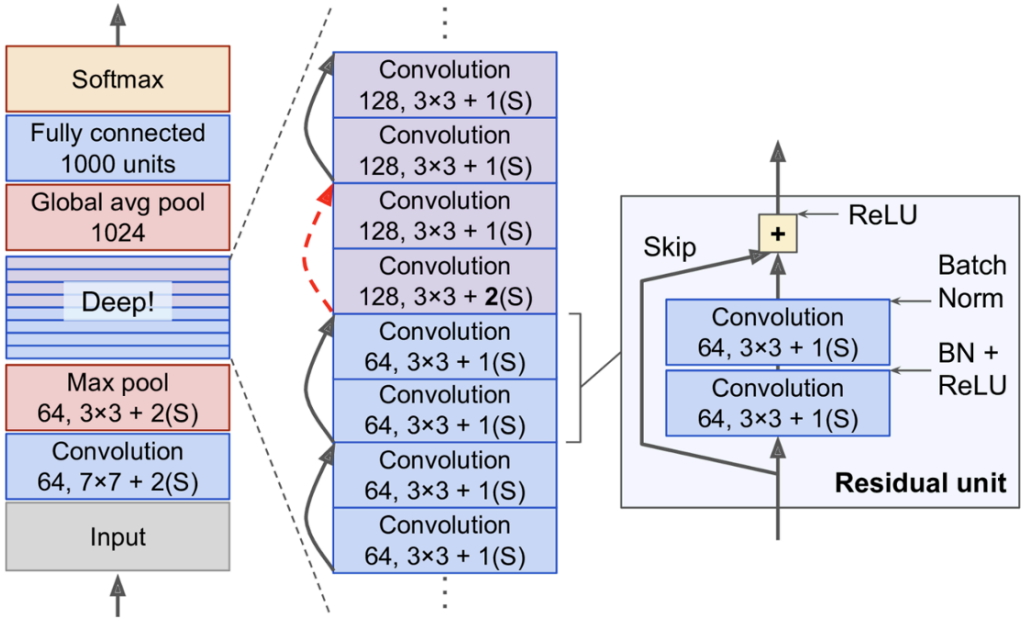
Source: (BISHOP; BISHOP, 2023)

2.4.1.1 ResNet

Residual Neural Networks (ResNet) were introduced by He et al. (2016) and represent a breakthrough in the development of deep learning architectures. Traditional deep neural networks suffer from an issue known as the vanishing gradient problem (BENGIO; SIMARD; FRASCONI, 1994) as they become deeper. It occurs when the gradients used to update the weights become increasingly small as they are backpropagated from the output layers to the earlier layers. The ResNet addresses this issue by introducing residual learning, allowing the creation of deeper convolutional networks with as many as 152 layers in the ResNet-152 that ultimately won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2015, obtaining state-of-the-art accuracy in image classification.

The main innovation of the ResNet lies in the residual block, which introduces skip connections that bypass one or more layers and feed the input directly into a layer deeper in the network, as can be seen in Figure 7 in the right side of the figure as the residual unit. Multiple versions of the ResNet architecture are available, with different amounts of layers (e.g., ResNet18, ResNet34, ResNet50, ResNet101), and since its introduction, it has become a foundational deep learning model used in various applications beyond image classification.

Figure 7 – ResNet base architecture.



Source: (GÉRON, 2022)

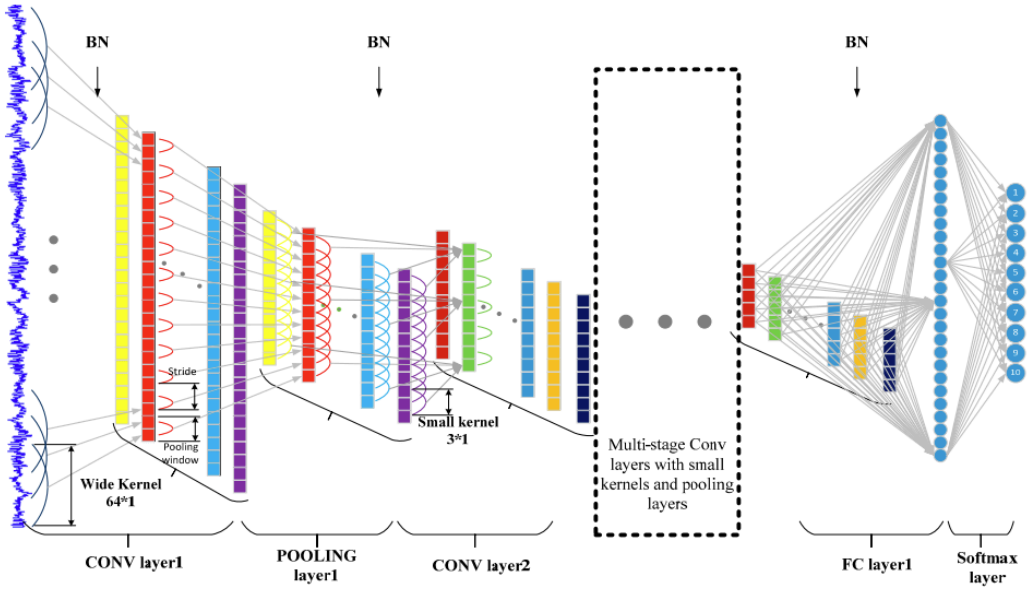
2.4.1.2 WDCNN

The Deep Convolution Neural Networks with Wide first-layer kernels (WDCNN) architecture introduced by Zhang et al. (2017) is a 1D convolutional neural network model de-

signed specifically for fault diagnosis in rotating machinery and other industrial applications that can use raw vibration data as its input. Traditional CNNs architectures have already been applied to fault diagnosis. However, the models before the WDCNN often failed to achieve a higher performance than traditional methods (ZHANG et al., 2017). This occurred for two main reasons: (1) the models are not deep enough, making it hard to obtain high nonlinear expressions from the signal; (2) the kernels of the models were not large enough to enable proper feature extraction from the signal.

To address these problems, the WDCNN's core innovation is the use of wider first-layer kernels to extract features and better suppress high-frequency noise. Subsequent small 3x1 kernels acquire better feature representation with a deeper model. Figure 8 illustrates the WDCNN architecture.

Figure 8 – WDCNN architecture.



Source: (ZHANG et al., 2017)

Zhang et al. (2017) demonstrated in their paper that WDCNN significantly outperforms traditional CNN models and other machine learning methods in fault diagnosis tasks on rotating machinery, being widely used in the literature.

3 METHODOLOGY

3.1 DATASET

The CWRU bearing fault dataset[1] is a collection of experiments that involved a single pair of healthy bearings and several artificially created faulty bearings. The faults were created through electro-discharge machining, introducing point faults with diameters of 7, 14, 21, and 28 mils in the inner race, outer race, and rolling element separately. For the outer race faults, the experiments considered faults located at three different positions relative to the load zone. The healthy and faulty bearings were reinstalled at both the drive end (DE) and fan end (FE) locations (where each configuration comprises either two healthy bearings or one healthy bearing and one faulty bearing), and data were collected synchronously, with one accelerometer at each location placed at the 12 o'clock position and attached to the housing with magnetic bases. It should be noted that the bearings at drive and fan end location have different specifications, where the drive end bearing is a 6205-2RS JEM SKF bearing and the fan end bearing is a 6203-2RS JEM SKF bearing.

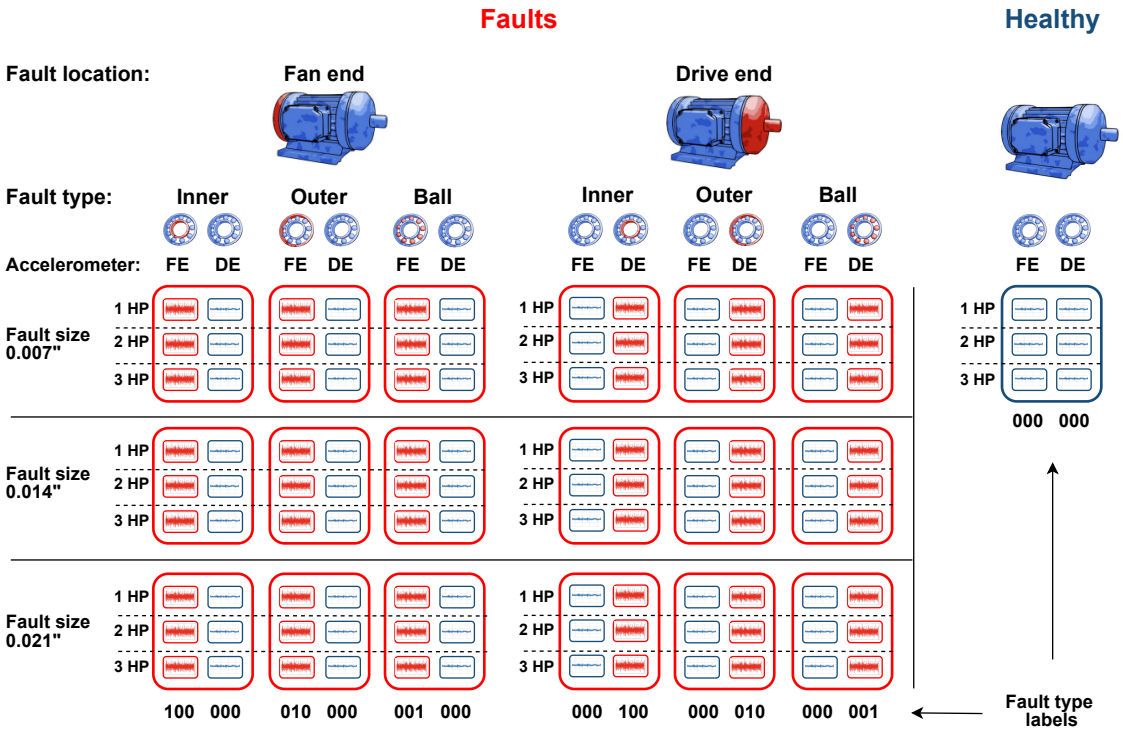
For each configuration, experiments were made using four operational motor load conditions ranging from 0 (no load) to 3 horsepower (HP). A 16 channel DAT recorder was used to collect the vibration signals that were later processed in a Matlab environment. Most of the experiments were collected at 12 kHz, but some experiments were also collected at 48 kHz for drive end bearing faults and healthy samples. The information for the speed and horsepower data were collected using the torque transducer/encoder and were recorded by hand. All the experiments consist of signals that are approximately 10 seconds long.

Following Hendriks, Dumond & Knox (2022), the configurations considered in this work used a load varying from 1 to 3 HP and fault sizes of 7, 14 and 21 mils. Measurements with a sampling rate of 12 kHz were used, except for the healthy bearing experiments that only had a sampling rate of 48 kHz available and were resampled to 12 kHz. Considering the three different fault positions at the outer race fault experiments, the "Centered @6:00" experiments were primarily used whenever possible. If the former did not exist, the "Orthogonal @3:00" experiments were used. All these configurations can be seen in Figure 9, where each box represents a different bearing configuration. This work also refers to these configurations as fault conditions (specified by fault location, type and size, plus the healthy state), as the dataset contains a single bearing configuration (a single pair of specific bearings) for each fault condition.

Observe that the boxes representing a different bearing configuration all contain signals from a single faulty bearing at a single location, while the location opposite to the fault contains a healthy bearing. Since only one healthy bearing is available in the CWRU dataset for that specific location, this bearing is the same as the one in the healthy bearing configuration

[1] Available at: https://engineering.case.edu/bearingdatacenter

Figure 9 – Signals from the CWRU bearing fault dataset considered in this work. Each box corresponds to a different bearing configuration (fault condition) specified by fault location, type and size, plus the healthy state. Six signals are acquired at two accelerometers (FE: fan end; DE: drive end) for each configuration at three different loads. The labels (which are the same for all signals in the same column) indicate the occurrence of an inner and/or outer and/or ball fault, in this order, at the same location where the signal is acquired.



Source: Author, 2024

(the configuration without a faulty bearing). However, upon closer inspection in an exploratory data analysis, we can see in Figure 10 that signals from a healthy bearing under a faulty configuration are quite different from those under a healthy configuration and have characteristics that resemble the fault occurring at the opposite location.

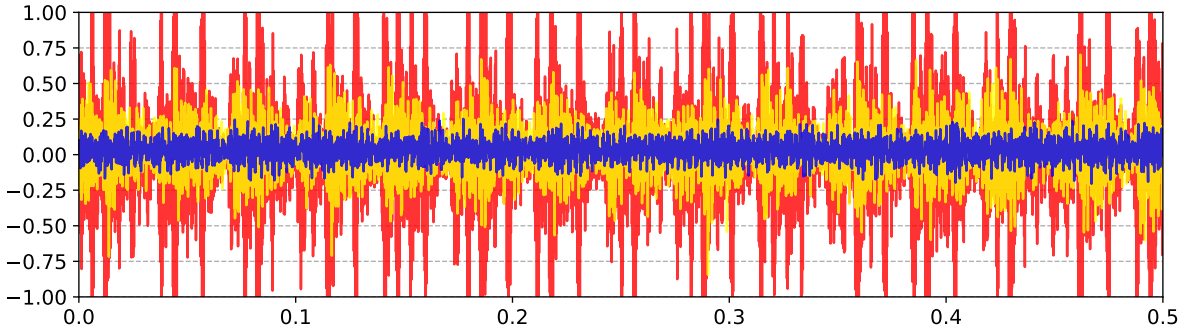
3.2 PROBLEM FORMULATION

The problem of detection and diagnosis of bearing faults can be formulated in many different ways, which can profoundly impact what a model can predict and how it is evaluated.

First, consider the problem of detecting faults located at the drive end using only signals acquired at the drive end. In this case, a simple binary healthy/faulty bearing classification is sufficient. However, suppose there is also a need to diagnose the fault type. If we assume that only a single fault type can happen simultaneously, i.e., the fault types are mutually exclusive, then multi-class classification is appropriate; the corresponding classes would be *healthy*, *inner*, *outer* and *ball*. On the other hand, if we allow the possibility for multiple fault types to appear simultaneously, then (binary) multi-label classification should be used. In this case,

Figure 10 – Raw vibration signals of a inner race fault at the drive end location with a fault size of 21 mils and load of 3 HP. Signals from both locations are compared to the healthy bearing configuration signal.

(a) Comparison between healthy bearing configuration signal with signals coming from faulty bearing configurations.



(b) Figure zoom to highlight the differences between healthy bearing signals from the healthy bearing configuration with the healthy bearing signal at the faulty bearing configuration.

Source: Author, 2024

each sample is labeled with three binary digits, each indicating whether an inner/outer/ball fault is present or not; in particular, the healthy state corresponds to the case where no fault type is present.

In a broader context, our proposed multi-label approach to fault detection and diagnosis is that of constructing specific detectors for each possible type of fault. Then, conventional fault detection amounts to simply verifying if *any* of the specific detectors gives a positive output, while fault diagnosis amounts to retrieving *which* specific detectors give a positive output.

Besides being physically more realistic, this multi-label formulation presents several advantages in the case of the CWRU dataset, where fault types are mutually exclusive. First, the negative class samples for a given fault type, say inner, comprise samples with the other fault types, say outer and ball, besides the healthy samples. This mitigates the problem of the healthy class's heavy class imbalance (in lack of samples). More importantly, it allows us to move all the healthy bearing configuration samples to the test set, avoiding data leakage. Additionally, fine-grained, prevalence-independent evaluation metrics for binary classification, such as the ROC curve, can be used, resulting in a more precise evaluation.

Similar reasoning can be applied to the problem of identifying, additionally, the loca-

tion of a fault (drive end or fan end) using both signals acquired at the drive end and the fan end. Hendriks, Dumond & Knox (2022) considered this problem under the assumption that the fault locations were mutually exclusive, formulating a multi-class classification problem with seven classes: *healthy*, *drive-inner*, *drive-outer*, *drive-ball*, *fan-inner*, *fan-outer* and *fan-ball*. Instead, we take a multi-label approach and treat the fault diagnosis at each location as independent problems. Specifically, each sample (consisting of a pair of DE/FE signals) is now labeled with two triples of binary digits, each triple containing the multi-label fault type classification for each location. Again, besides being more realistic, as it is physically possible for faults to arise at the two locations simultaneously, this formulation presents several advantages in the case of the CWRU dataset.

First, as before, the samples with faults at the FE serve as negative samples for fault diagnosis at the DE and vice-versa. Second, since one signal is acquired at each location where a fault is to be diagnosed, we can split the problem in two. Namely, one model can be responsible for detecting faults at the DE based on signals acquired at the DE, while another model is analogous with respect to the FE. In this case, each DE or FE signal becomes a distinct sample and the labels are also split between the two signals: the DE signal is labeled with the triple of binary digits referring to faults at the DE, while the FE signal is labeled with the triple of binary digits referring to faults at the FE. In other words, each sample is labeled positively for a given fault type if and only if that fault type occurred at the specific location where the corresponding signal was acquired. Naturally, each DE or FE model is trained with only their respective DE or FE signals. This approach has the additional advantage of relaxing the assumption that both signals are acquired synchronously, which is difficult to realize in practice. Indeed, this new formulation corresponds to the typical practical scenario where we wish to detect faults at a specific spot based on signals measured at that exact spot.

Finally, we can exploit the symmetry of the problem (assuming that DE and FE bearings are sufficiently similar) and consider a single model to be used at both ends, which is trained on all samples (DE and FE), effectively doubling the size of the training dataset. After a model is trained on such samples, the inference is made by running the model twice, once at each location, with the corresponding signal as input. This approach has the advantage of increasing the diversity of signals that a single model experiences, potentially improving its generalization to other locations. The latter is our final approach to multi-label classification, summarized in Figure 9.

3.3 DATASET DIVISION

In the division proposed by Hendriks, Dumond & Knox (2022), faulty bearings with the same fault size are grouped, resulting in three different subsets (7, 14, 21 mils) that later are used to train and evaluate models. For example, when the subset with a fault size of 7 mils is used for training, the remaining subsets of 14 and 21 mils are used as test sets. This division prevents the occurrence of the same faulty bearing configurations in both the train

and test datasets, effectively addressing the data leakage issue. However, this solution is not entirely foolproof, as the dataset contains a single healthy bearing configuration, which must be split into the three previously described data subsets. In the Hendriks, Dumond & Knox (2022) approach, this division is based on load, with healthy bearings at loads of 1, 2, and 3 HP being assigned to the 7, 14, and 21 mils subsets, respectively. It should be emphasized that any division of the healthy bearing configuration necessarily results in data leakage.

In contrast to the dataset division described above, in practice, faults may occur in various sizes, making it unrealistic to have a dataset distribution of only one fault size during training. This can lead to difficulty in training a model to detect faults at different sizes that did not appear in the training set. Therefore, it is important to consider a more diverse range of fault conditions in the dataset to ensure that the model can accurately predict them.

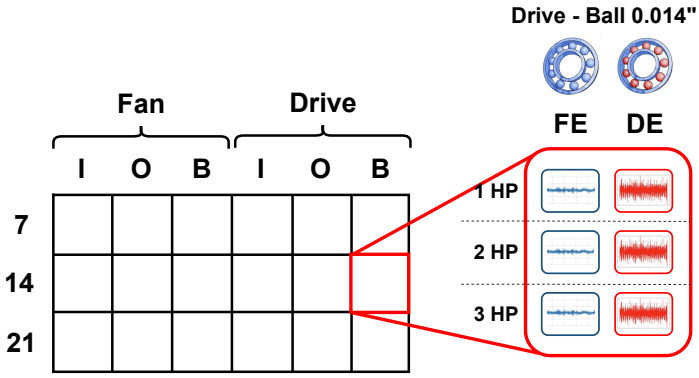
A more realistic approach would be a division where multiple loads, fault sizes, types, and locations randomly appear in each subset. Hence, the proposed division selects signals from a random fault size configuration for each fault location and type pair, provided that signals from the same bearing configuration appear only in a single (train or test) set to avoid data leakage. The selected signals—together with all the healthy bearing signals—are used for the test set, and the remaining signals are used in the training set[2]. This results in 3^6 possible train-test splits.

In addition to this hold-out method, it is also possible to conduct 3-fold cross-validation using the proposed division. The method begins by randomly selecting fault size configurations for each fault location and type pair, mirroring the process of the hold-out method. However, instead of designating this sample set as a test set, it becomes our initial fold. To generate the remaining two folds, a subsequent hold-out split is executed on the remaining configurations after excluding the first fold. This leads to a total of $3^6 \times 2^6/6$ potential 3-fold cross-validation splits. Note that only the faulty bearing signals are considered in this process, with the healthy bearing signals always being placed in each corresponding test set.

Figure 11 provides a simplified representation of the fault samples in the dataset shown in Figure 9. Each cell in Figure 11 corresponds to a fault condition (specified by a fault location/type column and a fault size row) and represents all the signals acquired in the corresponding configuration (namely, the signals acquired by both fan end and drive end accelerometers for every single load). Figure 12 shows an example of a possible train-test split for the fault signals, as well as an example of a 3-fold partition. It is important to note that, while not represented in Figure 12, the 6 signals corresponding to healthy bearings are always present in the test set.

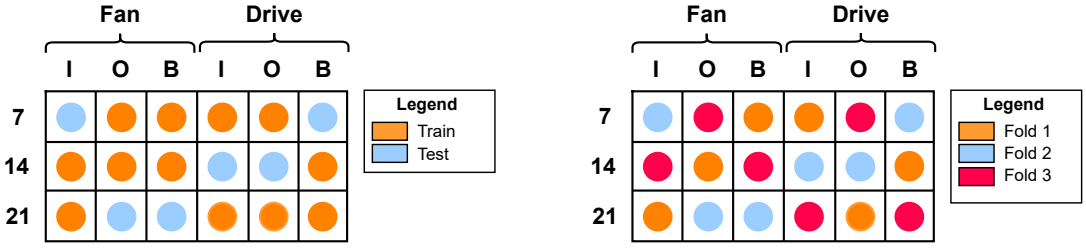
[2] It is important to remember that a bearing configuration contains signals from both of the faulty and healthy bearings, the latter having the same bearing as the healthy bearing configuration (since only one healthy bearing is available at the CWRU dataset). However, as previously seen in 3.1 the contamination caused by the faulty bearing in the opposite location makes very different to the healthy bearing configuration. While there is still some leakage considering unique bearings for each train-test split, we do not have bearing configuration leakage which seems to have worse effects during training and evaluation.

Figure 11 – Simplified dataset representation. Each column represents a fault type (I: Inner race; O: Outer race; B: Ball) and location, and each row represents a fault size. Each cell represents all the signals acquired in the corresponding fault condition (which, in the CWRU dataset, comprises a single-bearing configuration). This representation emphasizes that all signals acquired in the same bearing configuration, being potentially correlated, should belong to the same (train or test) data subset.



Source: Author, 2024

Figure 12 – Example of proposed division. Each column must contain exactly a single test group (blue dot). Note that the healthy bearing configuration signals are always placed in the test set.



(a) Example of a train-test split.

(b) Example of a 3-fold partition.

Source: Author, 2024

3.4 MODEL SELECTION AND EVALUATION METHODOLOGY

For hyperparameter optimization (also called model selection) and performance evaluation, we adopt the Double Cross-Validation Method (CVM-CV), described in (TSAMARDINOS; RAKHSHANI; LAGANI, 2015). This protocol consists of applying the cross-validation method for hyperparameter optimization (CVM) and then reevaluating it on different train-test splits (CV) for final performance estimation for the single, selected, best model. Note that using only CVM is well-known to overestimate performance since it returns the maximum performance achieved across several hyperparameter configurations. This bias can be reduced by reevaluating only the selected hyperparameter configuration on different train-test splits.

In this work, the CVM-CV method is used on the CWRU bearing fault dataset by applying a CVM step on a random 3-fold partition as described in Section 3.3 for hyperparameter optimization. For each set of hyperparameters, the model is trained and tested three times: in each iteration, two folds are used for training, and the remaining fold is used for validation. An example of a 3-fold partition can be seen in Figure 12b. The average validation metric across

the three folds is calculated to evaluate the performance of the hyperparameters. By repeating this for multiple hyperparameter configurations, the combination that yields the best average metric is selected. With the hyperparameters optimized, the CVM step is followed by a CV step using 30 random train-test splits (with static seeds) to evaluate performance on the selected hyperparameters. Each train-test split is made just as the example seen in Figure 12a and the seeds are reused across experiments to ensure that the different models are tested on exactly the same test sets.

3.5 MODEL ARCHITECTURES AND TRAINING DETAILS

Numerous techniques have been proposed for bearing fault diagnosis, including signal processing (RANDALL, 2021; SMITH; RANDALL, 2015), machine learning (LEI et al., 2020) and deep learning (JIA et al., 2016). Among these techniques, Convolutional Neural Networks (CNNs) have gained wide attention and exploration due to their exceptional performance and efficiency in image recognition systems. These models can be used for various tasks, including fault diagnosis, and have shown remarkable results (ZHANG; LI; DING, 2019; ZHANG et al., 2017; ZHANG et al., 2018). They typically use 1D or 2D vibration signal representations on time, frequency, or time-frequency domains and can be trained either from scratch or fine-tuned from pre-trained image models.

In this work, the ResNet architecture (HE et al., 2016) is used as the primary model for the experiments due to its promising results in the literature. The batch size, learning rate, and the number of epochs were optimized, aiming to maximize macro average AUROC by using a grid search strategy along the CVM-CV method described in Section 3.4. The hyperparameter grid included batches of sizes 32, 64, and 128, with learning rates between 10^{-7} and 10^{-3} (with a multiplicative step of 10). Experiments were done with 10 training epochs but with a checkpointing to save only the epoch with the highest metric. After hyperparameter optimization, the ResNet18 architecture was fine-tuned for four epochs, using a batch size of 128, a constant learning rate of 10^{-5} and the Adam optimizer (KINGMA; BA, 2014) starting from weights pre-trained on the ImageNet dataset.

Given the results with a pre-trained ImageNet vision model, new experiments were proposed with other similar model architectures from PyTorch's torchvision models (MAINTAINERS; CONTRIBUTORS, 2016) such as MobileNet (HOWARD et al., 2019), RegNet (RADOSAVOVIC et al., 2020), ConvNeXt (LIU et al., 2022b), Swin Transformer (LIU et al., 2022a), ViT (DOSOVITSKIY et al., 2020), EfficientNet (TAN; LE, 2021) and MaxVit (TU et al., 2022). As simpler baseline models, the 7-layer CNN and 9-layer CNN (ZHANG et al., 2021) (typically used in experiments on the CIFAR 10/100 dataset) were also included in our experiments. We also experimented with different-sized ResNet architectures (HE et al., 2016), such as the ResNet34, ResNet50, and ResNet101. All the models used pre-trained model weights and binary cross-entropy loss. For the experiments with 1D signal representations on the time or frequency domain, we utilized the WDCNN model architecture (ZHANG et al.,

2017)[3] The WDCNN is a 1D CNN that was built with the purpose of being used on machine fault diagnosis problems and has shown results close to those obtained by the ResNet model (HENDRIKS; DUMOND; KNOX, 2022). All of the model's hyperparameters, such as batch size, learning rate, and number of epochs, were optimized using exactly the same procedure as for the ResNet18 architecture, the optimized hyperparameter can be seen in Table 3.

Table 3 – Tuned hyperparameters for all models.

Model	**# Param.**	**Hyperparameters**		
		Batch Size	**Learning Rate**	**Epoch**
ResNet18 (HE et al., 2016)	**11.7M**	**128**	**10^{-5}**	**4**
ResNet34 (HE et al., 2016)	21.8M	128	10^{-5}	3
ResNet50 (HE et al., 2016)	25.6M	128	10^{-6}	7
ResNet101 (HE et al., 2016)	44.5M	128	10^{-6}	9
CNN7 (ZHANG et al., 2021)	40M	32	10^{-5}	1
CNN9 (ZHANG et al., 2021)	2.8M	64	10^{-3}	5
MobileNet V3 (large) (HOWARD et al., 2019)	5.5M	32	10^{-4}	7
RegNet (x_1_6gf) (RADOSAVOVIC et al., 2020)	9.2M	64	10^{-4}	3
ConvNeXt (tiny) (LIU et al., 2022b)	28.6M	64	10^{-4}	1
SwinTransformerV2 (tiny) (LIU et al., 2022a)	28.4M	64	10^{-4}	3
ViT (b_16) (DOSOVITSKIY et al., 2020)	86.6M	64	10^{-6}	1
EfficientNetV2 (small) (TAN; LE, 2021)	21.5M	64	10^{-4}	1
MaxViT (TU et al., 2022)	30.9M	32	10^{-5}	1
WDCNN (ZHANG et al., 2017) - Time	53.5K	32	10^{-4}	6
WDCNN (ZHANG et al., 2017) - Spectrum	53.5K	32	10^{-3}	8
WDCNN (ZHANG et al., 2017) - Power Cepstrum	40.7K	32	10^{-3}	6

Source: Author, 2024

The pre-processing steps followed those in (HENDRIKS; DUMOND; KNOX, 2022). For the 2D models, it involved a segmentation with an overlap of 97% (resulting in segments with a window length of 11500) and calculating the spectrogram of each segment using a window size of 104 points, 54 points of overlap, and 452 DFT points. These parameters give each resulting spectrogram a size of 227x228. However, to match the ResNet model's 224x224 network input size, the spectrogram is cropped by keeping the lower-left portion (removing some of the high frequencies and some of the last few time steps).

For the 1D models, three different signal representations were used (time, spectrum and power cepstrum) and the pre-processing steps differed slightly for each of them. For the time-domain signal, raw temporal accelerometer data is segmented with the same overlap of 97% but with a window length of 2048 samples. For the spectrum and power cepstrum signal

[3] Note that only the baseline WDCNN architecture proposed by the model's authors was used, and no experiments with the additional Adaptive Batch Normalization were done.

representations, a window length of 4096 samples is used. The spectrum is computed by taking the Fast Fourier Transform (FFT) of the raw data, while the power cepstrum is computed by taking the power spectrum of the logarithm of the signal's power spectrum.

The 1D and 2D features were then scaled using z-score normalization[4] fitted on the whole training set (i.e., resulting in a single scalar for the mean across all features and all inputs and similarly for the standard deviation).

[4] Also known as standardization, a data preprocessing technique to transform data into a standard scale with mean 0 and a standard deviation of 1.

4 EXPERIMENTS AND RESULTS

The experiments were conducted using Python with the PyTorch framework on a machine equipped with an RTX 3090 GPU. For each model, after hyperparameter optimization, each train-test run was repeated 30 times with different random dataset divisions as explained in Section 3.4, and the final results presented are the averages and standard deviations of those 30 runs.

The results and error analysis of the proposed methodology are given in Section 4.1, with ablation experiments for the proposed method being presented in Section 4.2. We discuss on Section 4.3 whether having two different detectors with different decision thresholds for each fault location/type is better than having a single combined fault type threshold. Further experiments on different convolutional vision models pre-trained on ImageNet are given in Section 4.4. Finally, Section 4.5 evaluates modifying the problem as fault detection.

4.1 MULTI-LABEL FAULT DIAGNOSIS

Table 4 shows the AUROC of the ResNet18 model for each fault location/type, while Figure 13 shows the corresponding ROC curves. Bear in mind that while we have used a single model to diagnose faults at both locations, in principle, each location defines a separate problem and, therefore, a potentially different detector (in particular, we may use different decision thresholds for each fault location/type detector). Thus, the results are presented separately for each fault location/type and subsequently averaged.

As can be seen, all detectors show good performance, with a macro-average AUROC of 0.911 ± 0.038. The highest and lowest performance are achieved, respectively, by the ball fault detector at the fan end, with an AUROC of 0.997 ± 0.011, and the inner fault detector at the fan end, with an AUROC of 0.857 ± 0.114.

This difference in performance, and in particular in the standard deviation for each ROC curve, can be explained by the error analysis in the next subsection.

Table 4 – $100 \times$ AUROC (mean $\pm$ std) for the ResNet18 model.

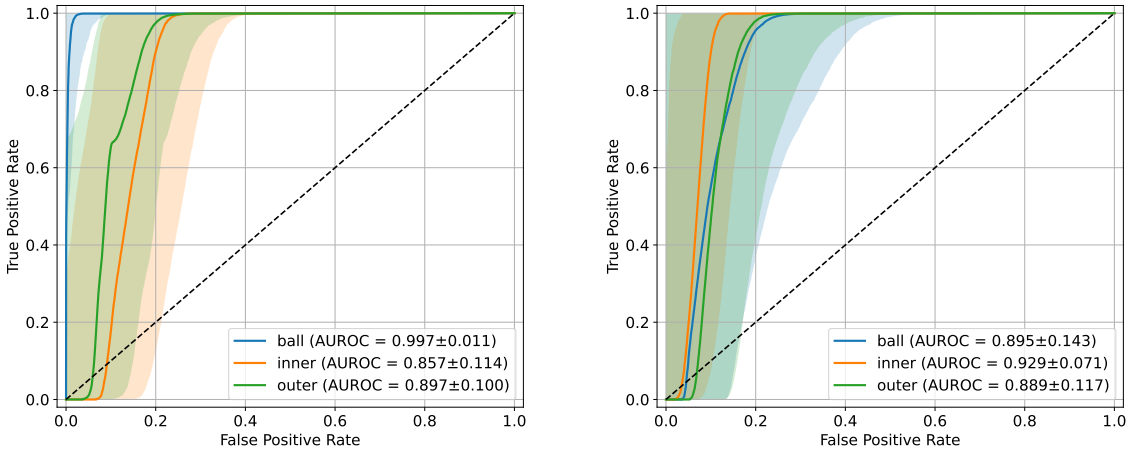
Model	**Fan end**			**Drive end**			**Macro Average**
	Ball	**Inner**	**Outer**	**Ball**	**Inner**	**Outer**	
ResNet18	99.7 $\pm$1.1	85.7 $\pm$11.4	89.7 $\pm$10.0	89.5 $\pm$14.3	92.9 $\pm$7.1	88.9 $\pm$11.7	91.1 $\pm$3.8

Source: Author, 2024

4.1.1 Error analysis

Figure 14 shows boxplot visualizations of the output *logits* (raw confidence scores for the positive class, before squashing by the logistic function), for each fault location/type

Figure 13 – ROC curves for multi-label fault diagnosis at each location. The solid curves represent the horizontal average ROC curve across 30 realizations, In all cases, the filled region represents the standard deviation.



(a) Fan end

(b) Drive end

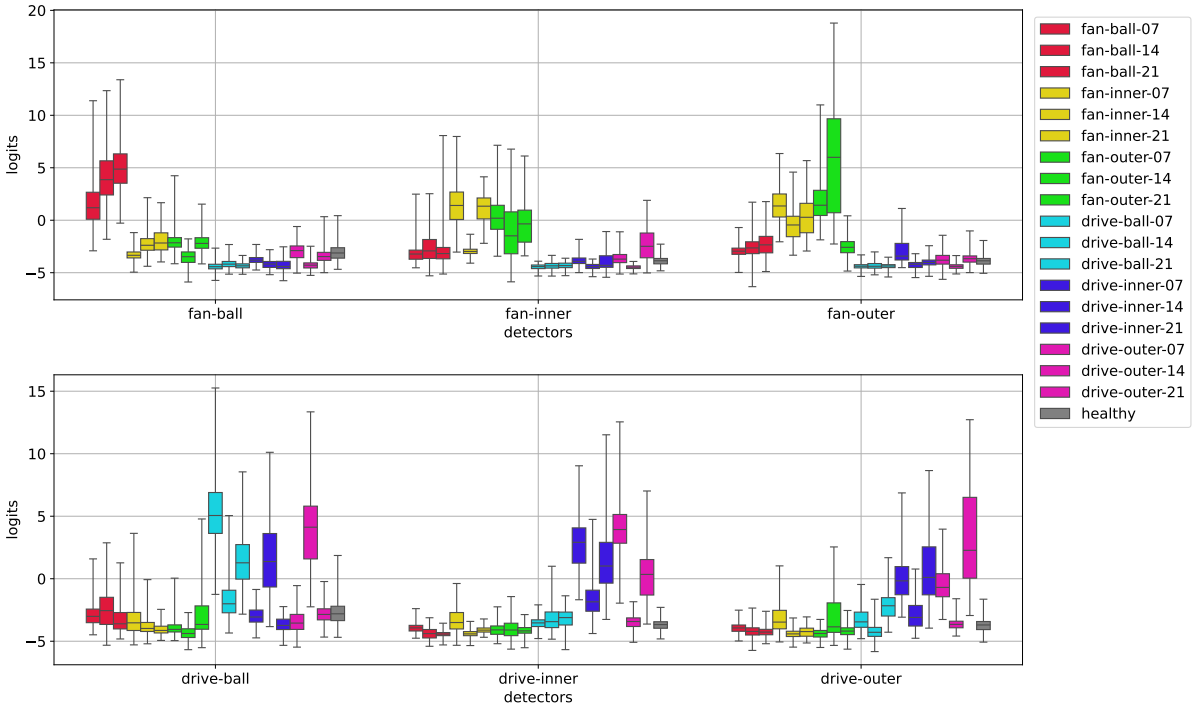
Source: Author, 2024

detector, obtained by grouping, for each of the 19 fault conditions, all realizations where that condition appears in the test set. The higher the logit output, the higher the confidence of the model that there is a fault of that specific type at that specific location. Usually, multi-label models have an added step of passing this score through a sigmoid activation function to transform it into a probability between 0 and 1, but this step was removed for this visualization to allow better interpretation without the probability capping at 1.

As can be seen in Figure 14, most detectors prioritize the correct fault conditions, by assigning higher scores than those of the corresponding negative conditions. (Note that only relative values matter, as different thresholds can be set for each detector in order to reach a desired trade-off between true and false positives.) However, with the exception of the fan-ball detector, which reached almost perfect discrimination, there was often some confusion between fault types at the same location, mostly between inner and outer faults. In particular, the fault conditions fan-inner-14, fan-outer-21, drive-inner-14, and drive-outer-14 achieved scores lower than expected for their respective detectors, while the latter two conditions additionally achieved (incorrectly) high scores on the drive-ball detector.

These errors can be partly explained with the help of the comprehensive analysis of the CWRU dataset performed by Smith and Randall (SMITH; RANDALL, 2015). They applied three established diagnostic techniques and found that relatively few of the records gave classical characteristics of the fault types, with many of the records showing evidence of mechanical looseness. As a consequence, some records were cited as difficult to diagnose or not diagnosable with any of the applied methods; these include, among the signals considered in our work, the drive-inner-14 and drive-outer-14 conditions and most of the ball faults. It should be noted that, according to our dataset division, when a particular fault condition (say, drive-inner-14) is included in the test set, the other two fault sizes of the same condition (drive-inner-07

Figure 14 – Boxplots of the raw confidence scores (logits) for fault diagnosis. Each group of boxplots corresponds to the detection of a specific fault type at a specific location. Each boxplot within a group corresponds to one of 19 fault conditions (denoted by fault location-type-size plus the healthy state) and displays the spread of logits among all train-test realizations where that condition appeared in the test set. At each boxplot, the whiskers represent the minimum and maximum values achieved.



Source: Author, 2024

and drive-inner-21) necessarily appear in the training set. If only the former is difficult to diagnose, it likely will not share similar characteristics with the other ones in the training set, and therefore, its scores will be low for the corresponding detector. Conversely, when difficult-to-diagnose (and possibly anomalous) signals appear in the training set, the model will be forced to learn unconventional (and possibly spurious) features of these signals in order to push their scores up, which may inadvertently give false positives to other fault types. This helps explain the frequent confusion between inner and outer faults, as well as why precisely the drive-inner-14 and drive-outer-14 conditions achieve a high score on the drive-ball detector (they likely share spurious features with the ball fault signals at the same location). Nevertheless, it becomes clear from Figure 14 that the model rarely mistakes the fault location, a property which is further explored in Section 4.5.

Another interesting observation that can be made from Figure 14 is that there is apparently no correlation between fault score and fault size. One might assume that fault scores would be higher for larger fault sizes, but that is not the case. Again, this has been previously suggested by Smith and Randall in (SMITH; RANDALL, 2015), who observed that the fault size seemed to have a lower impact on the diagnosis outcomes than the test rig assembly (since disassembly and reassembly are required to replace a bearing with a different fault size). These results suggest that it may be impossible to predict fault sizes using this dataset.

4.2 ABLATION STUDIES

Several experiments were performed to evaluate the proposed methodology on different choices regarding model specificity, train-test split type and proportion, and segment length. The first ablation study investigated what was most advantageous, having a separate model for each bearing location or a single model trained on data from every bearing and evaluated once at each location (the latter is our proposed approach). In principle, using separate models makes sense since each bearing type (and its external environment) induces a potentially different data distribution, formally known as a *domain*, so each model could benefit from specializing to a single domain. On the other hand, we should expect the single model approach to perform better if the two domains are not too different since the model would then be exposed to twice the amount of training data.

In Table 5, the first row represents the single model approach following the methodology described in Section 3.2, while the second row considers a separate model for the DE and FE locations. We can see a drop in performance for the separate approach, indicating that learning from these two domains jointly is viable and beneficial.

Table 5 – 100 × AUROC (mean ± std) for ablation experiments using the ResNet18 architecture. Numbers in boldface indicate the best result for each column.

Model	Split type	Split ratio	Signal length	Fan			Drive			Macro Average
				Ball	Inner	Outer	Ball	Inner	Outer	
Single	Random	2:1	Full	**99.7 ±1.1**	**85.7 ±11.4**	**89.7 ±10.0**	89.5 ±14.3	**92.9 ±7.1**	**88.9 ±11.7**	**91.1 ±3.8**
Separate DE/FE	Random	2:1	Full	99.1 ±3.01	82.7 ±11.7	89.5 ±10.5	90.6 ±13.4	92.0 ±7.3	76.5 ±13.1	88.4 ±4.3
Single	By fault size	2:1	Full	94.5 ±2.4	80.5 ±4.1	88.9 ±1.3	81.5 ±2.2	84.3 ±5.5	76.6 ±3.2	84.4 ±1.4
Single	Random	1:2	Full	92.5 ±6.9	77.9 ±7.5	80.3 ±15.4	70.0 ±14.1	84.6 ±11.3	73.1 ±17.9	79.7 ±7.0
Single	Random	2:1	Half	98.9 ±3.2	83.4 ±13.8	89.0 ±11.6	**90.7 ±13.1**	92.6 ±7.7	88.0 ±14.1	90.4 ±3.8
Separate DE/FE	By fault size	1:2	Full	89.0 ±1.0	76.7 ±2.8	73.6 ±3.5	82.8 ±2.7	90.6 ±2.9	56.0 ±4.5	78.1 ±1.4

Source: Author, 2024

Another ablation considered splitting the data into train and test sets based on fault size. In this approach, the bearing configurations for one of the fault sizes (7, 14, 21 mils) were allocated to the test set, along with the healthy state, while configurations with other fault sizes were used for training. This is similar to what is done in (HENDRIKS; DUMOND; KNOX, 2022), but with a train/test proportion of 2:1 instead of their 1:2 proportion. This results in three different train/test splits that are repeated 10 times for each split, resulting in 30 runs, just like the other experiments. The macro average AUROC dropped from 0.911 to 0.844, showing that the increase in data diversity (without leakage) brought by the proposed random split of Section 3.3 is indeed beneficial.

Many papers consider a 1:2 train/test split proportion when generating subsets based on load or fault size. From a purely machine learning perspective, this can be seen as an uncon-

ventional choice, as it is common practice to reserve a larger fraction of the data for training, arguably because models with sufficient capacity can benefit from more training data. To test whether this is the case for our problem, an experiment was performed using the 1:2 proportion, and as can be seen in Table 5, a considerable drop in performance was observed. This result, in turn, raises another question: is this loss in performance due to merely a smaller training set or due to a less *diverse* training set? To answer this question, an ablation experiment was performed using only the first half of each signal before segmentation, thus resulting in the same training set size as the 1:2 split but with the same diversity of fault conditions as the 2:1 split. The results show a macro average AUROC of 0.904, which is quite close to that of the original model, indicating that the leading cause of the drop in performance is the reduced diversity in the training data—or, equivalently, that increased training data diversity is highly beneficial.

A final experiment was made considering a baseline training approach with separate models for each location and a train/test split by fault size with a 1:2 split ratio (an attempt to approximate the setup of (HENDRIKS; DUMOND; KNOX, 2022) under our formulation), which resulted in a macro average AUROC of 0.781. It follows that the overall effect of our training approach is to significantly improve the performance by about 13 AUROC percentage points over the baseline.

4.3 SINGLE THRESHOLD FOR FAULT TYPES

Our proposed approach considered a single model trained on data from every bearing, evaluating each location individually. In Section 4.2, one of the proposed ablation studies has shown that performance-wise, it is better to have a single model trained on data from both drive and fan end bearings than separate models for each bearing location. While the benefits of this training approach were evident, evaluating each location individually (as having one detector for each fault location/type) can be challenging for real-world operations because of the need to optimize each detector's threshold when multiple different bearings appear.

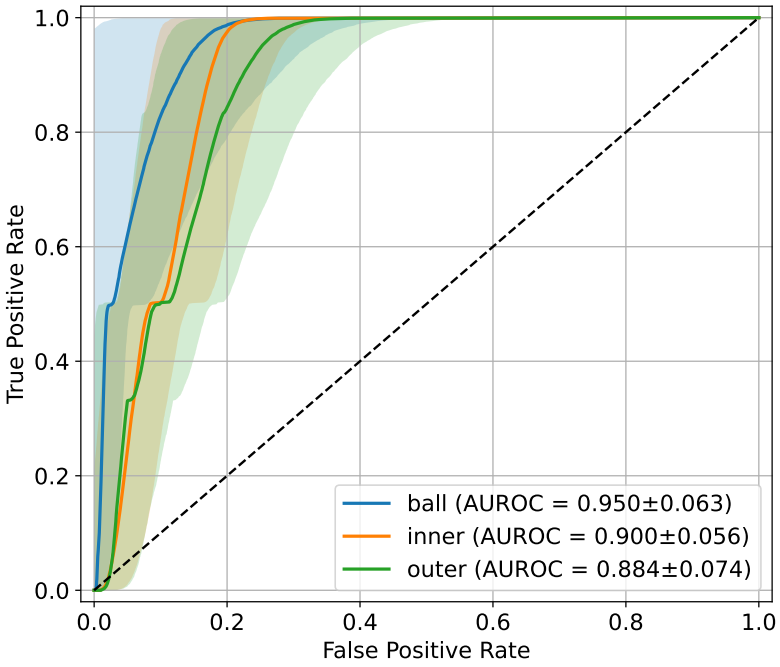
For this reason, another form of evaluation is proposed to simplify the problem by having, for each fault type, a single combined threshold independent of its location. With only a single threshold for each fault type, the threshold optimization process is limited to the number of fault types, resulting in less computational power needed to operationalize the model. Table 6 shows that having a single combined threshold for each fault type does not degrade the model's performance when considering the macro AUROC metric.

Table 6 – 100 × AUROC (mean ± std) for the ResNet18 model using a combined threshold for each fault type.

Model	**Ball**	**Inner**	**Outer**	**Macro Average**
ResNet18	95.0 ±6.3	90.0 ±5.6	88.4 ±7.4	91.1 ±3.6

Source: Author, 2024

Figure 15 – ROC curve for each fault type when combining both locations. Solid curves represent the average ROC curve across 30 realizations, while the filled region represents the standard deviation.



Source: Author, 2024

With the corresponding ROC curves shown in Figure 15, it is also possible to evaluate other business-oriented metrics for each fault type detector. Since false negatives usually have worse consequences in practice than false positives in fault detection and diagnosis, fixing the value of TPR (which is the complement of the false negative rate) is an appropriate way of setting an acceptable fault tolerance for false negatives, after which the FPR can be measured and compared. Assuming that 90% TPR is an acceptable rate for the system and considering the average ROC curve of Figure 15, the resulting fault-type average FPRs are 12.9%, 17.3%, and 22.4% for the ball, inner, and outer fault types, respectively.

4.4 OTHER MODEL ARCHITECTURES

In order to select the most suitable model architecture, various experiments were conducted using different convolutional vision models that were pre-trained on the ImageNet dataset, except for the CNN7 and CNN9 model architectures, which were trained from scratch and used as baselines. The experiments were divided into models that were simpler/smaller and more complex/larger than the ResNet18 to determine if any conclusions could be drawn regarding model complexity and performance. It is important to observe that the hyperparameters were optimized (and subsequently evaluated) for each experiment using the same model selection and evaluation methodology described in Section 3.4.

In Tables 7 and 8, it can be observed that neither the smaller nor larger models evaluated in these experiments yielded a better result than what was previously obtained with the

ResNet18, with the smaller models MobileNet and RegNet having a slight edge over the larger models present in Table 8.

Table 7 – 100 × AUROC (mean ± std) for simpler/smaller model architectures. Numbers in boldface indicate the best result for each column.

Model	Ball	Inner	Outer	Macro Average
CNN9	74.5 ±14.0	78.9 ±17.2	65.6 ±27.0	73.0 ±13.3
CNN7	78.8 ±13.3	76.1 ±13.1	**91.3 ±6.5**	82.1 ±5.2
MobileNet V3 (large)	**93.0 ±8.3**	88.7 ±10.2	81.9 ±14.4	**87.8 ±6.5**
RegNet (x_1_6gf)	92.2 ±8.2	**88.9 ±11.6**	79.1 ±13.7	86.7 ±7.0

Source: Author, 2024

Table 8 – 100 × AUROC (mean ± std) for more complex/larger model architectures. Numbers in boldface indicate the best result for each column.

Model	Ball	Inner	Outer	Macro Average
ConvNeXt (tiny)	**94.3 ±7.5**	81.4 ±12.2	75.8 ±18.1	83.9 ±6.9
SwinTransformerV2 (tiny)	91.7 ±10.3	83.8 ±13.4	75.6 ±14.7	83.7 ±7.9
ViT (base)	88.7 ±7.1	80.2 ±12.9	**83.3 ±17.2**	84.1 ±6.6
EfficientNetV2 (small)	91.5 ±8.9	84.8 ±10.9	82.9 ±15.9	**86.4 ±8.6**
MaxViT	90.0 ±9.8	**90.2 ±7.5**	78.6 ±21.5	86.3 ±8.3

Source: Author, 2024

Another experiment was performed with architectures from the ResNet family with higher numbers of layers, namely ResNet34, ResNet50, and ResNet101. In Table 9, it may be observed that although ResNet50 exhibited slightly better performance than that of ResNet18, this gain may be insufficient to justify its much longer training time due to a larger architecture. More importantly, the performance does not seem to steadily improve with increased complexity. Overall, these results suggest that, for the problem at hand, most gains are obtained not from increasing model complexity but rather from choosing a suitable architecture family, with the ResNet family appearing to be already the best choice among the candidates evaluated in this work.

Lastly, to address different representations other than the spectrogram, experiments with three different 1D signal representations were done with a smaller 1D CNN model architecture named WDCNN (ZHANG et al., 2017). The experiments for the time, spectrum and power cepstrum signal representations can be seen in Table 10. The time and spectrum representations are commonly seen in the literature, and their results were not as good as those of the spectrogram on the ResNet model, with the spectrum being marginally better than the time representation. The power cepstrum is not as commonly seen as the other two representations, as cepstrum analysis is mostly used for gearbox fault diagnosis, but it can also be used for detecting

Table 9 – 100 × AUROC (mean ± std) for other architectures within the RestNet family. Numbers in boldface indicate the best result for each column.

Model	Ball	Inner	Outer	Macro Average
ResNet34	92.2 ±8.0	88.4 ±6.6	87.7 ±8.2	89.5 ±5.0
ResNet50	92.9 ±5.2	**93.6 ±5.3**	**90.8 ±8.6**	**92.4 ±4.0**
ResNet101	**93.9 ±6.2**	90.4 ±6.4	90.4 ±9.3	91.6 ±4.6

Source: Author, 2024

the harmonics of bearing faults if they are well separated (RANDALL, 2021). Therefore, some other papers in the literature have attempted to use power cepstrum as input for deep learning models aiming to detect faults in the CWRU dataset (BHAKTA et al., 2019; CAI; TAN; CHEN, 2021). Table 10 shows that the power cepstrum signal representation achieved a macro average AUROC of 0.890, which, although inferior to that of the spectrogram on the ResNet model, is still quite competitive, as the WDCNN is much smaller, trains faster, and needs a smaller data segment than the ResNet models.

Table 10 – 100 × AUROC (mean ± std) for experiments using the WDCNN architecture on different signal representations. Numbers in boldface indicate the best result for each column.

Model	Signal Representation	Ball	Inner	Outer	Macro Average
WDCNN	Time	82.7 ±13.1	77.2 ±12.0	79.2 ±13.0	79.7 ±6.4
	Spectrum	86.5 ±12.7	79.7 ±12.4	**86.6 ±6.4**	84.2 ±6.4
	Power Cepstrum	**95.0 ±4.5**	**87.3 ±7.9**	84.7 ±9.4	**89.0 ±4.8**

Source: Author, 2024

4.5 FAULT DETECTION

As mentioned in Subsection 4.1.1, while some confusion between fault types occurs, it is mostly restricted to fault types within the same location. This suggests that the model may be effectively used for the simpler problem of detecting when a fault is present at a specific location, i.e., the model should output a positive classification if and only if a fault (of any type) is present at the same location where the signal fed to model is acquired. This may be seen as the conventional problem of fault detection (without diagnosis), except that localizing the fault is implicitly performed, i.e., whenever a fault is detected by this method, its location is already identified.

To evaluate this fault detection problem, the three fault type labels were condensed (by the max operation) into a single binary label indicating whether a fault has occurred . Correspondingly, the model was modified by taking the average[1] of the probability output of the three

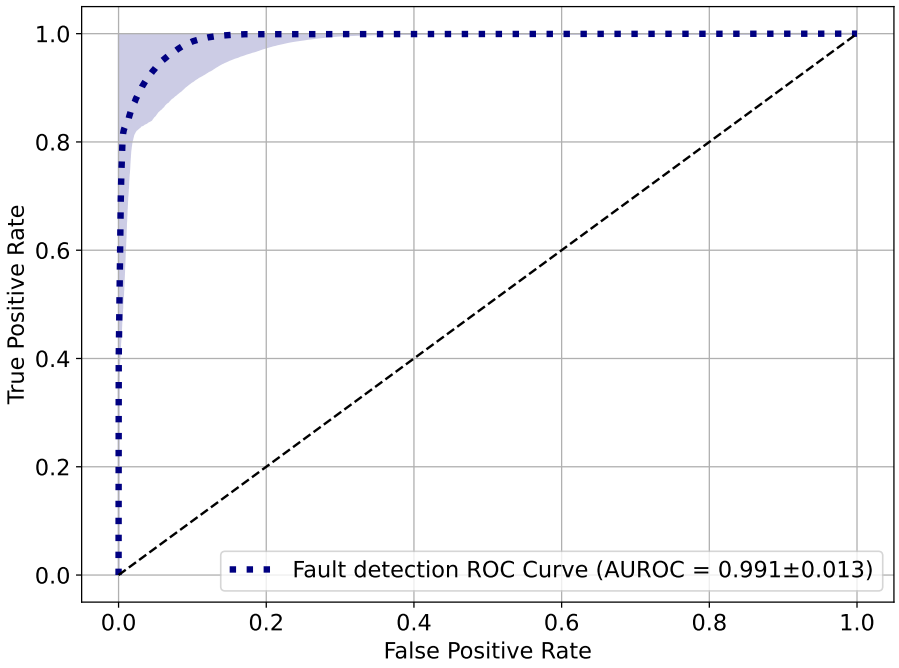
[1] We also experimented with taking the maximum of the probability output, but the results were not improved.

fault-type detectors (akin to a soft-voting ensemble). In other words, for a given sample, at a specific location, the predicted probability of a fault is computed as $\hat{p}_F = (\hat{p}_I + \hat{p}_O + \hat{p}_B)/3$, where $\hat{p}_I$, $\hat{p}_O$ and $\hat{p}_B$ denote, respectively, the corresponding predicted probabilities of inner, outer and ball faults.

It should be emphasized that the fault detection problem cannot be formulated without leakage on the CWRU dataset using a multi-class approach since the healthy class (containing a single-bearing configuration) will necessarily have to be split between train and test sets. Only a multi-label formulation that places all signals from the healthy state on the test set, as pioneered by this work, enables a rigorous evaluation of machine learning models for fault detection using the CWRU dataset.

Figure 16 confirms the observation made from the logits boxplots that the model does indeed perform very well at detecting faults with a AUROC of 0.991 ± 0.013. Considering the real-world use of these models, where the crucial, most important information is whether a component is faulty and needs to be replaced, these results seem promising. While the fault-type information is still important for interpretability and trust in the model predictions, as well as for the decision on when to change the bearings, reliably detecting when and where a bearing fault occurs (without finer diagnosis) may already be useful in practice, where numerous fault types other than bearing faults happen and differentiating what is a bearing fault aggregates significant value.

Figure 16 – ROC curve for fault detection. The solid curve represents the average ROC curve across 30 realizations, while the filled region represents the standard deviation.



Source: Author, 2024

As shown before in Section 4.3, it is possible to choose the operating point of the fault detector by defining an acceptable TPR (e.g., 90% TPR). By defining the acceptable TPR of

90% as the Figure 16 ROC curve operation point, we can take the horizontal average of FPRs across the multiple runs and have an average FPR with similar TPR performance. Through this estimation, we obtain an average FPR of 3.1%.

4.5.1 Confusion matrix analysis

In the previous section, we used multiple model runs to obtain an estimate of the model's performance by taking the horizontal average between runs. With this approach, we can obtain the statistical average and standard deviation of the model's performance ROC curve metrics. However, it does not enable us to determine a single fault detection threshold from the ROC curve operating points, as it is an estimation across multiple realizations, each with different operating points. Having a single threshold is desirable for evaluating the confusion matrix of the model or when considering the real-world application of the model, where a single threshold needs to be defined.

In Subsection 4.1.1, we analyzed the model's outputs through a boxplot and observed that we could define a threshold that could detect faults and separate between the fan and drive end locations. Now, we can confirm this observation by defining a threshold and observing the ROC curve of each bearing location.

To obtain a single threshold, we stacked the predictions from every different model run to obtain a single ROC curve. With this new ROC curve, we chose an operating point and the corresponding threshold to analyze the confusion matrix. Using the same 90% TPR operation point, we obtained the confusion matrices of the fault detection problem separated by each location's detector can be seen in Tables 11 and 12. It should be noted that the TPR is exactly 90% when each TP, FP, FN, and TN in both confusion matrices are summed.

Bear in mind that the numbers in the confusion matrix represent the classification of each individual CWRU experiment segment with the 97% overlap that was described in Section 3.5, hence a large number of predictions.

Table 11 – Fan end location: Fault detection confusion matrix.

		Prediction	
		healthy	**faulty**
Target	**healthy**	112645	2648
	faulty	6904	79129

Table 12 – Drive end location: fault detection confusion matrix.

		Prediction	
		healthy	**faulty**
Target	**healthy**	109205	5508
	faulty	10362	76251

Source: Author, 2024

From the confusion matrices, it is evident that both locations have no problems detecting faults. The signals from the fan end location performing slightly better than those from the drive end location due to the lower number of false positives and negatives.

We can further analyze the model's prediction errors by differentiating the source of each condition with the corresponding type and location. The detailed confusion matrices can

be seen in Tables 13 and 14.

Table 13 – Fan end location: Fault type / location confusion matrix.

			Prediction	
	Location		healthy	faulty
Target at each location	fan	drive		
	healthy	healthy	28564	116
	healthy	ball	28845	0
	healthy	inner	28611	262
	healthy	outer	26625	2270
	ball	healthy	689	28006
	inner	healthy	3583	25101
	outer	healthy	2632	26022

Table 14 – Drive end location: Fault type / location confusion matrix.

			Prediction	
	Location		healthy	faulty
Target at each location	fan	drive		
	healthy	healthy	27460	1220
	ball	healthy	27432	1263
	inner	healthy	27364	1320
	outer	healthy	26949	1705
	healthy	ball	9214	19631
	healthy	inner	693	28180
	healthy	outer	455	28440

Source: Author, 2024

One interesting observation can be made in 14 when looking at the false negatives at the drive end location signals. Almost all of the false negatives are due to the drive ball fault condition being confused with a healthy condition.

5 DISCUSSION AND CONCLUSION

In this work, we propose several modifications to the standard formulation of the bearing fault detection and diagnosis problem using the CWRU dataset in order to bring it closer to real-life industry settings. Hendriks, Dumond & Knox (2022) (and, more recently, also Abburi et al. (2023)) have demonstrated that bearing data leakage profoundly affects the performance of machine learning models, posing a serious shortcoming when the goal is to design algorithms that can detect and diagnose faults in different bearings than those used for training. However, their work has not resonated enough within the published literature, and most papers still use a dataset division containing data leakage, generating over-optimistic results.

Our study takes their approach a step further by proposing a multi-label problem formulation that allows for the detection of different fault types occurring simultaneously at each location, without the need for synchronous acquired signals, while keeping it completely data-leakage-free. By construction, the proposed methodology completely solves the healthy bearing data leakage and imbalance and ensures that the model evaluation is solid and reliable while more accurately reflecting real-world conditions.

Consolidating our framework, we explored various training approaches to enhance model performance. This resulted in our proposed dataset division that takes greater advantage of the CWRU dataset by expanding the train/test split ratio from 1:2 to 2:1 and adding diversity with random fault size configurations for each fault location and type pair. Additionally, we proposed dividing the fault type and location detection into two different problems while using only a single model, effectively doubling our training dataset. In our ablation studies, these proposed approaches have demonstrated significant performance improvements, with the macro average AUROC going from 0.781 to 0.911.

As an application of our approach, we conducted a comparative benchmark using several model architectures of different sizes and levels of complexity, as well as different signal representations as the input to the network. None of the architectures/representations evaluated achieved significant gains over the ResNet18 applied to spectrogram images. However, the WDCNN applied to the power cepstrum obtained competitive results considering its lower computational requirements.

It is worth mentioning that, among previous works applying deep neural networks to the CWRU dataset, we have found many that do not describe their hyperparameter optimization procedure. This raises the possibility that the work was done using test set performance as the objective, which would result in a biased evaluation. Thus, another contribution of our work comes from the specification and use of a model selection and evaluation procedure to minimize such bias.

After fine-tuning the model, we conducted an error analysis that generated interesting insights. We noticed that our model struggled to differentiate between inner and outer faults and had difficulty detecting certain specific fault conditions. However, these errors could mainly be explained by certain aspects of the CWRU dataset discussed in Smith and Randall's thorough

dataset analysis (SMITH; RANDALL, 2015). This adds to the credibility of our results, as one should always be skeptical of surprisingly good results that diverge significantly from what would be expected after a careful analysis by domain experts.

In conclusion, the CWRU dataset remains an important resource for research on rolling bearing fault detection and diagnosis and will likely continue to be widely utilized. We recommend that future researchers using the CWRU dataset with machine learning models for fault detection and diagnosis use our proposed framework, as it offers a realistic and reliable evaluation.

5.1 FUTURE WORK

While this work has successfully addressed its main objectives to develop a robust machine learning-based fault detection and diagnosis system, there are several opportunities for future research and development that could expand upon the findings presented here.

One possibility that was not investigated involves the removal of all healthy signals contained in the bearing configurations used in training, reserving them exclusively for the test set, eliminating potential data leakage that still might appear. Although it is expected that fault contamination from the opposite location makes these healthy samples sufficiently distinct to avoid influencing the training process, testing this hypothesis would provide valuable insights into its validity.

Another promising direction, which has been explored in a recent paper (VIEIRA et al., 2024), involves training traditional machine learning models using the methodology proposed in this work. Comparing their performance with the results obtained from the deep neural networks developed here could offer a deeper understanding of the advantages and limitations of each approach.

Finally, this methodology could be expanded to other datasets with similar dataset size constraints and potentially aid in the design and construction of new datasets for bearing fault diagnosis.

BIBLIOGRAPHY

ABBURI, H. et al. A closer look at bearing fault classification approaches. In: **Annual Conference of the PHM Society**. [S.l.: s.n.], 2023. v. 15, n. 1.

AHMED, H.; NANDI, A. K. **Condition monitoring with vibration signals: Compressive sampling and learning algorithms for rotating machines**. [S.l.]: John Wiley & Sons, 2020.

BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. **IEEE transactions on neural networks**, IEEE, v. 5, n. 2, p. 157–166, 1994.

BHAKTA, K. et al. Fault diagnosis of induction motor bearing using cepstrum-based preprocessing and ensemble learning algorithm. In: IEEE. **2019 International conference on electrical, computer and communication engineering (ECCE)**. [S.l.], 2019. p. 1–6.

BISHOP, C. M.; BISHOP, H. **Deep learning: Foundations and concepts**. [S.l.]: Springer Nature, 2023.

BOGERT, B. P. The quefrency alanysis of time series for echoes: Cepstrum, pseudoautocovariance, cross-cepstrum and saphe cracking. In: **Proc. Symposium Time Series Analysis, 1963**. [S.l.: s.n.], 1963. p. 209–243.

CAI, Y.; TAN, L.; CHEN, J. Evaluation of deep learning neural networks with input processing for bearing fault diagonosis. In: IEEE. **2021 IEEE International Conference on Electro Information Technology (EIT)**. [S.l.], 2021. p. 140–145.

CHEN, Z. et al. Multi-label fault diagnosis based on convolutional neural network and cyclic spectral coherence. **Book of Proceedings Survishno 2019**, INSA Lyon, p. 740–740, 2019.

DOSOVITSKIY, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. **arXiv preprint arXiv:2010.11929**, 2020.

FAWCETT, T. An introduction to roc analysis. **Pattern recognition letters**, Elsevier, v. 27, n. 8, p. 861–874, 2006.

GÉRON, A. **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow**. [S.l.]: " O'Reilly Media, Inc.", 2022.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. http://www.deeplearningbook.org.

HARRIS, T. A. **Rolling Bearing Analysis**. [S.l.]: Wiley, 2001. Google-Books-ID: Pt9SAAAAMAAJ. ISBN 978-0-471-35457-4.

HE, K. et al. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778.

HENDRIKS, J.; DUMOND, P.; KNOX, D. Towards better benchmarking using the CWRU bearing fault dataset. **Mechanical Systems and Signal Processing**, Elsevier, v. 169, p. 108732, 2022.

HOWARD, A. et al. Searching for mobilenetv3. In: **Proceedings of the IEEE/CVF international conference on computer vision**. [S.l.: s.n.], 2019. p. 1314–1324.

JIA, F. et al. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. **Mechanical systems and signal processing**, Elsevier, v. 72, p. 303–315, 2016.

JIN, Y. et al. Actual bearing compound fault diagnosis based on active learning and decoupling attentional residual network. **Measurement**, Elsevier, v. 173, p. 108500, 2021.

KAPOOR, S.; NARAYANAN, A. Leakage and the reproducibility crisis in machine-learning-based science. **Patterns**, Elsevier, v. 4, n. 9, 2023.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

LASI, H. et al. Industry 4.0. **Business & information systems engineering**, Springer, v. 6, p. 239–242, 2014.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Ieee, v. 86, n. 11, p. 2278–2324, 1998.

LEI, Y. et al. Applications of machine learning to machine fault diagnosis: A review and roadmap. **Mechanical Systems and Signal Processing**, Elsevier, v. 138, p. 106587, 2020.

LIU, Z. et al. Swin transformer v2: Scaling up capacity and resolution. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2022. p. 12009–12019.

LIU, Z. et al. A convnet for the 2020s. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2022. p. 11976–11986.

MAINTAINERS, T.; CONTRIBUTORS. **TorchVision: PyTorch's Computer Vision library**. [S.l.]: GitHub, 2016. https://github.com/pytorch/vision.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, p. 115–133, 1943.

MINSKY, M.; PAPERT, S. **Perceptrons: An Introduction to Computational Geometry**. Cambridge, MA, USA: MIT Press, 1969.

MITCHELL, T. **Machine learning**. [S.l.]: McGraw-hill New York, 1997. v. 1.

MURPHY, K. P. **Probabilistic machine learning: an introduction**. [S.l.]: MIT press, 2022.

NANDI, S.; TOLIYAT, H.; LI, X. Condition monitoring and fault diagnosis of electrical motors—a review. **IEEE Transactions on Energy Conversion**, v. 20, n. 4, p. 719–729, dez. 2005. ISSN 1558-0059.

NEUPANE, D.; SEOK, J. Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review. **Ieee Access**, IEEE, v. 8, p. 93155–93178, 2020.

POWERS, D. M. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. **arXiv preprint arXiv:2010.16061**, 2020.

RADOSAVOVIC, I. et al. Designing network design spaces. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2020. p. 10428–10436.

RANDALL, R. B. **Vibration-based condition monitoring: industrial, automotive and aerospace applications**. [S.l.]: John Wiley & Sons, 2021.

RASCHKA, S. Model evaluation, model selection, and algorithm selection in machine learning. **arXiv preprint arXiv:1811.12808**, 2018.

ROSENBLATT, F. **Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms**. [S.l.]: Spartan Books, 1961.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986.

SHEN, J. et al. A deep multi-label learning framework for the intelligent fault diagnosis of machines. **IEEE Access**, IEEE, v. 8, p. 113557–113566, 2020.

SMITH, W. A.; RANDALL, R. B. Rolling element bearing diagnostics using the case western reserve university data: A benchmark study. **Mechanical systems and signal processing**, Elsevier, v. 64, p. 100–131, 2015.

TAN, M.; LE, Q. Efficientnetv2: Smaller models and faster training. In: PMLR. **International conference on machine learning**. [S.l.], 2021. p. 10096–10106.

TSAMARDINOS, I.; RAKHSHANI, A.; LAGANI, V. Performance-estimation properties of cross-validation-based protocols with simultaneous hyper-parameter optimization. **International Journal on Artificial Intelligence Tools**, World Scientific, v. 24, n. 05, p. 1540023, 2015.

TU, Z. et al. Maxvit: Multi-axis vision transformer. In: SPRINGER. **European conference on computer vision**. [S.l.], 2022. p. 459–479.

VIEIRA, J. P. et al. Bearing fault diagnosis using machine learning and a novel set of fault-related spectral features. In: SBC. **Anais do XXI Encontro Nacional de Inteligência Artificial e Computacional**. [S.l.], 2024.

YU, C. et al. Multi-label fault diagnosis of rolling bearing based on meta-learning. **Neural Computing and Applications**, Springer, v. 33, p. 5393–5407, 2021.

ZHANG, Q. et al. Cjc-net: A cyclical training method with joint loss and co-teaching strategy net for deep learning under noisy labels. **Information Sciences**, Elsevier, v. 579, p. 186–198, 2021.

ZHANG, S. et al. Deep learning algorithms for bearing fault diagnostics—a comprehensive review. **IEEE Access**, IEEE, v. 8, p. 29857–29881, 2020.

ZHANG, W. et al. A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. **Mechanical systems and signal processing**, Elsevier, v. 100, p. 439–453, 2018.

ZHANG, W.; LI, X.; DING, Q. Deep residual learning-based fault diagnosis method for rotating machinery. **ISA transactions**, Elsevier, v. 95, p. 295–305, 2019.

ZHANG, W. et al. A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. **Sensors**, MDPI, v. 17, n. 2, p. 425, 2017.