



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Vicente Knobel Borges

**Análise de transformações bidimensionais de sinais de vibração para
classificação de defeitos e falhas em rolamentos usando Redes Neurais
Convolucionais**

Florianópolis
2023

Vicente Knobel Borges

**Análise de transformações bidimensionais de sinais de vibração para
classificação de defeitos e falhas em rolamentos usando Redes Neurais
Convolucionais**

Relatório final da disciplina DAS5511 (Projeto de Fim de Curso) como Trabalho de Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Catarina em Florianópolis.

Orientador: Prof. Danilo Silva, Ph.D.

Florianópolis

2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Borges, Vicente Knobel

Análise de transformações bidimensionais de sinais de vibração para classificação de defeitos e falhas em rolamentos usando Redes Neurais Convolucionais / Vicente Knobel Borges ; orientador, Danilo Silva, 2023.

63 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia de Controle e Automação, Florianópolis, 2023.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Falha de rolamento. 3. Manutenção Preditiva. 4. Aprendizado profundo. I. Silva, Danilo. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Controle e Automação. III. Título.

Vicente Knobel Borges

**Análise de transformações bidimensionais de sinais de vibração para
classificação de defeitos e falhas em rolamentos usando Redes Neurais
Convolucionais**

Esta monografia foi julgada no contexto da disciplina DAS5511 (Projeto de Fim de Curso) e aprovada em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação

Florianópolis, 11 de Julho de 2023.

Prof. Hector Bessa Silveira, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Danilo Silva, Ph.D.
Orientador
UFSC/CTC/EEL

Prof. Danilo Silva, Ph.D.
Supervisor
UFSC/CTC/EEL

Gabriel Thaler, ME.
Avaliador
UFSC

Prof. Eduardo Camponogara, Dr.
Presidente da Banca
UFSC/CTC/DAS

Aos meus pais, Roxana e Thomaz.

AGRADECIMENTOS

Primeiramente agradeço aos meus pais, Roxana e Thomaz, pelo apoio, incentivo e carinho. Amo vocês!

Agradeço ao professor Danilo Silva, pela orientação séria e compromisso com o meu aprendizado, bem como a oportunidade de trabalhar neste projeto.

Ao meu colega de projeto Rodrigo, por sua paciência e disponibilidade, bem como sua extensiva documentação do desenvolvimento das atividades.

À minha namorada, Sara, agradeço pela compreensão, amor e apoio, principalmente nos momentos mais difíceis e incertos. Sua presença e otimismo ajudaram-me a seguir em frente.

À Dynamox pela parceria e apoio concedido pela equipe. Aos professores da Universidade Federal de Santa Catarina que me forneceram todas as bases necessárias para a realização deste trabalho, agradeço com profunda admiração pelos ensinamentos passados com diligência e entusiasmo.

A todos os demais amigos, colegas e familiares que, direta ou indiretamente, me auxiliaram, motivaram e apoiaram na realização deste trabalho, meu muito obrigado.

*“Todos podem ver as táticas de minhas conquistas, mas ninguém consegue discernir a estratégia que gerou as vitórias.”
(Sun Tzu)*

DECLARAÇÃO DE PUBLICIDADE

Florianópolis, 11 de Julho de 2023.

Na condição de representante do Grupo de Pesquisa em Aprendizado de Máquina e Aplicações (GAMA) no qual o presente trabalho foi realizado, declaro não haver ressalvas quanto ao aspecto de sigilo ou propriedade intelectual sobre as informações contidas neste documento, que impeçam a sua publicação por parte da Universidade Federal de Santa Catarina (UFSC) para acesso pelo público em geral, incluindo a sua disponibilização *online* no Repositório Institucional da Biblioteca Universitária da UFSC. Além disso, declaro ciência de que o autor, na condição de estudante da UFSC, é obrigado a depositar este documento, por se tratar de um Trabalho de Conclusão de Curso, no referido Repositório Institucional, em atendimento à Resolução Normativa nº 126/2019/CUn.

Por estar de acordo com esses termos, subscrevo-me abaixo.



Documento assinado digitalmente

Danilo Silva

Data: 12/07/2023 11:59:06-0300

CPF: ***,557.924-**

Verifique as assinaturas em <https://v.ufsc.br>

Prof. Danilo Silva, Ph.D.

Grupo de Pesquisa em Aprendizado de Máquina e Aplicações

RESUMO

O avanço tecnológico dos sensores e a suas aplicações para identificar possíveis falhas em máquinas industriais tem transformado o setor da manutenção preventiva. O uso de sensores tem possibilitado a manutenção baseada em dados, o que significa que é possível monitorar o funcionamento dos ativos industriais. Este trabalho tem por objetivo desenvolver e validar modelos de Redes Neurais Convolucionais baseados nos dados de sensores, viabilizando o diagnóstico automatizado de defeitos em rolamentos. A metodologia de desenvolvimento CRISP-DM foi empregada como forma de apoio para o processo de desenvolvimento dos modelos. Uma nova metodologia de avaliação dos modelos em cima de um conjunto de dados amplamente utilizado em pesquisas na área foi implementada evitando problemas de vazamento, proporcionando uma maior diversidade ao conjunto de treino e permitindo uma maior robustez à mudança de domínio. Foi explorado o uso de diferentes representações visuais do sinal captado, que agregam informações como a frequência em um dado instante do tempo, transformando um objeto vetorial unidimensional em uma representação matricial bidimensional. A representação dos sinais por meio de cepstrogramas resultou em um desempenho de classificação significativamente superior ao atual estado da arte. Modelos explorados no decorrer desse trabalho podem habilitar gestores do setor da manutenção a tomar decisões mais acertadas com base no diagnóstico de possíveis falhas, permitindo uma gestão mais eficiente do chão de fábrica.

Palavras-chave: Falha de rolamento. Manutenção Preditiva. Aprendizado profundo.

ABSTRACT

The technological advancement of sensors and their applications for identifying potential failures in industrial machines has transformed the preventive maintenance sector. The use of sensors has enabled data-based maintenance, which means that it is possible to monitor the functioning of industrial assets. The objective of this work is to develop and validate Convolutional Neural Network models based on sensor data, enabling the automated diagnosis of defects in bearings. The CRISP-DM development methodology was employed as a support for the model development process. A new evaluation methodology for the models on a widely used dataset in research in the area was implemented, avoiding leakage issues, providing greater diversity to the training set, and allowing greater robustness to domain variation. The use of different visual representations of the captured signal, which aggregate information such as frequency at a given moment in time, transforming a one-dimensional vectorial object into a two-dimensional matrix representation, was explored. The representation of signals through cepstograms resulted in significantly superior classification performance compared to the current state of the art. Models explored in this work can enable maintenance sector managers to make more accurate decisions based on the diagnosis of potential failures, allowing for more efficient factory floor management.

Keywords: Bearing failure. Predictive maintenance. Deep learning.

LISTA DE FIGURAS

Figura 1 – Locais das falhas nos rolamentos	21
Figura 2 – Dimensões das falhas em elemento rolante	21
Figura 3 – Extremidades do motor	22
Figura 4 – Sistema de Captura de dados dos ensaios.	22
Figura 5 – Dados de vibração de um ensaio.	23
Figura 6 – Base da divisão de rolamentos entre subconjuntos na estrutura original (vermelho) e na estrutura proposta (azul) para uma única classe de falha.	24
Figura 7 – Resultados do artigo <i>Towards better benchmarking using the CWRU bearing fault dataset no original framework</i>	26
Figura 8 – Resultados do artigo <i>Towards better benchmarking using the CWRU bearing fault dataset no proposed framework</i>	26
Figura 9 – Diferentes disposições de janela e sobreposição no espectrograma de um sinal.	29
Figura 10 – Diferentes wavelets mães usadas na CWT.	30
Figura 11 – Resolução tempo-frequência de representações de sinais.	31
Figura 12 – Fluxograma de criação de um cepstrograma.	32
Figura 13 – Exemplo de geração de um cepstrograma.	33
Figura 14 – Fluxograma de criação de um espectrograma de modulação.	34
Figura 15 – Exemplo de geração de um espectrograma de modulação.	34
Figura 16 – Funcionamento Conceitual da pipeline estabelecida	35
Figura 17 – Nova proposta de uso dos conjuntos	37
Figura 18 – Validação Cruzada Aninhada	41
Figura 19 – Espectrograma das classes de falha com carga de 3 HP	45
Figura 20 – Escalograma das classes de falha com carga de 3 HP	49
Figura 21 – Cepstrograma das classes de falha com carga de 3 HP	52
Figura 22 – Espectrograma de modulação das classes de falha com carga de 3 HP	54
Figura 23 – Matrizes de Confusão da <i>pipeline</i> do cepstrograma	57

LISTA DE TABELAS

Tabela 1 – Entradas das redes do artigo <i>Towards better benchmarking</i>	27
Tabela 2 – Resultados do laboratório e do artigo estudado.	36
Tabela 3 – Resultados com a nova proposta de uso dos conjuntos.	37
Tabela 4 – Hiperparâmetros de entrada e treino	41
Tabela 5 – Grade de dimensões possíveis para o espectrograma.	43
Tabela 6 – Resultados finais – otimização com espectrograma.	44
Tabela 7 – Rankings das Wavelets	47
Tabela 8 – Resultados finais – otimização com escalograma.	48
Tabela 9 – Resultados finais – otimização com cepstrograma.	51
Tabela 10 – Parametrizações testadas para o espectrograma de modulação. . .	53
Tabela 11 – Resultados finais – otimização com espectrograma de modulação. .	53
Tabela 12 – Resultados finais com a Validação Cruzada Aninhada.	55
Tabela 13 – Novas transformadas na estrutura anterior	56

LISTA DE ABREVIATURAS E SIGLAS

CNN	<i>Convolutional Neural Networks</i> (Redes Neurais Convolucionais)
CWT	<i>Continuous wavelet transform</i> (Transformada de Wavelet Continua)
GAMA	Grupo de Pesquisa em Aprendizado de Máquina e Aplicações
LCS	Laboratório de Comunicações, Processamento de Sinais e Aprendizado de Máquina
ML	<i>Machine Learning</i> (Aprendizado de Máquina)
STFT	<i>Short-Time Fourier Transform</i> (Transformada de Fourier de Curto Termo)

LISTA DE SÍMBOLOS

\mathcal{F} Transformada de Fourier

SUMÁRIO

1	INTRODUÇÃO	15
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	APRENDIZADO DE MÁQUINA	17
2.1.1	Treinamento	17
2.1.2	Generalização do modelo	18
2.1.3	Hiperparâmetros e o conjunto de validação	19
2.1.4	Arquitetura de Redes Neurais Convolucionais	19
2.2	<i>CWRU BEARING FAULT DATASET</i>	20
2.2.1	Uso do Dataset	23
2.3	REPRESENTAÇÕES VISUAIS DE SINAIS UNIDIMENSIONAIS	27
2.3.1	Espectrograma	27
2.3.2	Escalograma	28
2.3.3	Cepstrograma	31
2.3.4	Espectrograma de modulação	33
3	PESQUISA ANTECEDENTE	35
4	METODOLOGIAS	39
4.1	METODOLOGIA DE DESENVOLVIMENTO	39
4.2	METODOLOGIA DE OTIMIZAÇÃO	39
4.2.1	Divisão de Conjuntos de Treino e Teste	39
4.2.2	Conjunto de validação para otimização de Hiperparâmetros	40
5	RESULTADOS	43
5.1	OTIMIZAÇÃO DE HIPERPARÂMETROS	43
5.1.1	Espectrograma	43
5.1.2	Escalograma	46
5.1.3	Cepstrograma	50
5.1.4	Espectrograma de modulação	53
5.2	AVALIAÇÃO DOS MODELOS	55
5.2.1	Avaliação no conjunto de teste	55
5.2.2	Performance na estrutura anterior	56
5.2.3	Matriz de Confusão	57
6	CONCLUSÃO	58
	REFERÊNCIAS	60

1 INTRODUÇÃO

O monitoramento da condição atual de uma máquina é uma parte importante para a implementação do conjunto de técnicas da manutenção preditiva, que trazem vantagens econômicas quando implementadas em diversas indústrias e processos (RANDALL, R., 2011). Neste campo, Diagnóstico de Falha Inteligente (DFI) refere-se a aplicações de teorias de *machine learning* para diagnóstico de falhas em máquinas. Esta é uma maneira promissora de liberar a contribuição do trabalho humano e reconhecer automaticamente os estados de saúde das máquinas, um tema que atraído muita atenção nas últimas três décadas (LEI *et al.*, 2020). Neste trabalho, exploramos o uso de redes neurais convolucionais para a classificação de falhas em rolamentos em motores elétricos com base em dados de vibração coletados no chassi dos mesmos, avaliando diferentes maneiras de apresentar essa informação de maneira bidimensional, incorporando representações tempo-frequência, dentre outros, para uma classificação mais acurada.

Técnicas de DFI se enquadram no escopo da engenharia de controle e automação, integrando conhecimentos gerais de processamento de sinais, programação e conexão a um sistema maior de gestão empresarial, tanto no escopo de chão de fábrica quanto na economia de empresa como um todo. Em um contexto da nascente indústria 4.0, o assunto de gestão de equipamento baseada em dados se mostra de extrema importância em geral e em particular para o controle de rolamentos, um componente praticamente ubíquo em acionamentos elétricos na indústria, que requer frequentes trocas para preservar a saúde do motor como um todo.

No segundo semestre de 2022, o bolsista pesquisador ingressou no Grupo de Pesquisa em Aprendizado de Máquina e Aplicações (GAMA), uma entidade dentro do Laboratório de Comunicações, Processamento de Sinais e Aprendizado de Máquina (LCS) da Universidade Federal de Santa Catarina. Em parceria com a empresa Dynamox SA, o grupo deu início a uma frente de pesquisa onde se buscou reproduzir estudos com o uso de aprendizado de máquina para o diagnóstico de falhas em máquinas rotativas. Após uma rodada de revisão bibliográfica, o enfoque da equipe de pesquisa foi a reprodução dos modelos e *pipelines* de processamento de dados do artigo *Towards better benchmarking using the CWRU bearing fault dataset* (HENDRIKS; DUMOND; KNOX, 2022).

Neste artigo, uma variedade de redes neurais convolucionais é usada para classificar falhas em rolamentos, treinadas usando os dados do conjunto de dados público da Case Western Reserve University (CASE. . . , 2021) e utilizando um paradigma autoral onde o foco é a acurácia de classificação de defeitos em rolamentos não incluídos no conjunto de treino do modelo. No processo de reprodução, o enfoque foi em cima dos modelos com a maior acurácia de teste média encontrada.

Com uma *pipeline* de processamento de dados e treinamento de modelos estabelecida, mas acurácias de teste abaixo das reportadas no artigo, no início de 2023 a frente de pesquisa foi redirecionada para a investigação de transformações bidimensionais do sinal do acelerômetro que pudessem compor *pipelines* de classificação mais acuradas. Esse trabalho cobre as atividades do bolsista pesquisador no curso de implementar essas transformações e aplicar uma nova metodologia de otimização, usada para calibrar essas transformações visando melhores acurácias de classificação em suas respectivas *pipelines*. O objetivo deste trabalho é encontrar parâmetros da representação bidimensional de sinais de vibração, bem como parâmetros de treinamento de redes neurais convolucionais que tenham uma acurácia de classificação superior às reportadas no artigo *Towards better benchmarking using the CWRU bearing fault dataset* (HENDRIKS; DUMOND; KNOX, 2022).

Esse trabalho foi realizado em colaboração com o membro da equipe de pesquisa Rodrigo Kobashikawa Rosa, Eng., e o desenvolvimento e resultados encontrados serão também usados para compor sua dissertação de mestrado.

O presente trabalho foi organizado em 6 capítulos que esclarecem o âmbito do problema estudado, as ferramentas e sistemas de informação aplicados e os resultados validados. O primeiro capítulo é destinado à introdução, exposição dos objetivos e contextualização da pesquisa realizada. No segundo capítulo, é dada a fundamentação teórica de conceitos gerais de aprendizado de máquina e redes neurais convolucionais, seguida de uma introdução ao conjunto de dados utilizado e seu uso na literatura, finalizando com a conceitualização das transformadas bidimensionais usadas no trabalho. O terceiro capítulo explora a pesquisa antecedente realizada pelo laboratório na área, que fundamenta as atividades subsequentes. No quarto capítulo detalhamos as metodologias de desenvolvimento e otimização de modelos utilizadas. No quinto capítulo, detalhamos os procedimentos de otimização aplicados as *pipelines* de classificação e seu desempenho de validação resultante em conjuntos de dados de validação e teste, discutindo os resultados apresentados. No sexto capítulo, é apontada a conclusão do trabalho e direcionamentos para desenvolvimentos futuros na área.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos teóricos básicos do trabalho. Primeiramente, será abordado o conceito de aprendizado de máquina, um campo da inteligência artificial que se preocupa com o desenvolvimento de algoritmos capazes de aprender a partir de dados, seguido por uma introdução à arquitetura de redes neurais convolucionais. Em seguida, serão apresentados os conceitos da estrutura do *CWRU bearing fault dataset* e seu uso no desenvolvimento de modelos de previsão de falhas. Por fim, serão introduzidas as diferentes representações visuais utilizadas para inserir os dados originalmente unidimensionais de vibração em redes convolucionais com entrada bidimensional.

2.1 APRENDIZADO DE MÁQUINA

O aprendizado de máquina é um campo da inteligência artificial que se preocupa com o desenvolvimento de algoritmos capazes de “aprender” a partir de dados. Esses algoritmos são treinados com conjuntos de dados que contêm exemplos de entrada e saída desejada, de modo a criar modelos que podem ser aplicados para prever a saída para novos conjuntos de dados.

2.1.1 Treinamento

O processo de treinamento em aprendizado de máquina envolve ajustar os parâmetros de um algoritmo com base em um conjunto de dados de treino. O objetivo é encontrar valores para os parâmetros que produzam o melhor desempenho do algoritmo em relação ao conjunto de treino com base em uma métrica estabelecida, que informa ao algoritmo quão “certa” ou “desejada” é aquela saída.

Para iniciar o treinamento, o algoritmo é inicializado com valores iniciais para os parâmetros. Em seguida, o conjunto de treino é fornecido ao algoritmo, que processa cada exemplo do conjunto, comparando a saída produzida pelo algoritmo com a saída desejada fornecida no conjunto de treino com base na métrica dada. Isso permite que o algoritmo calcule um “erro” que representa o quão bem ou mal o modelo está se comportando em relação ao conjunto de treino.

Com base no erro calculado, o algoritmo ajusta os parâmetros de modo a minimizar o erro. Isso é geralmente feito usando técnicas de otimização, como o gradiente descendente, que buscam encontrar valores ótimos para os parâmetros que minimizem o erro. O processo é repetido iterativamente até que o erro atinja um nível aceitável ou até que se atinja um número máximo de iterações. (LECUN; BENGIO; HINTON, G., 2015)

Ao final do treinamento, o algoritmo terá ajustado os parâmetros de modo a

produzir o melhor desempenho possível em relação ao conjunto de treino. Isso não garante necessariamente que o modelo terá bom desempenho em dados que ele não viu durante o treino, mas a ideia é que o modelo tenha aprendido a generalizar o comportamento a partir do conjunto de treino.

É importante notar que o processo de treinamento envolve uma série de escolhas que podem afetar o desempenho do modelo. Isso inclui escolhas como o algoritmo a ser usado, a forma como os dados são preparados e divididos entre conjunto de treino e teste, e os valores iniciais dos parâmetros. Escolhas adequadas podem levar a modelos que aprendem eficientemente a partir dos dados, enquanto escolhas inadequadas podem resultar em modelos que não aprendem ou que apresentam *overfitting* ou *underfitting*.

Overfitting ocorre quando um modelo é “viciado” nos dados de treino, ou seja, quando ele aprende de forma muito detalhada as características específicas do conjunto de treino, mas não é capaz de generalizar o comportamento para dados novos. Isso pode levar a um modelo que apresenta um desempenho excelente no conjunto de treino, mas um desempenho muito pior em dados que ele não viu durante o treino. *Underfitting* ocorre quando um modelo não é capaz de aprender os padrões presentes nos dados de treino. Isso pode levar a um modelo que apresenta um desempenho ruim tanto no conjunto de treino quanto em dados novos.

2.1.2 Generalização do modelo

Um dos principais desafios no aprendizado de máquina é garantir que o modelo seja capaz de generalizar o comportamento aprendido para novos casos, em vez de apenas memorizar o conjunto de treino. Para avaliar a capacidade de generalização do modelo, é comum utilizar um conjunto de teste independente do conjunto de treino. Esse conjunto de teste é usado para avaliar o desempenho do modelo em dados que ele não viu durante o treino, o que é importante para garantir que o modelo não sofre de *overfitting*.

No entanto, é possível que um conjunto de teste muito similar ao conjunto de treino não forneça uma avaliação precisa do desempenho do modelo em dados mais variados. Isso pode levar a incerteza se o modelo está realmente preparado para lidar com casos mais complexos e variados. Por exemplo, imagine que se desenvolva um modelo de aprendizado de máquina para prever o preço de um imóvel com base em características como tamanho, localização e idade. Se o conjunto de treino e teste consistir apenas em imóveis de uma determinada região e com características semelhantes, o modelo pode aprender a prever preços precisos nesse conjunto de dados, mas pode não ser capaz de generalizar para imóveis com características diferentes em outras regiões.

Para evitar esse tipo de problema, é importante garantir que o conjunto de

teste seja representativo da diversidade de dados que o modelo pode encontrar em situações reais. Isso pode incluir dados de diferentes regiões, com características diferentes e com variações na saída desejada. Além disso, é importante avaliar o desempenho do modelo em diferentes conjuntos de teste, para garantir que ele esteja realmente preparado para lidar com a variedade de dados que pode encontrar em situações reais.

2.1.3 Hiperparâmetros e o conjunto de validação

A otimização de hiperparâmetros é uma etapa importante no desenvolvimento de modelos de aprendizado de máquina, pois envolve a busca pelos melhores valores para os parâmetros que não são aprendidos pelo algoritmo, mas que influenciam seu desempenho. Esses hiperparâmetros podem incluir a taxa de aprendizado, o número de camadas ocultas em uma rede neural, ou no nosso caso, a parametrização das transformações bidimensionais que aplicamos no sinal de vibração do acelerômetro.

A razão pela qual não devemos otimizar os hiperparâmetros no conjunto de teste é que isso pode levar ao *overfitting* dos hiperparâmetros aos dados de teste. O conjunto de teste deve ser reservado para avaliar o desempenho final do modelo, ou seja, para obter uma estimativa não enviesada de quão bem o modelo generaliza para dados não vistos.

Se otimizarmos os hiperparâmetros no conjunto de teste, corremos o risco de escolher hiperparâmetros que sejam específicos para os dados do conjunto de teste, em vez de capturar padrões mais gerais que se aplicam a dados não vistos. Isso pode resultar em um modelo que funciona muito bem nos dados de teste, mas que não generaliza bem para novos dados.

Em vez disso, é de praxe dividir os dados disponíveis em três conjuntos: treinamento, validação e teste. O conjunto de treinamento é usado para treinar o modelo, o conjunto de validação é usado para ajustar os hiperparâmetros e o conjunto de teste é usado apenas para avaliar o desempenho final do modelo após a otimização dos hiperparâmetros.

Ao ajustar os hiperparâmetros no conjunto de validação, podemos avaliar diferentes combinações de hiperparâmetros e escolher aqueles que resultam em melhor desempenho na validação. Essa abordagem nos permite encontrar um conjunto de hiperparâmetros que otimiza o desempenho geral do modelo, sem viciar a avaliação final no conjunto de teste.

2.1.4 Arquitetura de Redes Neurais Convolucionais

A arquitetura de *Convolutional Neural Networks* (Redes Neurais Convolucionais) (CNN) é uma classe de modelos de aprendizado profundo que foram projetadas para processar dados de imagem. Elas se inspiram na estrutura do córtex visual do cérebro

humano e são capazes de aprender características específicas a partir de imagens através da aplicação de filtros e camadas de pooling.

Uma das principais vantagens das CNNs é que elas são capazes de aprender características diretamente a partir dos dados de entrada, sem a necessidade de pré-processamento.

Uma das redes neurais convolucionais mais bem-sucedidas e influentes é a ResNet (Rede Residual), proposta por Kaiming He *et al.* (2015). A ResNet introduziu o conceito de “resíduos” como uma forma de resolver o problema do “desvanecimento do gradiente”, que ocorre quando os gradientes de uma rede neural se tornam muito pequenos durante o treinamento, o que dificulta a convergência do modelo.

A ResNet resolve o problema do "desvanecimento do gradiente" adicionando uma “conexão residual” entre as camadas da rede, permitindo que o sinal de entrada passe diretamente para a saída sem passar por todas as camadas intermediárias. Isso permite que a rede aprenda uma função de identidade como uma tarefa fácil, o que ajuda a manter o sinal de gradiente forte durante o treinamento.

Além disso, a ResNet também introduziu o uso de camadas *bottleneck*, ou “funil”, que são camadas com menos filtros do que as camadas anteriores e posteriores. Isso ajuda a reduzir o número de parâmetros na rede e a evitar o overfitting.

Outra característica interessante da ResNet é o seu uso de camadas de “ampliação de escala”, que aumentam o tamanho dos mapas de características entre as camadas da rede. Isso permite que a ResNet capture características mais complexas e abstrações mais altas, o que é útil para tarefas como a classificação de imagens.

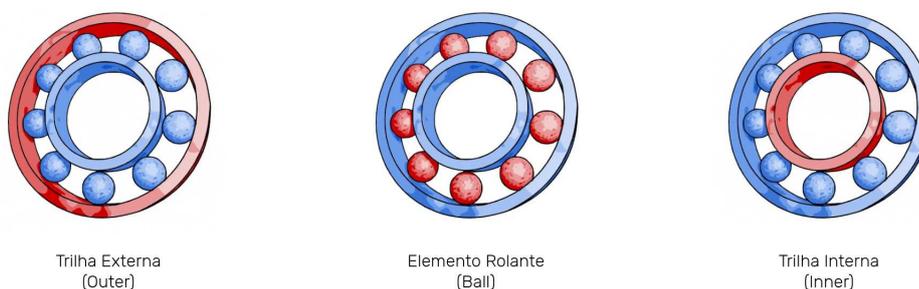
2.2 CWRU BEARING FAULT DATASET

O *CWRU bearing fault dataset* é um conjunto de dados de falhas em rolamentos criado pela *Case Western Reserve University* (CWRU) para fins de pesquisa. Ele é amplamente utilizado em estudos sobre detecção de falhas em rolamentos e é considerado um *benchmark* para o desenvolvimento de técnicas de análise de falhas em rolamentos.

O conjunto de dados consiste em sinais de vibração de rolamentos em diferentes condições de falha. As falhas foram induzidas artificialmente nos rolamentos usando usinagem por eletroerosão. As falhas foram induzidas no elemento rolante, na trilha interna e na trilha externa do rolamento. A Figura 1 ilustra esses elementos do rolamento destacados em vermelho.

As falhas nos rolamentos foram dimensionadas em 4 categorias de magnitude, representando de maneira discreta condições diferentes de deterioração nas partes estabelecidas. As falhas tem tamanhos de 7, 14 e 21 milipolegadas (em inglês, essa unidade é denominada *mils*). A figura 2 mostra as 3 categorias de defeito destacadas em vermelho em um elemento rolante de 8 mm, padrão do rolamento da empresa SKF

Figura 1 – Locais das falhas nos rolamentos



Fonte: Arquivo pessoal.

usado nos testes. Se ressalta que a mesmas categorias de defeito foram aplicadas a trilha interna e a trilha externa de outros rolamentos. Cada rolamento tem defeitos em apenas um local e apenas uma magnitude de defeito por ensaio.

Figura 2 – Dimensões das falhas em elemento rolante

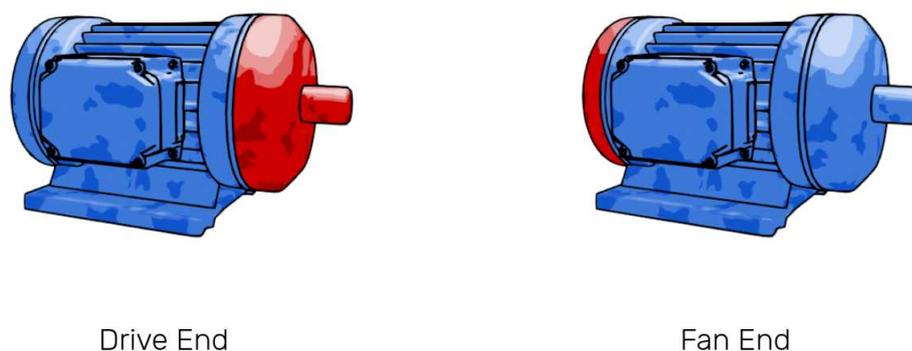


Fonte: Arquivo pessoal.

Outro critério para a geração de estados de falha é qual extremidade do motor o rolamento defeituoso está. Defeitos do lado do acionamento do motor, onde se extruda o eixo que move a carga, são denominados de defeitos no *drive end*. Defeitos do lado oposto, onde geralmente um acionamento industrial tem um ventilador acoplado para refrigeração, são denominados defeitos no *fan end*. A figura 3 aponta as duas extremidades em um diagrama de motor respectivamente destacadas em vermelho.

Um critério final para a aquisição de estados de falha é a carga no motor. Um motor rodando sem carga tem sinais e dinâmicas internas diferentes de um motor carregado com uma força contrária a seu giro. Os ensaios foram realizados com um transdutor/sensor de torque aplicando forças de 0 (giro livre), 1, 2 e 3 *hp* no motor, e os sinais de vibração dos ensaios foram extraídos por dois acelerômetros com amostragem de

Figura 3 – Extremidades do motor

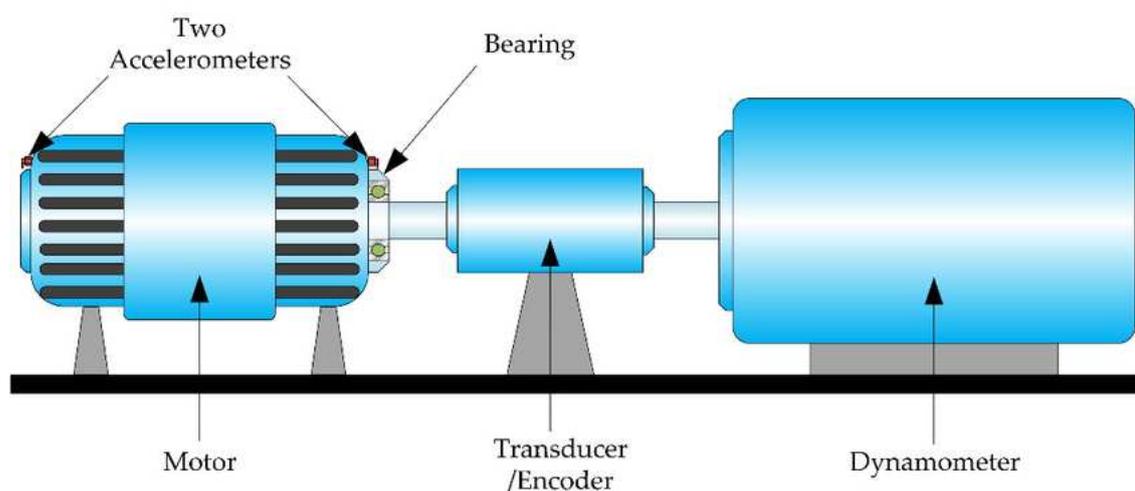


Fonte: Arquivo pessoal.

12 mil amostras por segundo, instalados no *fan end* e no *fan end*. Adicionalmente, um dinamômetro mediu a velocidade do motor em RPM e em alguns ensaios, mediu-se a vibração por meio de um acelerômetro fixado na base do motor, mas para os propósitos deste trabalho e suas abordagens, esses sinais são considerados irrelevantes.

A figura 4 mostra o sistema de captura de dados e indução de torque.

Figura 4 – Sistema de Captura de dados dos ensaios.



Fonte: Qiuchen He *et al.* (2022).

Considerando as variáveis independentes para a geração de um ensaio, um ensaio é definido por 10 segundos de operação do motor com uma combinação de:

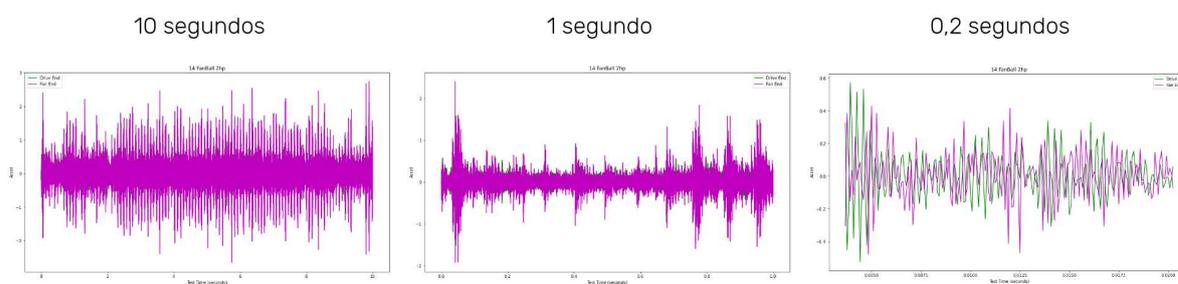
- Local da falha interna no rolamento (*outer*, *ball* ou *inner*)
- Magnitude da falha interna (7, 14 ou 21 *mils*)

- Local do rolamento falho no motor (*fan end* ou *drive end*)
- Torque aplicado no motor (0, 1, 2 ou 3 hp)

Totalizando um total de 72 ensaios possíveis, aos quais se somam 4 ensaios do motor em condição de normalidade, que são dados pelo motor sem condições de falha nos rolamentos e os 4 torques definidos.

A figura 5 mostra uma combinação dessas variáveis independentes, mostrando os sinais de vibração capturados no *fan end* (magenta) e no *drive end* (verde) com uma falha interna do tipo *ball* no rolamento do *fan end* com um defeito de magnitude 14 *mils* e uma carga no motor de 2 hp.

Figura 5 – Dados de vibração de um ensaio.



Fonte: Arquivo pessoal.

2.2.1 Uso do Dataset

No artigo *Towards better benchmarking using the CWRU bearing fault dataset* (HENDRIKS; DUMOND; KNOX, 2022), os autores fazem apontamentos sobre a condição de estado-da-arte do uso do *CWRU bearing fault dataset* para o treinamento de modelos de aprendizado de máquina e fazem recomendações para um uso mais adequado para treinar e fazer o *benchmarking* desses modelos.

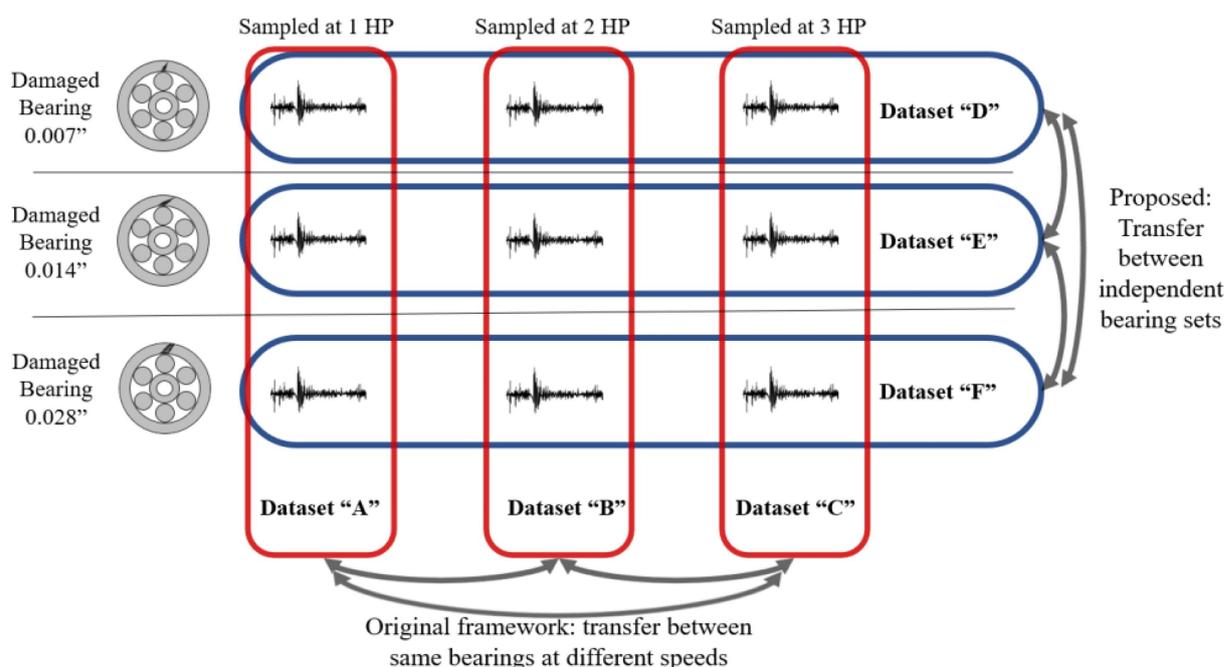
No que foi denominado o *Original Framework*, identificado como o usado em diversos artigos no campo de detecção de falha, dentre eles os trabalhos de Zhang *et al.* (2017), Zhu *et al.* (2019) e Hoang e Kang (2019), os ensaios do conjunto de dados são divididos pela carga em hp usada no ensaio, e cada subconjunto é usado como a parte de treino e a parte de teste dos algoritmos de aprendizado de máquina arbitrariamente. Esses *subsets* são tradicionalmente referenciados como A, B e C, e abrangem as cargas de 1, 2 e 3 hp, com os dados do motor sem carga usualmente desconsiderados.

Os autores de *Towards better benchmarking* argumentam que embora no *Original Framework* (Estrutura Original) se usem medições independentes em cada um

dos conjuntos de dados A, B e C, essas medições se originam de um conjunto compartilhado de rolamentos danificados. Além disso, as mudanças reais de RPM nos conjuntos de dados (decorrentes das cargas diferentes) não são significativas. Isso torna a transferência de conhecimento entre esses conjuntos de dados relativamente fácil para os algoritmos de aprendizado de máquina resolverem. Isso é uma instância de um conjunto de validação muito similar ao conjunto de treino, cujos problemas são detalhados na subseção 2.1.2 deste trabalho.

Os autores propõem o tautologicamente nomeado *Proposed Framework* (Estrutura Proposta), onde os ensaios do conjunto de dados são divididos pela magnitude da falha interna do rolamento, com as magnitudes de 7, 14 e 21 *mils* sendo utilizadas como critério para a separação em conjuntos denominados D, E e F. Quando usamos essa divisão para conjuntos de treino e validação, podemos ter certeza de que o modelo nunca viu as dinâmicas temporais daquele conjunto de rolamentos em seu treino, garantindo uma boa separação e previsões mais coerentes com a natureza dos dados no estágio de implementação, pois quando um rolamento falhar, não vai ser com os modos de falha específicos que foram vistos no conjunto de treinamento. A figura 6 mostra as diferenças na divisão do conjunto de dados entre o *Original* e o *Proposed Framework*.

Figura 6 – Base da divisão de rolamentos entre subconjuntos na estrutura original (vermelho) e na estrutura proposta (azul) para uma única classe de falha.



Fonte: Hendriks, Dumond e Knox (2022).

Para demonstrar as propriedades inerentes as distribuições de subconjuntos, os

autores elaboraram o seguinte problema de classificação de classe de falha: modelos de aprendizado de máquina classificam como classes de falhas os dados dos defeitos como defeitos no *Fan End* ou *Drive End* e como defeitos na Trilha Externa, na Trilha Interna e no Elemento rolante desses rolamentos. Assim temos seis classes de falhas:

- Defeito na Trilha Interna do rolamento de *Fan End* (*FanInner*)
- Defeito na Trilha Externa do rolamento de *Fan End* (*FanOuter*)
- Defeito no Elemento Rolante do rolamento de *Fan End* (*FanBall*)
- Defeito na Trilha Interna do rolamento de *Drive End* (*DriveInner*)
- Defeito na Trilha Externa do rolamento de *Drive End* (*DriveOuter*)
- Defeito na no Elemento Rolante do rolamento de *Drive End* (*DriveBall*)

Se soma a estas classes de falha uma sétima classe de normalidade, que o modelo deve classificar os dados do funcionamento normal do motor com ela, distinguindo um motor saudável de um motor com rolamentos ruins. Assim as 7 classes finais são *FanInner*, *FanOuter*, *FanBall*, *DriveInner*, *DriveOuter*, *DriveBall* e *Normal*. Classificando nestas 7 classes, os autores testaram 6 diferentes *pipelines* de processamento e implementação de modelos, comparando seus resultados tanto no *original* quanto no *proposed framework* na métrica de acurácia. A fórmula da acurácia (em inglês, *accuracy*) é:

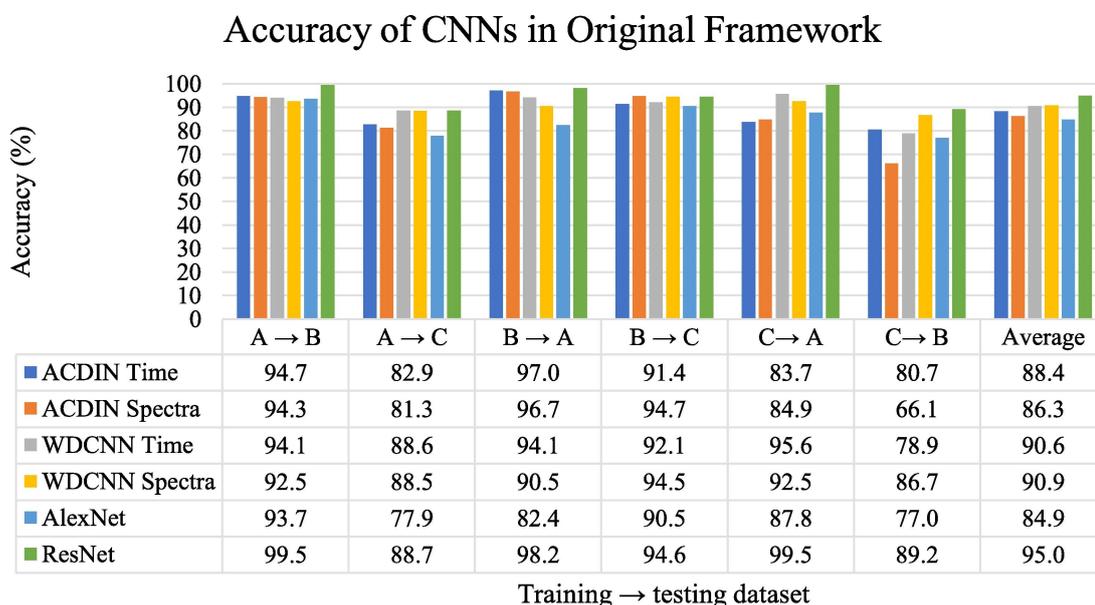
$$Acurácia = \frac{V. Positivos + V. Negativos}{V. Positivos + V. Negativos + F. Positivos + F. Negativos} \quad (1)$$

As figuras 7 e 8 mostram a acurácia alcançada por cada modelo no *original* e no *proposed framework* respectivamente. Aqui, os modelos são treinados em cada conjunto da estrutura e testados nos outros dois conjuntos da estrutura, medindo a acurácia de classificação alcançada nesse teste.

Dentre esses modelos, o modelo ACDIN (CHEN, Y. *et al.*, 2018) e o WDCNN (ZHANG *et al.*, 2017) tem como entrada um sinal unidimensional, expressado em função de um recorte do experimento no tempo ou da transformada espectral desse recorte, e os modelos AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, G. E., 2012) e ResNet (HE, K. *et al.*, 2015) são modelos de processamento de imagem, e recebem um espectrograma do sinal. A tabela 1 mostra a disposição dessas entradas.

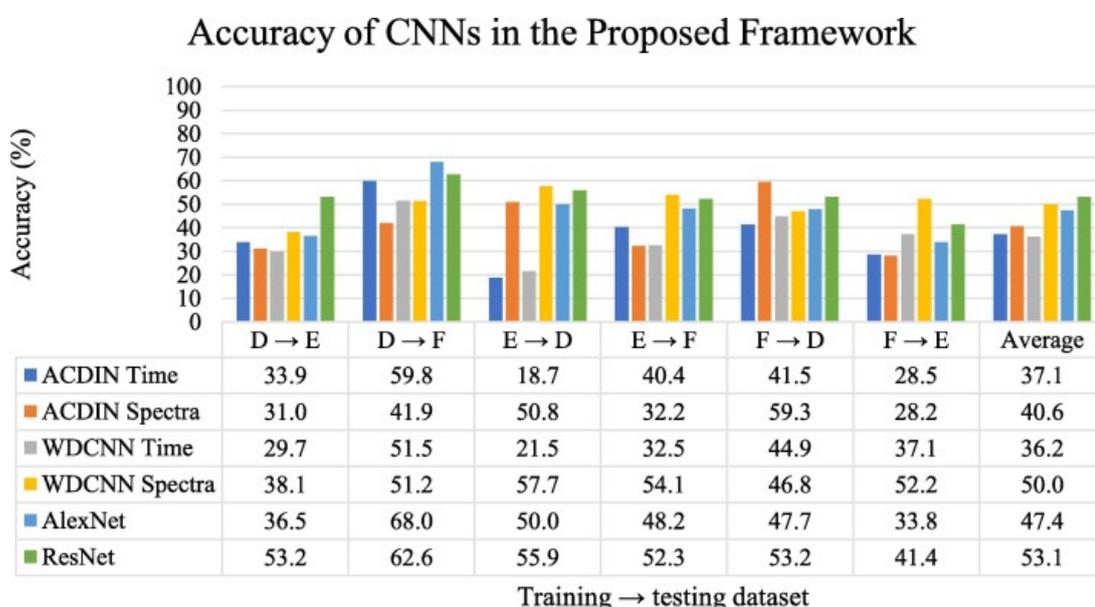
As amostras de sinal original para a composição dessas entradas são tiradas dos ensaios com sobreposição de 97% (exemplificando com um ensaio de 120000 amostras, podemos extrair 315 entradas (segmentos) de 11500 amostras com sobreposição de 97%).

Figura 7 – Resultados do artigo *Towards better benchmarking using the CWRU bearing fault dataset no original framework.*



Fonte: Hendriks, Dumond e Knox (2022).

Figura 8 – Resultados do artigo *Towards better benchmarking using the CWRU bearing fault dataset no proposed framework.*



Fonte: Hendriks, Dumond e Knox (2022).

Os resultados obtidos por Hendriks, Dumond e Knox (2022) corroboram com a tese proposta, com modelos que na estrutura original tinham a acurácia média entre 85-95% caindo para a faixa de 35-55% com o uso da estrutura proposta de divisão de

Tabela 1 – Entradas das redes do artigo *Towards better benchmarking*.

Rede	Tipo de Entrada	Comprimento do sinal Original	Acelerômetros Usados	Dimensões da Entrada na Rede
ACDIN	Tempo	5118	Drive End	[5118, 1]
ACDIN	Espectral	10236	Drive End	[5118, 1]
WDCNN	Tempo	2048	Drive End	[2048, 1]
WDCNN	Espectral	4096	Drive End	[2048, 1]
AlexNet	Espectrograma	11500	Drive End, Fan End	[227, 227, 2]
ResNet	Espectrograma	11500	Drive End, Fan End	[224, 224, 2]

Fonte: Hendriks, Dumond e Knox (2022).

dados.

2.3 REPRESENTAÇÕES VISUAIS DE SINAIS UNIDIMENSIONAIS

Para usarmos Redes Neurais Convolucionais como a ResNet para receber os sinais de acelerômetros e, com base neles, fazer uma inferência sobre o estado dos rolamentos do acionamento medido, precisamos converter uma amostra do sinal – Um vetor unidimensional de acelerações amostradas em um dado intervalo de tempo – em uma imagem, que traz informações em uma matriz bidimensional, onde cada elemento é tradicionalmente chamado pixel.

Ao elaborar estas representações, buscamos representar características que apresentam correlação com a condição do rolamento – como a disposição espectral em um dado instante de tempo do sinal, ou a magnitude da aceleração.

2.3.1 Espectrograma

Utilizado por Hendriks, Dumond e Knox (2022) para transformar os sinais do CWRU em imagens, o espectrograma é uma representação tempo-frequência de sinais no tempo, onde a intensidade do sinal em diferentes frequências é plotada em função do tempo. Essa representação é amplamente utilizada em diversas áreas, incluindo processamento de sinais, análise de áudio, processamento de imagem e reconhecimento de padrões.

O espectrograma é obtido por meio da aplicação da *Short-Time Fourier Transform* (Transformada de Fourier de Curto Termo) (STFT) em um sinal de entrada. A STFT divide o sinal em segmentos de tempo curtos (janelas) e calcula a Transformada de Fourier em cada segmento. Isso permite que as componentes de frequência do sinal sejam analisadas ao longo do tempo.

O resultado da STFT é uma matriz bidimensional em que o eixo horizontal representa o tempo e o eixo vertical representa a frequência. A intensidade do sinal em cada ponto da matriz é representada por uma cor ou um valor de intensidade, formando assim uma imagem que ilustra a distribuição de energia do sinal em diferentes

frequências ao longo do tempo.

A imagem final gerada pela STFT é controlada por uma série de parâmetros: O tamanho das janelas (dado em amostras do sinal original) é diretamente proporcional a resolução espectral da figura, e inversamente proporcional à resolução temporal. Janelas grandes resultam em uma maior resolução espectral, permitindo uma melhor distinção entre frequências próximas umas das outras. No entanto, a resolução temporal é reduzida, o que significa que eventos de curta duração podem não ser bem representados.

Por outro lado, janelas pequenas fornecem uma melhor resolução temporal, permitindo capturar eventos de curta duração com mais detalhes. No entanto, a resolução espectral é reduzida, o que pode levar a uma menor capacidade de distinguir entre frequências próximas.

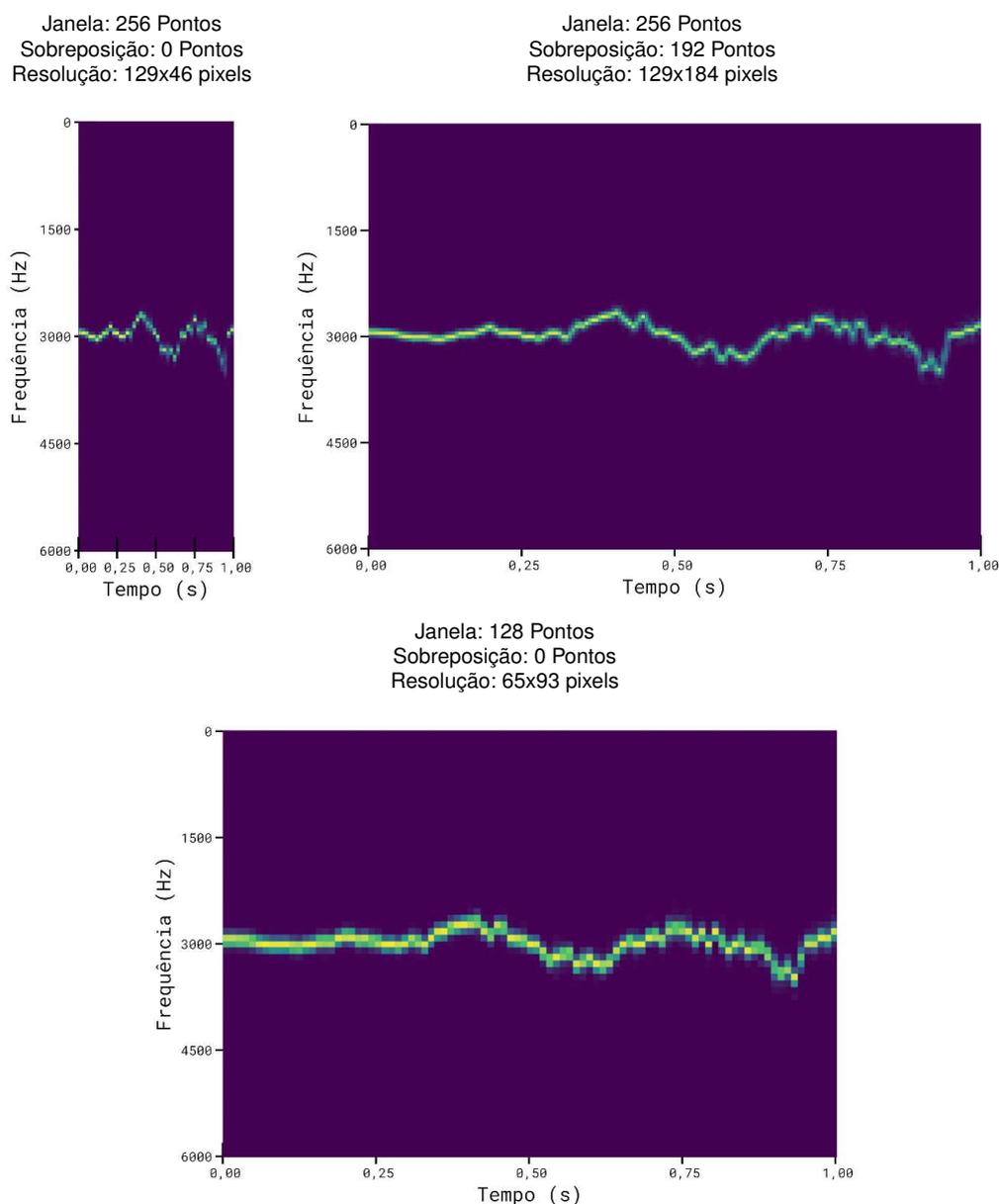
Além do tamanho das janelas, outro parâmetro importante é a sobreposição entre janelas adjacentes. Uma sobreposição maior entre os segmentos pode suavizar as variações temporais, enquanto uma sobreposição menor pode capturar mudanças rápidas no sinal, mas com menos suavidade. A sobreposição (também medida em amostras do sinal original) pode controlar o tamanho horizontal das imagens geradas, já que cada janela se torna uma coluna de pixels na imagem final.

A figura 9 mostra diferentes disposições de janela e sobreposição para um sinal sinusoidal com frequência variante ao longo do tempo: Na primeira representação, a janela de 256 pontos nos dá 129 bins de frequência, mas o tempo amostrado (1 segundo) é dividido em 46 segmentos, sem o uso de sobreposição. Na segunda representação, a janela tem 256 pontos como a primeira representação, mas com uma sobreposição de 192 pontos, sua dimensão horizontal fica mais proporcional, como o tempo dividido em 184 segmentos. Na terceira representação, temos uma janela de 128 pontos sem sobreposição, o que deixa o instante de transição entre as frequências claro e uma imagem resultante proporcional – mas perdemos a definição da frequência específica para qual o sinal transita, ou seja, os pixels aparecem mais altos por representar uma banda de frequências mais larga.

2.3.2 Escalograma

O artigo *Wavelets for fault diagnosis of rotary machines: A review with applications* (YAN; GAO; CHEN, X., 2014) explora o uso da transformada de wavelet para diagnóstico de falhas em máquinas rotativas. Na *Continuous wavelet transform* (Transformada de Wavelet Contínua) (CWT) temos uma função mãe denominada wavelet, um sinal oscilatório que concentra sua energia em um breve intervalo de tempo. A partir da wavelet mãe, geramos wavelets filhas em diferentes escalas temporais e convoluçionamos estas funções com o sinal que gostaríamos de analisar. Diferentes escalas da wavelet mãe realçam e destacam partes do sinal que tem sinergia com a wavelet na

Figura 9 – Diferentes disposições de janela e sobreposição no espectrograma de um sinal.



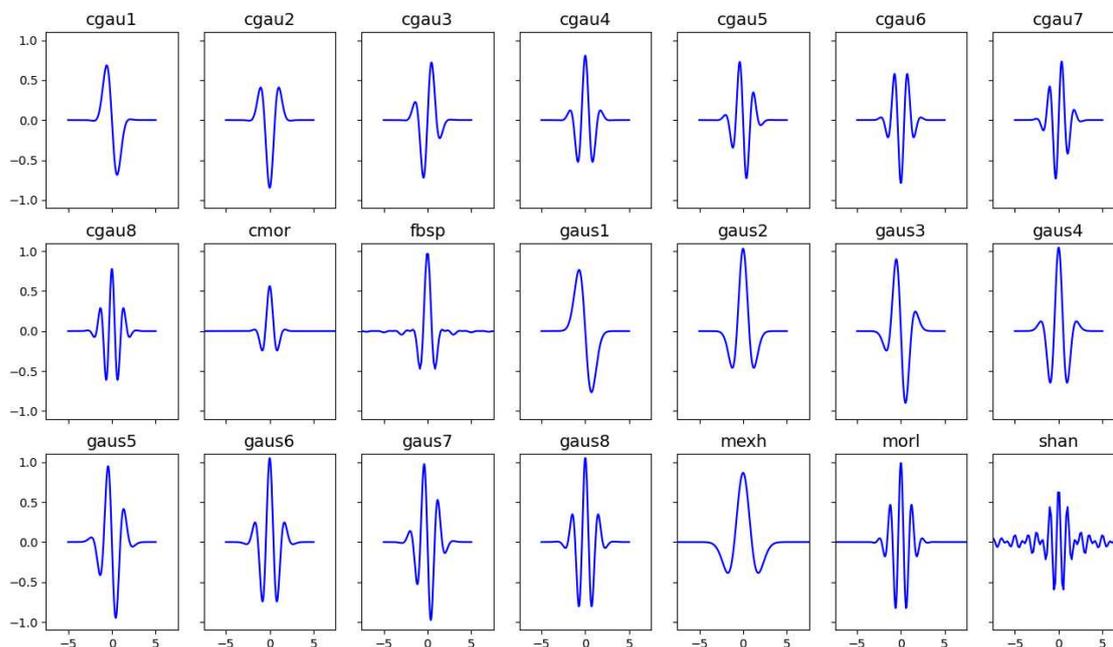
Fonte: Arquivo pessoal.

escala específica convolucionada. Podemos concatenar o sinal convolucionado com diferentes escalas verticalmente, obtendo uma representação tempo-escalas matricial do sinal original, denominada escalograma.

A figura 10 mostra diferentes formas de wavelets que podem ser usadas na CWT, que ao serem escaladas realçam diferentes features do sinal. Essas wavelets vem inclusas no pacote de python PyWavelets (LEE *et al.*, 2019), usado no decorrer do trabalho para gerar os escalogramas utilizados.

A imagem final tem a dimensão horizontal igual ao número de pontos do sinal

Figura 10 – Diferentes wavelets mães usadas na CWT.

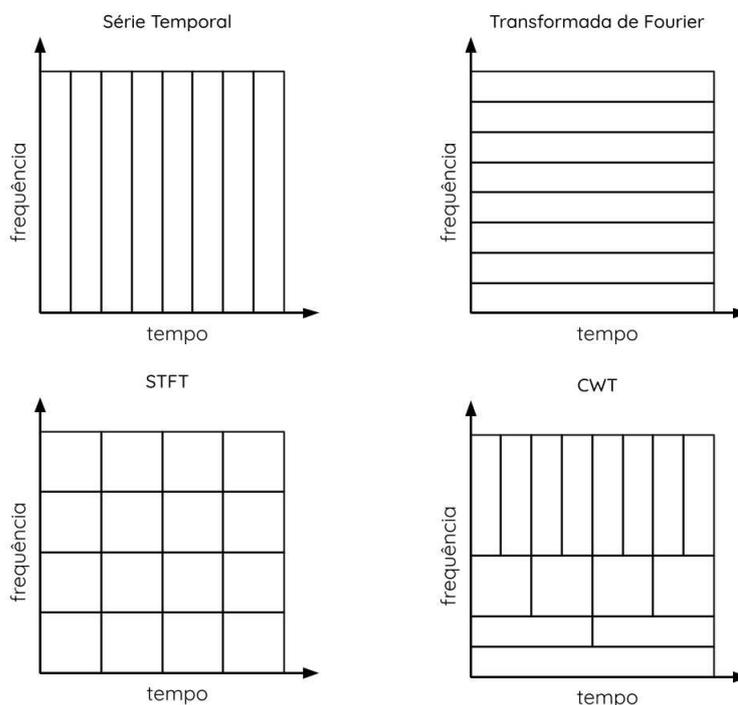


Fonte: Arquivo pessoal.

de entrada, e vertical igual ao número de escalas escolhidas para convolucionar. O número e dimensão das escalas é decidido de acordo com a aplicação. O pacote `pyWavelets` permite correlacionar escalas de wavelets filhas com frequências dadas em hertz, então podemos escolher qualquer espaçamento frequencial e quantidade de frequências para compor nosso escalograma.

Escalogramas apresentam a seguinte propriedade considerando sua capacidade de representação tempo-frequência: Wavelets em pequenas escalas representam as componentes de alta frequência do sinal, e possuem boa resolução temporal e baixa resolução espectral. Wavelets em grandes escalas apresentam a propriedade inversa: Representam as componentes de baixa frequência do sinal com baixa resolução temporal e alta resolução espectral. Pode-se intuir que é uma propriedade útil, pois há um interesse maior em analisar o instante de tempo que uma componente de alta frequência tem energia, e como componentes de baixa frequência já tem longa duração, suas posições no tempo podem ser menos relevantes que suas posições específicas no domínio da frequência. A imagem 11 mostra um diagrama de como a representação tempo-frequência da CWT se compara com outras transformadas, como a STFT e a transformada de Fourier pura.

Figura 11 – Resolução tempo-frequência de representações de sinais.



Fonte: Arquivo pessoal.

2.3.3 Cepstrograma

Com a disseminação do cepstro de um sinal para sistemas de reconhecimento de fala, Che e Lin (1994) elaboraram um programa para explorar a representação por cepstrogramas. Neste trabalho, implementamos um pacote autoral de python para realizar esta transformada.

O cepstro de potência é o resultado de computar o espectro de potência do logaritmo do espectro do sinal original. A Equação (2) mostra o cálculo do cepstro de potência.

$$C_p = \left| \mathcal{F} \left\{ \log \left(|\mathcal{F}\{f(t)\}|^2 \right) \right\} \right|^2. \quad (2)$$

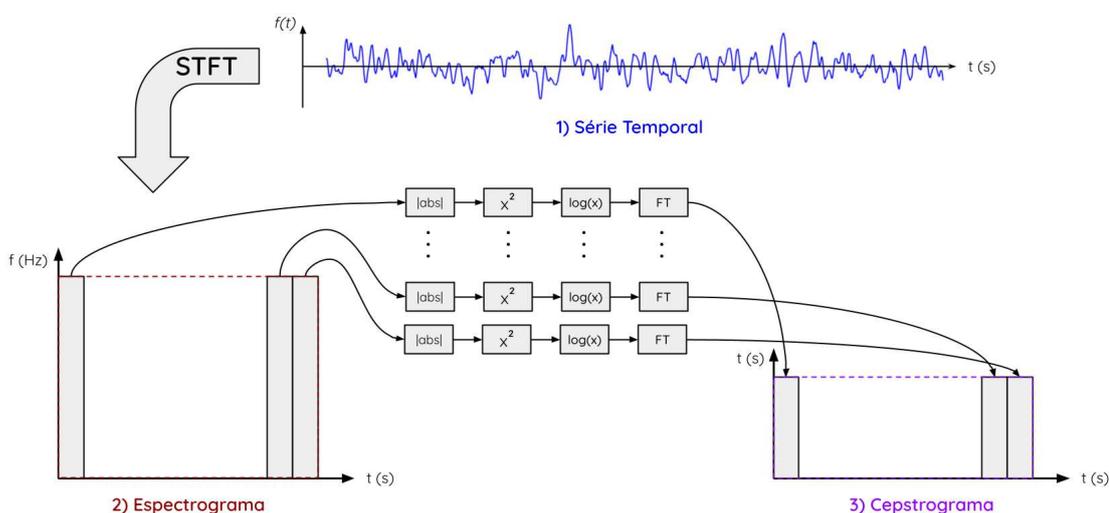
Por sua natureza como “espectro do espectro” a terminologia referente ao cepstro de potência faz alusão a sua natureza de inversão – se a frequência é o inverso do tempo, a frequência da frequência é o inverso da frequência – assim, cepstro é “spectro” com o começo invertido e conseqüentemente, as frequências cepstrais são denominadas quefrecias (medidas em tempo) e picos cepstrais associados a uma quefrecia fundamental são denominados rahmonicas, por exemplo.

O cepstro de potência é um método útil para detectar choques periódicos no sinal, como é o caso muitas vezes nas falhas de rolamentos (BOUDIAF *et al.*, 2016). Um

problema da técnica é que ela produz muito picos próximos ao ponto zero dificultando a interpretação do sinal.

Para obter um cepstrograma de um sinal, inicialmente obtemos um espectrograma convencional de magnitudes - em seguida operamos coluna a coluna (cada coluna representando a condição espectral do sinal em um dado instante de tempo) e considerando cada coluna uma entrada $|\mathcal{F}\{f(t)\}|$ completamos o cálculo do cepstro de potência, obtendo a distribuição de energia quefrenca deste dado instante de tempo. essas colunas processadas são concatenadas horizontalmente, formando o cepstrograma. A figura 12 mostra as transformações na série temporal necessárias para criar um cepstrograma.

Figura 12 – Fluxograma de criação de um cepstrograma.

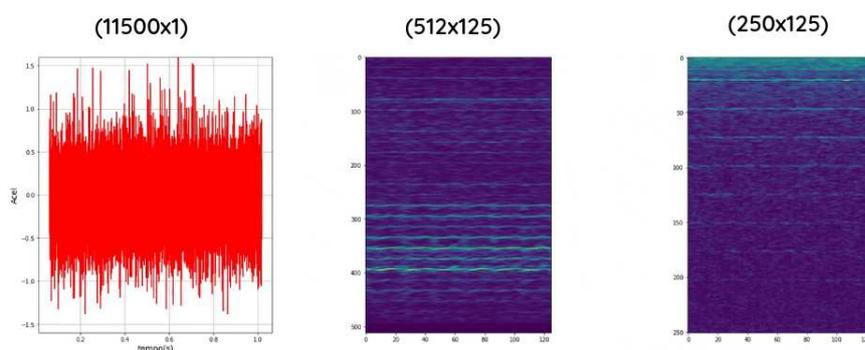


Fonte: Arquivo pessoal.

O cepstrograma resultante mantém a mesma resolução horizontal (número de pixels) do espectrograma que o originou, mas por conta da natureza simétrica da transformada de Fourier discreta, tem metade dos pixels em resolução vertical. A figura 13 mostra a geração de um cepstrograma a partir dos dados de vibração de um acelerômetro.

Um cepstrograma é principalmente parametrizado nos termos da janela e sobreposição de seu espectrograma inicial. Alguns procedimentos podem opcionalmente ser implementados para melhorar a legibilidade do cepstrograma resultante, como remover as primeiras linhas de quefrenca (remove os picos próximos do zero que dificultam a interpretação do sinal como especificado anteriormente) e, depois de tirar o log do espectro, podemos tirar a média ou até a tendência linear deste resultado, antes de fazer a transformada de Fourier final. Na figura 13 por exemplo, foram removidas as 7 primeiras linhas de quefrenca.

Figura 13 – Exemplo de geração de um cepstrograma.



Fonte: Arquivo pessoal.

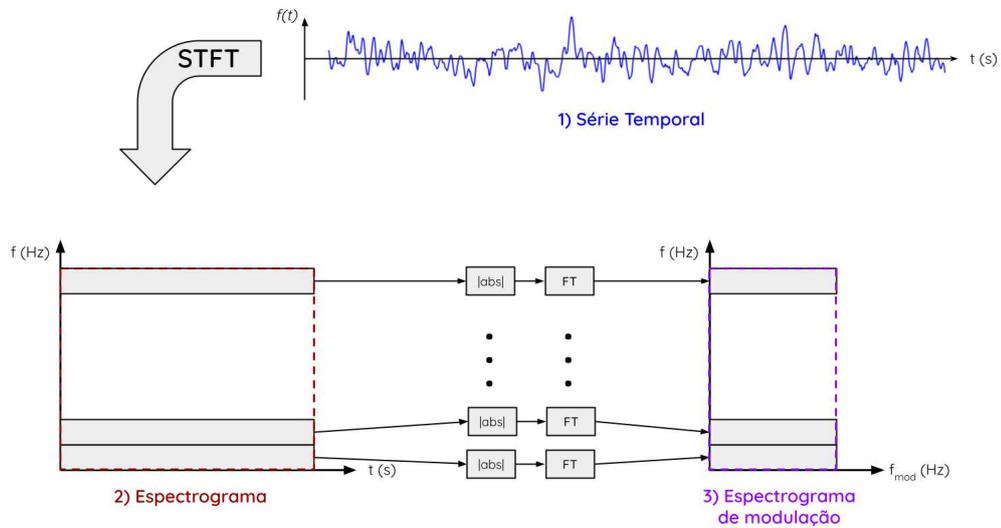
2.3.4 Espectrograma de modulação

O trabalho de Cassani *et al.* (2019) coloca que uma propriedade interessante de muitos sinais naturais não estacionários é que as mudanças ao longo do tempo em seu conteúdo espectral muitas vezes ocorrem de forma periódica única ou múltipla, resultando em cicloestacionaridade. Nesse sentido, um sinal não estacionário pode ser modelado como o resultado da interação de dois sinais independentes, um sinal de baixa frequência que altera (ou modula) as propriedades (como amplitude, fase e/ou frequência) de um sinal de alta frequência. Essa interação é o conhecido processo não linear de modulação. Falhas em rolamentos resultam em sinais modulados nas vibrações coletadas por acelerômetros, e esses sinais são claramente não estacionários.

Um espectrograma convencional descreve o comportamento temporal das componentes espectrais que compõem um sinal não estacionário, mas não caracteriza o comportamento periódico dessas componentes. Tomando a magnitude desses componentes espectrais complexos, podemos calcular o espectro deles. O resultado dessa operação, onde calculamos o espectro de cada linha de pixels (componente espectral) de um espectrograma e concatenamos as magnitudes desse espectro verticalmente em uma nova imagem é chamado espectrograma de modulação (CASSANI *et al.*, 2019). A figura 14 mostra as transformações na série temporal necessárias para criar um espectrograma de modulação.

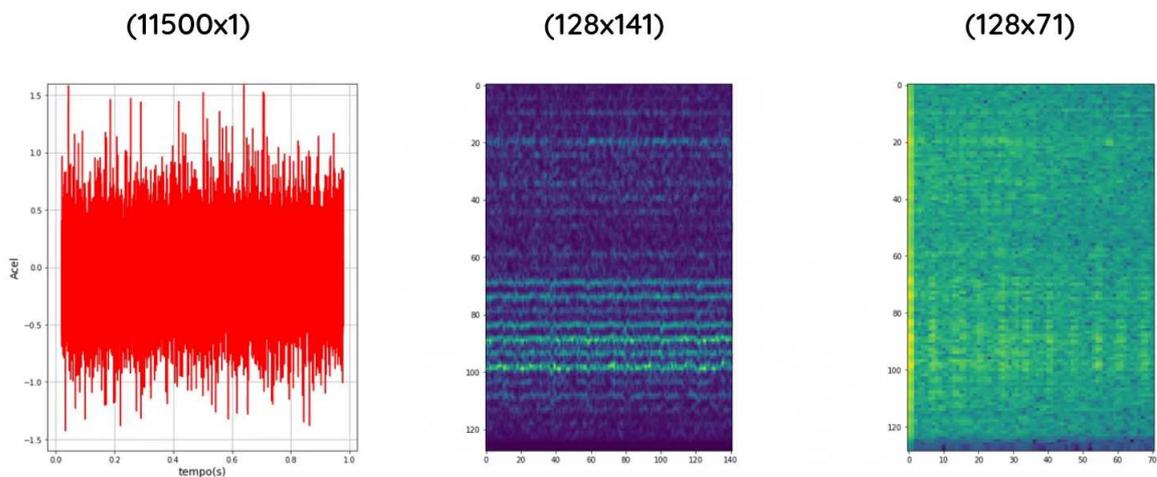
O espectrograma de modulação resultante mantém a mesma resolução vertical (número de pixels) do espectrograma que o originou, mas por conta da natureza simétrica da transformada de Fourier discreta, tem metade dos pixels em resolução horizontal. A figura 15 mostra a geração de um espectrograma de modulação a partir dos dados de vibração de um acelerômetro, com a magnitude final dos pixels escalada em decibéis.

Figura 14 – Fluxograma de criação de um espectrograma de modulação.



Fonte: Arquivo pessoal.

Figura 15 – Exemplo de geração de um espectrograma de modulação.



Fonte: Arquivo pessoal.

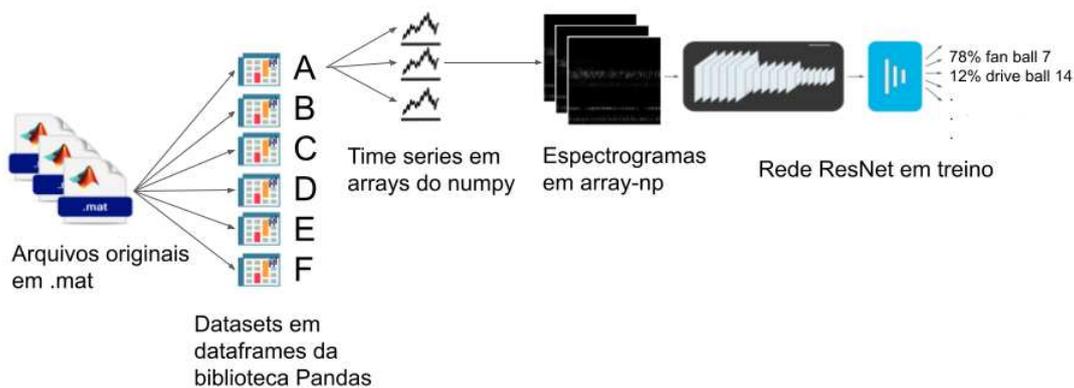
Como no caso de um cepstrograma, o espectrograma de modulação é principalmente parametrizado nos termos da janela e sobreposição de seu espectrograma inicial. O uso de uma sobreposição maior nos resulta em uma imagem com mais partições de frequências de modulação, logo uma imagem mais larga. Aumentar o tamanho da janela aumenta a resolução frequencial, mas diminui o total de segmentos no tempo e conseqüentemente, a resolução de frequências de modulação na imagem final, ou seja, reduz o número de pixels horizontalmente.

3 PESQUISA ANTECEDENTE

No segundo semestre de 2022, o bolsista pesquisador deu início a suas atividades no laboratório, iniciando uma frente de pesquisa orientada a encontrar e reproduzir os resultados de pesquisas do estado da arte na área de classificação de falhas. Em uma rodada de pesquisa bibliográfica foi levantado o artigo *Towards better benchmarking using the CWRU bearing fault dataset* (HENDRIKS; DUMOND; KNOX, 2022), que apontou falhas no uso do *CWRU Bearing Dataset* na pesquisa acadêmica atual e levantou uma possível direção de pesquisa, usando CNNs tradicionalmente utilizadas para a classificação de dados em imagens para classificar falhas em rolamentos, usando o espectrograma do sinal do acelerômetro como entrada.

Foi tomada a decisão de reproduzir os experimentos do artigo – especificamente com a ResNet, que apresentou os melhores resultados na estrutura proposta delineada na subseção 2.2.1, e em uma série de iterações de ciclos CRISP-DM (metodologia de desenvolvimento detalhada na na seção 4.1), foi elaborada uma pipeline de processamento de dados que extrai dos arquivos originais em formato .mat do *CWRU Bearing Dataset* os dados de acelerômetro, os divide de acordo com as estruturas mostradas na seção 2.2.1, os fragmenta em amostras e os transforma em espectrogramas, que podem ser alimentados a uma ResNet para treino ou inferência na rede. A figura 16 mostra os itens acima de forma gráfica.

Figura 16 – Funcionamento Conceitual da pipeline estabelecida



Fonte: Arquivo pessoal.

Foram estabelecidas algumas mudanças do processo de treino e teste usado por Hendriks *et al.*: Ao invés de concatenar os espectrogramas dos dois acelerômetros do ensaio ao longo dos canais de cores de uma imagem, foi usado apenas o espectrograma do acelerômetro posicionado no *Drive End* (mas ainda com o objetivo de classificar falhas no lado oposto), e cada configuração de experimento de treino e teste

é reproduzida cinco vezes, para obtermos uma média e um desvio padrão da acurácia de classificação dos conjuntos avaliados. O processo de treino é estocástico, e configurações iniciais dos pesos da rede e até a ordem dos exemplos de treino usados para a otimização (randomizados em cada experimento) podem afetar o desempenho do modelo.

O artigo estudado não menciona a arquitetura específica de ResNet utilizada - para a pesquisa, foi utilizada a arquitetura ResNet50.

O artigo também não menciona como ele dividiu os dados de normalidade entre os conjuntos da estrutura proposta. A abordagem utilizada na reprodução dos experimentos é que cada subconjunto D, E e F recebe um experimento de normalidade com uma carga diferente, com D recebendo o experimento de normalidade em 1 HP, E recebendo o de 2 HP e F o de 3 HP.

Treinando com os parâmetros de espectrograma dados em *Towards better benchmarking* (Janela de 104 pontos, com sobreposição de 54) e tentando otimizar hiperparâmetros de aprendizado como taxa de aprendizado e *weight decay*, ou escalonamento das entradas, obtivemos resultados na média significativamente aquém dos resultados de acurácia apresentado pelo artigo estudado. A tabela 2 compara o melhor modelo desenvolvido com os resultados do artigo.

Tabela 2 – Resultados do laboratório e do artigo estudado.

Rede	D→E	D→F	E→D	E→F	F→D	F→E	Média
ResNet (HENDRIKS; DUMOND; KNOX, 2022)	53,2%	62,6%	55,9%	52,3%	53,2%	41,4%	53,1 ±6,2%
Resnet50 Espectrogramas de 1 canal	38,25 ±4,64%	36,30 ±4,99%	47,00 ±6,28%	35,62 ±12,6%	51,51 ±11,1%	40,42 ±8,31%	41,52 ±6,38%

Fonte: Arquivo pessoal.

Se pode especular que no artigo estudado, os resultados de cada categoria são os melhores experimentos de um conjunto de experimentos repetidos, ou que os três ensaios de normalidade foram colocados em todos os conjuntos da estrutura proposta, perpetuando um problema de contaminação.

Teorizando que a CNN ResNet esteja aprendendo padrões muito específicos do sinal emitido pelos defeitos dos rolamentos em que foi treinada, que conseqüentemente a impedem de classificar acuradamente as falhas dos conjuntos de teste, foi proposta a ideia de treinar com dois conjuntos da estrutura proposta e testar a acurácia do modelo no conjunto restante. Isso preserva a noção de que devemos ter rolamentos distintos para conjuntos de treino e teste, mas dá uma variedade maior das formas que um defeito pode ter no espectrograma. Utilizando o mesmos parâmetros de configuração do treino da ResNet50 que resultaram nas nossas melhores acurácias e esse sistema de validação cruzada, observou-se uma significativa melhora, mostrada na tabela 3.

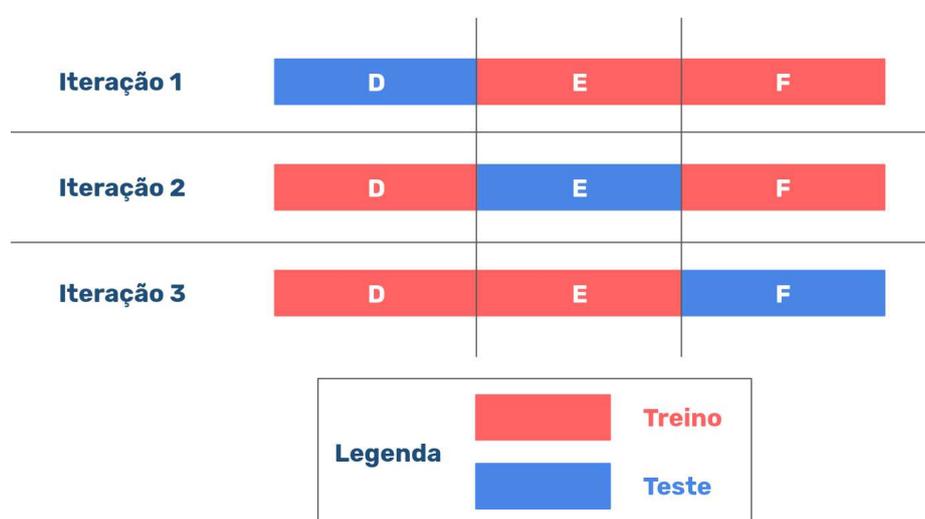
A figura 17 ilustra os novos conjuntos de treino e teste fazendo o uso da estrutura proposta.

Tabela 3 – Resultados com a nova proposta de uso dos conjuntos.

Rede	EF→D	DF→E	DE→F	Média
ResNet (HENDRIKS; DUMOND; KNOX, 2022)	-	-	-	53,1±6,2%
Resnet50 Espectrogramas de 1 canal	70,70±5,94%	43,77±5,13%	62,92±3,95%	59,13±13,8%

Fonte: Arquivo pessoal.

Figura 17 – Nova proposta de uso dos conjuntos



Fonte: Arquivo pessoal.

Com estes resultados, uma nova etapa de pesquisa se estruturou: Incorporando esta nova divisão de conjuntos de treino e teste como parte da metodologia para a avaliação da qualidade dos modelos (detalhada na subseção 4.2.1) e especificando uma metodologia interna aos conjuntos de treino para a otimização de hiperparâmetros (detalhada na subseção 4.2.2), o enfoque deixou de ser a reprodução dos resultados do artigo *Towards better benchmarking using the CWRU bearing fault dataset* (HENDRIKS; DUMOND; KNOX, 2022) e se orientou a exploração de diferentes transformações bidimensionais dos sinais de vibração (e a parametrização interna destas transformações) que podemos usar para alimentar a CNN ResNet, com o objetivo de criar *pipelines* de classificação mais acuradas, que se consolidou neste Trabalho de Conclusão de Curso.

Como uma mudança significativa para essa próxima etapa foi trocada a CNN utilizada: De ResNet50 para ResNet18, visando evitar *overfitting* no conjunto de treino

por excesso de memória, que pode carregar padrões excessivamente específicos para classificar falhas, bem como reduzir tempos de treino com a iteração por vários experimentos, já que a ResNet18 é uma versão reduzida da ResNet50, com menos camadas ocultas para otimizar.

4 METODOLOGIAS

Considerando nosso objetivo geral de elaborar modelos de *Machine Learning* (Aprendizado de Máquina) (ML) que classifiquem acuradamente falhas em rolamentos usando o *CWRU Bearing Dataset*, determinamos duas metodologias que orientam o processo de criação desses modelos: Uma metodologia de desenvolvimento que serviu como guia para organizar o trabalho dos pesquisadores, e outra metodologia – orientada a avaliação – determinou parâmetros e restrições na forma em que treinamos os modelos, visando uma otimização dos modelos sem falácias como contaminação entre dados de treino e teste.

4.1 METODOLOGIA DE DESENVOLVIMENTO

Para o processo de pesquisa e desenvolvimento dos modelos, foi empregada a metodologia *CRISP-DM* (WIRTH; HIPP, 2000), no qual o bolsista pesquisador realizou ciclos de desenvolvimento iterando pela preparação dos dados, modelagem e avaliação dos modelos, realizando reuniões semanais com o objetivo de informar o progresso das atividades e conhecimentos adquiridos ou identificar impedimentos e discutir possíveis soluções. As reuniões foram realizadas com o professor orientador e outros bolsistas do projeto, bem como representantes da empresa patrocinadora do estudo, que forneceram retornos e direcionamentos baseados no material apresentado.

4.2 METODOLOGIA DE OTIMIZAÇÃO

4.2.1 Divisão de Conjuntos de Treino e Teste

Em *Towards better benchmarking using the CWRU bearing fault dataset* (HENDRIKS; DUMOND; KNOX, 2022), os autores argumentam que, ao contrário da abordagem tradicionalmente utilizada, a divisão do conjunto de dados em treino e teste deve ser feita por rolamentos físicos, de forma que não apareçam no conjunto de teste sinais de um mesmo rolamento que apareça no conjunto de treinamento, uma vez que esta situação não seria realista. Assim, os autores fazem uso de uma avaliação dos modelos em que cada um dos conjuntos D, E e F é treinado individualmente e testado nos outros dois.

Visando proporcionar uma maior diversidade ao conjunto de treinamento, mas mantendo a abordagem de utilizar rolamentos distintos para treinamento e teste, é proposta uma modificação que combina dois subconjuntos para o treinamento e utiliza o subconjunto restante para o teste.

Essa abordagem implica em reduzir pela metade os testes entre diferentes domínios, igualando a nova metodologia de avaliação a uma validação cruzada *K-Fold* com $K = 3$. No entanto, é esperado que haja um ganho em relação à adaptação ao

tamanho da falha devido ao aumento da diversidade presente no conjunto de treinamento, o que deverá contribuir para que os modelos generalizem melhor, e portanto classifiquem melhor.

4.2.2 Conjunto de validação para otimização de Hiperparâmetros

Na subseção 4.2.1, estruturamos uma divisão dos ensaios do *CWRU Bearing Dataset* entre treino e teste, mas não podemos otimizar hiperparâmetros no conjunto de teste (no caso, testar que hiperparâmetros treinam a CNN no conjunto de treino que resultam na melhor acurácia no conjunto de teste). Como detalhado na subseção 2.1.3 essa prática pode levar a um *overfitting* dos hiperparâmetros aos dados de teste. O conjunto de teste deve ser reservado para avaliar o desempenho final do modelo, ou seja, para obter uma estimativa não enviesada de quão bem o modelo generaliza para dados não vistos.

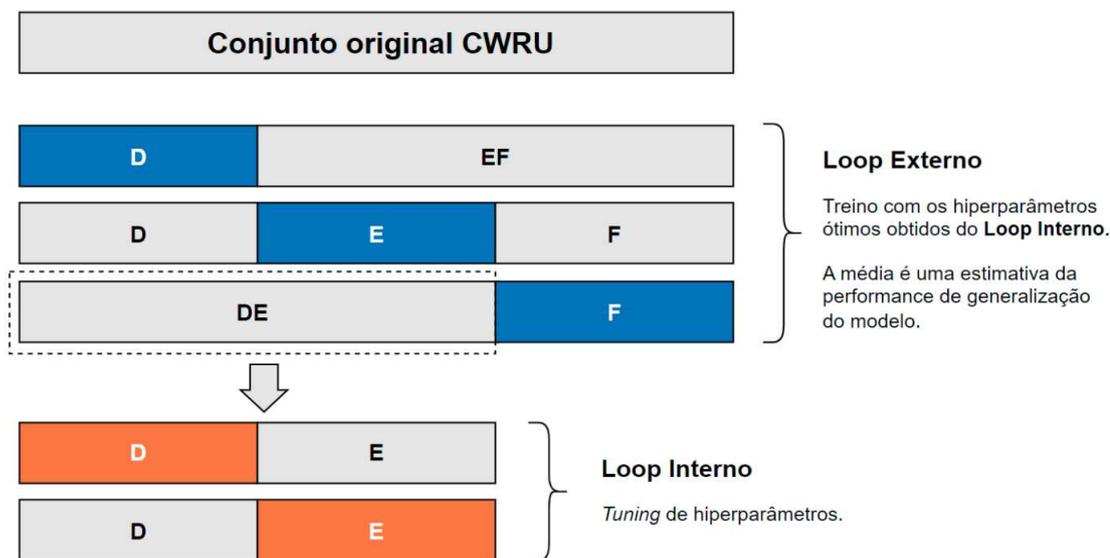
Para separarmos um conjunto de validação e realizar a otimização partimos dos nossos novos conjuntos de treino, oriundos da junção de dois conjuntos da estrutura proposta: EF, DF e DE. Dentro de cada um desses conjuntos, aplicamos uma validação cruzada *K-Fold* com $K = 2$. Essa estratégia corresponde ao laço interno de um processo de validação cruzada aninhada (RASCHKA, 2018). Desta forma, para encontrar os melhores hiperparâmetros de cada modelo, é analisada a curva de validação média do treinamento individual de cada um dos subconjuntos. Por exemplo, para a escolha dos hiperparâmetros do treinamento do subconjunto DE e teste no subconjunto F, é feito o treinamento em D, validando em E e logo após, o treinamento em E e validação em D. A partir das curvas de validação, é feita a média de ambas para avaliar quais curvas desempenharam melhor. A figura 18 ilustra a validação cruzada aninhada para nosso conjunto de dados.

Teoricamente podemos considerar cada elemento do loop externo da figura 18 um conjunto de dados para aprendizado de máquina completo e fazer a otimização de hiperparâmetros que retorna os melhores resultados dentro de cada loop interno, porém como se tem a motivação de estabelecer uma *pipeline* média de classificação que processe de maneira uniforme os dados de entrada, elaboramos o seguinte procedimento:

Em um primeiro momento, são fixados os hiperparâmetros de treino da rede - taxa de aprendizado e número de épocas de treino - e é testada a performance por acurácia dos diversos parâmetros da transformação bidimensional dos sinais de vibração, que determinam a forma da entrada na CNN ResNet, coletivamente denominados hiperparâmetros de entrada. São escolhidos parâmetros que performam bem no conjunto de curvas validação médias de cada elemento do loop externo.

Uma vez estabelecidos os hiperparâmetros da entrada, cada elemento do loop externo tem esses parâmetros fixados com o mesmo valor. Para cada loop interno

Figura 18 – Validação Cruzada Aninhada



Fonte: Arquivo pessoal.

variámos a taxa de aprendizagem (por 8 valores distintos, variando de 10^{-6} a 10^{-2} .) e escolhemos a com o melhor resultado observando a curva de validação média, e a partir dessa curva, determinamos a melhor época para parar o treino do modelo.

Finalmente, para cada elemento do loop externo o conjunto treino+validação é concatenado no conjunto de treino final, e a rede é treinada com ele usando os hiperparâmetros de entrada coletivos e os hiperparâmetros de treino individuais para aquele elemento. A rede treinada é usada para classificar as falhas do conjunto de teste associado, e uma média final é estabelecida com as acurácias de classificação de cada rede em seu conjunto de teste. A tabela 4 lista hiperparâmetros de entrada e de treino, definindo as duas categorias.

Tabela 4 – Hiperparâmetros de entrada e treino

Hiperparâmetros	
de Entrada	de Treino
<i>parametrizam características da imagem que o modelo irá classificar</i>	<i>parametrizam o algoritmo de aprendizagem do modelo</i>
Janela	Taxa de Aprendizado
Sobreposição	Épocas de treino
Normalização	
Redimensionamento	

Fonte: Arquivo pessoal.

É importante mencionar que para cada experimento em cada etapa, determinado por seu conjunto de treino, seu conjunto de teste (ou validação, dependendo da etapa) e seus hiperparâmetros, foram realizadas 5 iterações para obter a média de desempenho e o desvio padrão entre diferentes realizações, empregando uma inicialização aleatória dos pesos do modelo. Essa variação é um fator importante, podendo impactar na decisão entre hiperparâmetros que têm resultados similares na média, mas desvios padrão muito diferentes.

5 RESULTADOS

Neste capítulo, a primeira seção detalha a aplicação da metodologia de otimização exposta na seção 4.2 para 4 *pipelines* de classificação e os hiperparâmetros encontrados. Cada *pipeline* é composta de uma transformação do sinal de vibração em uma imagem por meio de uma representação visual (detalhada na seção 2.3) e uma CNN ResNet18 a ser treinada para diagnosticar falhas de acordo com a visualização do sinal provida. Cada *pipeline* é denominada de acordo com a transformação bidimensional usada.

A segunda seção detalha a avaliação destas *pipelines* no loop externo da validação cruzada aninhada utilizada, onde é testada a performance de generalização dos modelos com o uso dos conjuntos de teste designados, também é avaliada a performance dos modelos já otimizados na estrutura passada de avaliação, usada no artigo *Towards better benchmarking using the CWRU bearing fault dataset* (HENDRIKS; DUMOND; KNOX, 2022).

5.1 OTIMIZAÇÃO DE HIPERPARÂMETROS

5.1.1 Espectrograma

Para iniciar a otimização do espectrograma, é fixada a taxa de aprendizado em 10^{-6} e as redes são treinadas por 10 épocas, variando o tamanho da janela e o número de pontos de sobreposição, com o objetivo de controlar a largura e a altura final da imagem resultante. as larguras e alturas estipuladas são mostradas na tabela 5.

Tabela 5 – Grade de dimensões possíveis para o espectrograma.

HxW	32	64	128	256
32	32x32	64x32	128x32	256x32
64	32x64	64x64	128x64	256x64
128	32x128	64x128	128x128	256x128
256	32x256	64x256	128x256	256x256

Fonte: Arquivo pessoal.

Com uma acurácia de validação muito similar para representações com largura e altura maiores ou iguais a 64, foi escolhido um formato fora da grade, com a tomada de decisão sendo influenciada pelo uso de um *software* proprietário da empresa patrocinadora da pesquisa, que idealiza um espectrograma de acordo com a frequência de aquisição e velocidade do motor analisado. Não foram usados os valores de janela e sobreposição exatos do *software*, pois é esperado no futuro usar o sistema em outros conjuntos de dados, com outros motores e frequências de aquisição. Mas buscando um formato com fatores próximos de potências de 2 e relativamente “quadrado”, foi

escolhida uma janela de 192 pontos com sobreposição de 104 pontos, que resulta em um espectrograma de 96 pixels de altura por 104 pixels de largura, dada uma entrada (preservada do artigo de Hendriks, Dumond e Knox (2022), e determinada como a entrada padrão de todas as transformadas visuais) de 11500 pontos amostrais.

A normalização da imagem é legada dos resultados da pesquisa do ano passado, e usa *Standard Scaling*, cuja fórmula é:

$$z = \frac{x - \mu}{\sigma}. \quad (3)$$

Onde z é o novo valor de cada pixel da imagem, dado que x é o valor original do pixel, μ é o valor médio de todos os pixels do conjunto de dados de treino e σ é o desvio padrão de todos os pixels do conjunto de dados de treino.

Fixados os parâmetros do espectrograma, foi realizada uma batelada de experimentos para otimizar a taxa de aprendizado considerando as curvas de validação médias de cada elemento do loop externo dos conjuntos de dados (mostrado na imagem 18).

Determinando as melhores taxas de aprendizado e época de parada para cada conjunto do loop externo, foi organizada a tabela 6, que apresenta as taxas de aprendizado, épocas de parada e acurácia de validação média na época de parada obtida em cada conjunto do loop externo da validação cruzada aninhada.

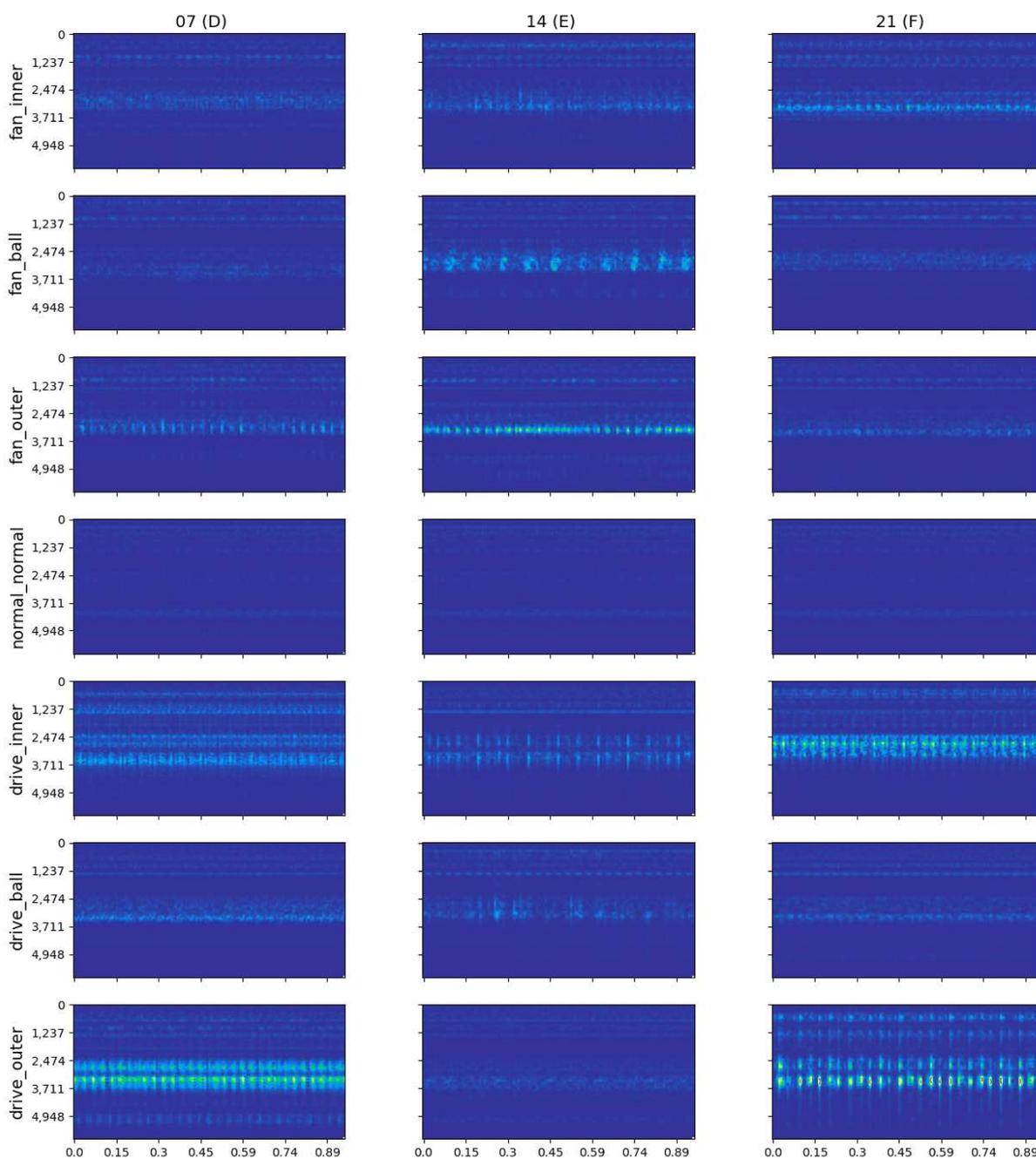
Tabela 6 – Resultados finais – otimização com espectrograma.

Loop Externo	TA	Época	Acurácia
EF	5,00E-04	5	46,18±6,37%
DF	1,00E-03	6	45,96±10,20%
DE	1,00E-03	4	35,44±5,33%
Média	-	-	42,52±5,01%

Fonte: Arquivo pessoal.

A figura 19 mostra os espectrogramas de todas as classes de defeitos (linhas) através de todos os tamanhos de defeitos, com a carga comum de 3 *HP*. Se colocando no lugar da ResNet18, pode-se inferir que uma boa transformada visual do sinal de vibração é diferente entre as linhas, permitindo a diferenciação da classe de falha, mas dentro de cada linha as colunas devem ser semelhantes, indicando uma boa possibilidade de generalização do aprendizado, ou seja que no momento de fazer uma inferência em um conjunto diferente, o que se aprendeu no conjunto original é útil.

Figura 19 – Espectrograma das classes de falha com carga de 3 HP



Fonte: Arquivo pessoal.

5.1.2 Escalograma

Para iniciar a otimização do escalograma, é fixada a taxa de aprendizado em 10^{-3} e as redes são treinadas por 10 épocas, variando a wavelet utilizada, e utilizando para cada wavelet a conversão de frequências para escala para selecionar as mesmas frequências em todas elas.

As frequências selecionadas são 11 frequências em decaimento exponencial da maior frequência mensurável de acordo com o teorema da amostragem de Nyquist (6000 Hz). Para esse experimento, a base do decaimento é 2 e as frequências convertidas em escala são: 6000 Hz, 3000 Hz, 1500 Hz e assim sucessivamente até 11,71875 Hz e 5,859375 Hz.

As wavelets testadas são as wavelets disponíveis no pacote de *python PyWavelets* (LEE *et al.*, 2019). O escalograma resultante tem 11500 pixels de largura e 11 pixels de altura, mas é linearmente redimensionado para 224x224 pixels, pois a imagem em seu tamanho original é muito grande e estipula-se que a ResNet é melhor em reconhecer *features* em imagens quadradas, pois seus *kernels* de convolução são quadrados.

A acurácia de validação de cada wavelet na última época é organizada em um ranking dentro de cada conjunto do loop externo, e um ranking geral é composto da soma da posição de cada wavelet nos rankings dos conjuntos. A tabela 7 mostra os rankings locais e geral dos conjuntos – quanto menor o valor do ranking geral, melhor foi o desempenho da wavelet dentro de cada conjunto.

Na batelada de experimentos seguinte, as 5 primeiras wavelets do ranking geral da tabela 7 foram testadas com escalogramas, variando as escalas testadas. Foram testadas as conversões de frequência para escala com o decaimento exponencial previamente estabelecido, bem como um espaçamento linear entre 6000 Hz e 10 Hz com 10, 100 e 200 amostras. Avaliando as curvas de validação, estabeleceu-se que o decaimento exponencial tem a melhor performance através dos conjuntos, e foi consolidada como forma de wavelet final a wavelet “fbsp”.

Em seguida, testou-se métodos de normalização, comparando a normalização *Standard Scaling* com *Max*. A fórmula da normalização Max é:

$$z = \frac{x}{\max(x)}. \quad (4)$$

Onde z é o novo valor de cada pixel da imagem, dado que x é o valor original do pixel, e $\max(x)$ é o valor máximo que um pixel pode atingir dentro de todas as imagens do conjunto de dados de treino. Nas curvas de validação, *Standard Scaling* se mostrou ligeiramente melhor.

Por fim, se descobriu que a fórmula que ditava o decaimento exponencial $\frac{6000}{2^n}$ poderia ser generalizada em uma fórmula para interpolar decaindo exponencialmente

Tabela 7 – Rankings das Wavelets

Wavelet	Ranking DE	Ranking DF	Ranking EF	Ranking Geral
fbsp	10	5	1	16
gaus8	12	3	2	17
gaus6	5	9	3	17
shan	9	6	4	19
gaus7	4	8	8	20
cgau4	6	12	6	24
cgau6	16	4	5	25
morl	1	7	20	28
gaus4	2	10	17	29
gaus5	3	17	10	30
gaus2	13	1	16	30
cgau1	19	2	15	36
cgau8	11	16	11	38
cgau3	7	13	18	38
cgau7	8	14	21	43
gaus1	20	15	9	44
gaus3	14	19	12	45
cgau5	15	11	19	45
cmor	18	21	7	46
cgau2	17	20	14	51
mexh	21	18	13	52

Fonte: Arquivo pessoal.

entre dois valores:

$$\frac{f_{maior}}{x^n} = f_{menor}, (n \in \mathbb{N}) \quad (5)$$

Onde f_{maior} é a maior frequência mensurável – 6000 Hz, e f_{menor} é a menor frequência de interesse – 10 Hz. Se escolhe um número de pontos de interpolação n e resolve-se a equação por x para encontrar um valor de x que faça a interpolação com decaimento exponencial entre 6000 Hz e 10 Hz em n pontos.

Testou-se então decaimentos exponenciais de $n=10, 100$ e 200 , bem como uma batelada de funções operando pixel a pixel dadas por $y = x^2$, $y = \log(x)$ e $y = \log(x^2)$, com uma taxa de aprendizado de 10^{-5} . As curvas de validação deram os melhores resultados gerais para o decaimento exponencial de 6000 a 10 Hz em 200 escalas e a função $y = \log(x^2)$, finalizando a etapa de otimização de hiperparâmetros para a entrada.

A parametrização final da transformada de wavelet utilizada foi com o uso da wavelet “fbsp”, 200 escalas equivalentes ao decaimento frequencial de 6000 a 10 Hz, normalização *Standard Scaling* e uma função *pixel wise* dada por $y = \log(x^2)$.

Repetindo o procedimento final da *pipeline* de espectrogramas, foi realizada uma batelada de experimentos para otimizar a taxa de aprendizado considerando as curvas de validação médias de cada elemento do loop externo dos conjuntos de dados (mostrado na imagem 18). Determinando as melhores taxas de aprendizado e época de parada para cada conjunto do loop externo, foi estruturada a tabela 8 com as taxas de aprendizado, épocas de parada e acurácia de validação média na dada época de parada de cada conjunto.

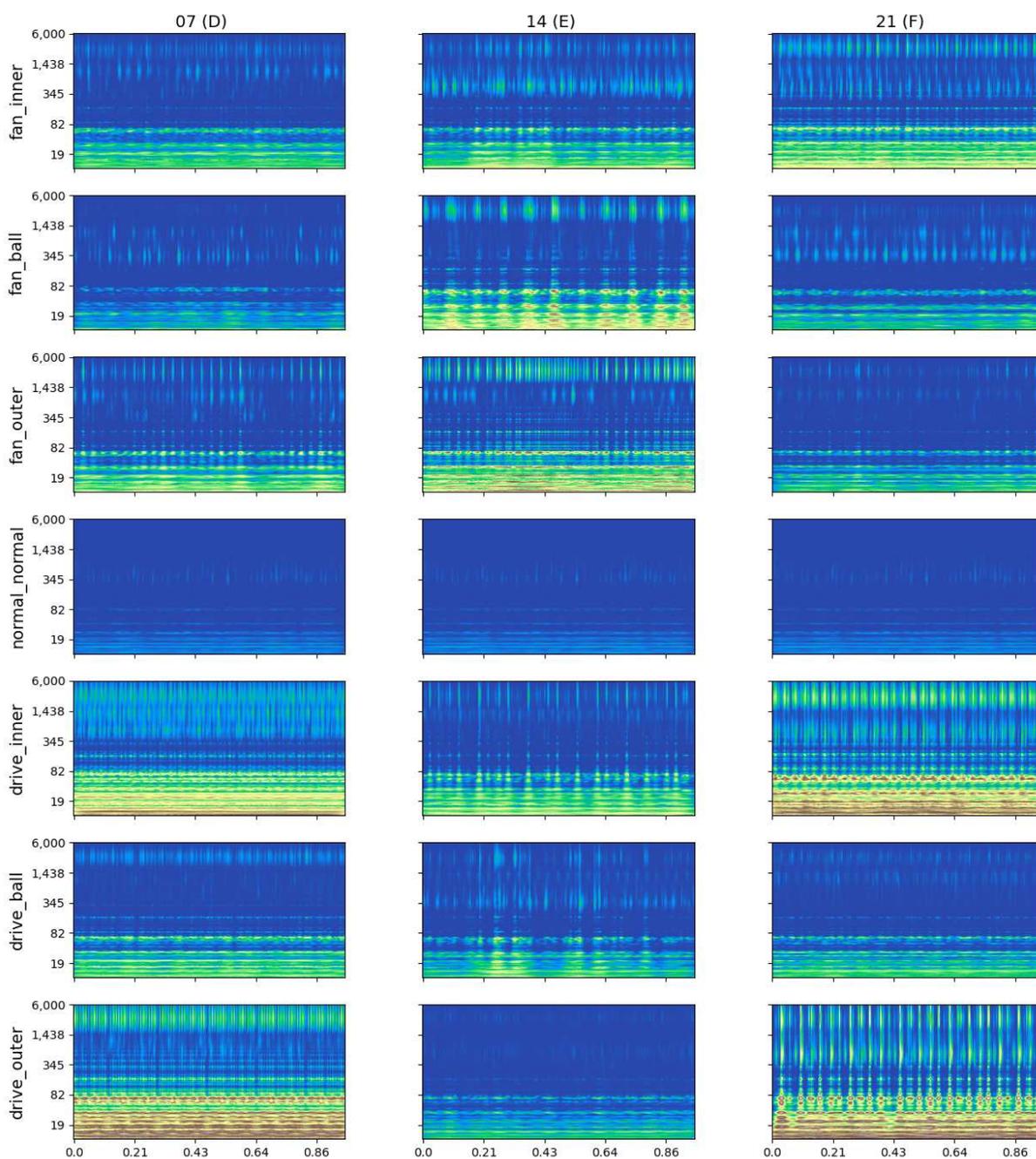
Tabela 8 – Resultados finais – otimização com escalograma.

Loop Externo	TA	Época	Acurácia
EF	1,00E-03	6	44,90±11,26%
DF	1,00E-05	8	54,29±8,95%
DE	1,00E-03	10	25,49±14,94%
Média	-	-	41,56±11,99%

Fonte: Arquivo pessoal.

A figura 20 mostra os escalogramas de todas as classes de defeitos (linhas) através de todos os tamanhos de defeitos, com a carga comum de 3 *HP*. Observa-se que, dentro de cada conjunto de dados, as imagens de cada classe de falhas são apropriadamente diferentes, mas também se vê significativas diferenças na representação da mesma classe de falha entre diferentes conjuntos de dados, especialmente o conjunto E comparado aos demais. Ao se observar o deslocamento da janela de entrada (de 11500 pontos) pelos experimentos do conjunto de dados, pode se notar que esta transformada tem *features* que variam intensamente ao longo do tempo - prejudicando a classificação.

Figura 20 – Escalograma das classes de falha com carga de 3 HP



Fonte: Arquivo pessoal.

5.1.3 Cepstrograma

A otimização do Cepstrograma foi mais breve, com mais parâmetros de entrada sendo testados simultaneamente, e avaliando-se o impacto sinérgico das varias opções de configuração: É fixada a taxa de aprendizado em 10^{-3} e as redes são treinadas por 10 épocas, variando:

- O uso de normalização *Standard* ou *Max Scaling*;
- A remoção da média do logaritmo do espectro (sim ou não);
- A remoção da tendência linear do logaritmo do espectro (sim ou não);
- A remoção dos primeiros n bins de quefrenca ($n = 0, 2$ ou 5);
- Janela e sobreposição do espectrograma intermediário, nas configurações:
 - janela de 192 pontos e sobreposição de 104 (resolução final: 48x129 pixels)
 - janela de 1024 pontos e sobreposição de 940 (resolução final: 256x125 pixels)

Observando essas 48 curvas de validação distintas para cada conjunto do loop externo, se observa que o critério de entrada com maior impacto foi a seleção da Janela e sobreposição: O uso da janela de 256 pontos levando as curvas de validação através de todos os conjuntos para uma faixa de acurácias médias entre 20 e 40%, e o uso da janela de 1024 pontos levou a faixa de acurácias médias de validação para ficar entre os 40 e 60%.

Os outros parâmetros apresentaram contribuições mais mistas, com melhores resultados na curva de validação de um conjunto e piores em outra, mas no geral, prevaleceu como vantajoso não remover a média do logaritmo do espectro e não remover a tendência linear. A remoção dos 5 bins de quefrenca (faixas horizontais de pixels do topo da imagem final) foi a opção mais vantajosa da categoria.

A parametrização final da transformada de cepstrograma fica dada como uma janela de 1024 pontos e sobreposição de 940 para o espectrograma intermediário, com uma normalização *Standard Scaling* dos pixels da imagem final, bem como a remoção dos primeiros 5 bins de quefrenca (primeiras 5 linhas de pixels da imagem final). A imagem final tem uma resolução de 251x125 pixels.

A otimização dos parâmetros da rede para cada elemento do loop externo foi realizada como detalhado nas seções anteriores. Determinando as melhores taxas de aprendizado e época de parada para cada conjunto do loop externo em uma nova batelada de experimentos, registramos esses valores na tabela 9, bem como a média da acurácia de validação na época de parada selecionada para cada conjunto.

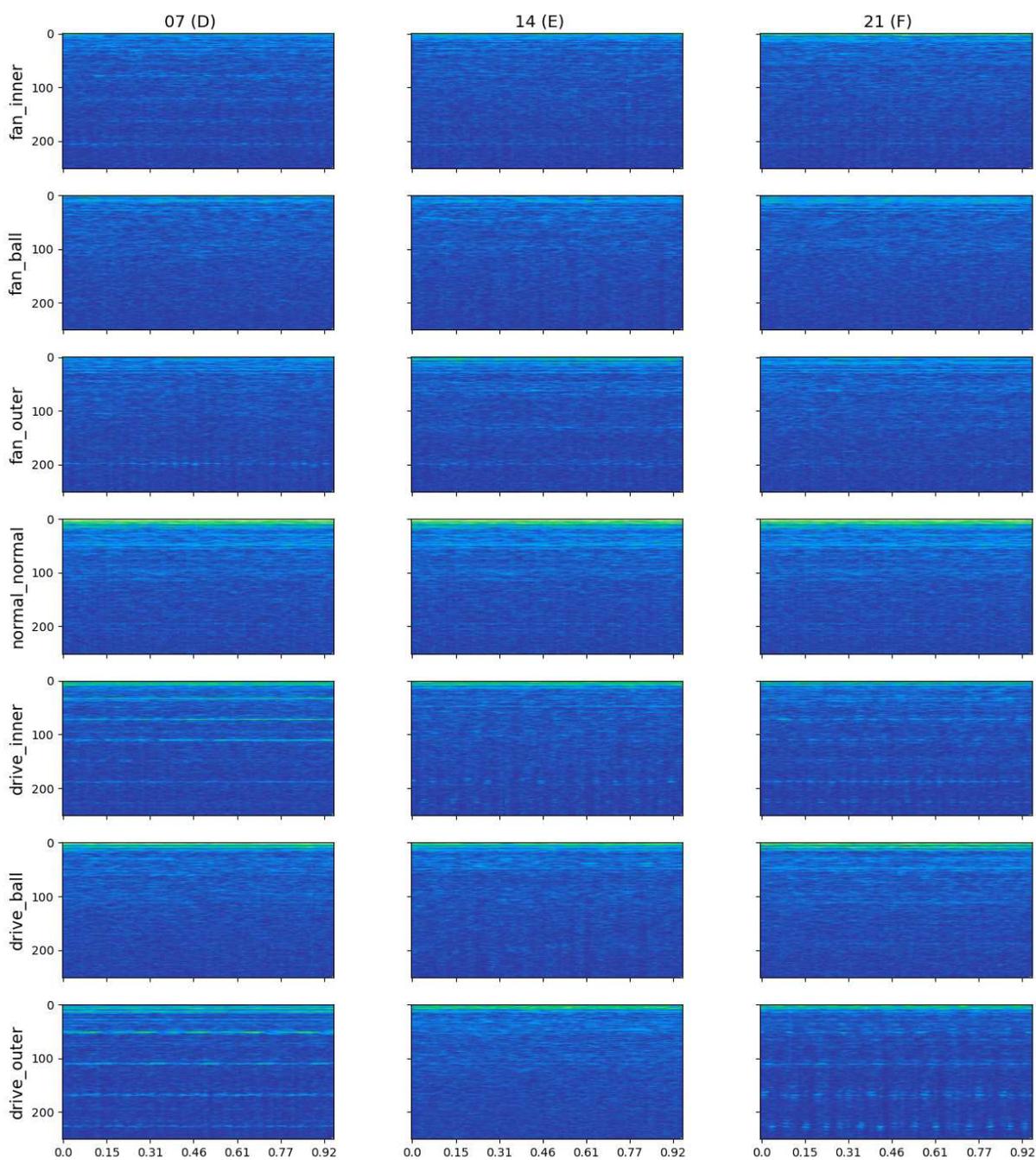
Tabela 9 – Resultados finais – otimização com cepstrograma.

Loop Externo	TA	Época	Acurácia
EF	5,00E-04	4	60,91±13,39%
DF	1,00E-03	3	61,42±5,49%
DE	1,00E-03	6	53,96±12,89%
Média	-	-	58,76±3,40%

Fonte: Arquivo pessoal.

A figura 21 mostra os cepstrogramas de todas as classes de defeitos (linhas) através de todos os tamanhos de defeitos, com a carga comum de 3 *HP* gerados usando os parâmetros de entrada escolhidos. O cepstrograma – apesar de seu uso na *pipeline* de processamento de dados ter resultado na maior acurácia média de validação dentre as transformadas expostas – é difícil de processar por “visão humana”, com *features* horizontais longas e invariantes no tempo, mas pode se observar diferenças entre classes de falha e se destaca dentre todas as imagens a classe de normalidade, que tem componentes de baixa quefrecia bem mais altos que as classes de falha, que tem picos de alta e média quefrecia, no caso as linhas claras que cortam a imagem.

Figura 21 – Cepstrograma das classes de falha com carga de 3 HP



Fonte: Arquivo pessoal.

5.1.4 Espectrograma de modulação

Sem variações internas do algoritmo como as presentes na geração de cepsogramas, para a geração de espectrogramas de modulação a otimização da transformada se orientou a testar composições de janela e sobreposição para o espectrograma intermediário, nas combinações detalhadas na tabela 10. Para cada uma dessas configurações, foi testado o uso de normalização *Standard* ou *Max Scaling*.

Tabela 10 – Parametrizações testadas para o espectrograma de modulação.

	Hiperparâmetros de entrada			
	Janela (pontos)	Passo (pontos)	Sobreposição (pontos)	Tamanho final (pixels)
Experimentos	179	90	89	90x64
	192	88	104	97x65
	256	20	236	129x282
	256	40	216	129x142
	256	60	196	129x95
	256	80	176	129x71
	128	20	108	65x285
	128	40	88	65x143
	128	60	68	65x96
	128	80	48	65x72

Fonte: Arquivo pessoal.

Avaliando as curvas de acurácia de validação médias através de todos os conjuntos do loop externo, a parametrização escolhida para o espectrograma de modulação visando um melhor desempenho de classificação geral foi: Janela do espectrograma intermediário de 256 pontos com sobreposição de 176 pontos e normalização *Standard Scaling* nas imagens finais.

A otimização dos parâmetros da rede para cada elemento do loop externo foi realizada como detalhado nas seções anteriores. Determinando as melhores taxas de aprendizado e época de parada para cada conjunto do loop externo em uma nova batelada de experimentos, é finalizada a otimização de hiperparâmetros da *pipeline*, registrando esses valores – bem como a acurácia de validação média na época de parada – na tabela 11.

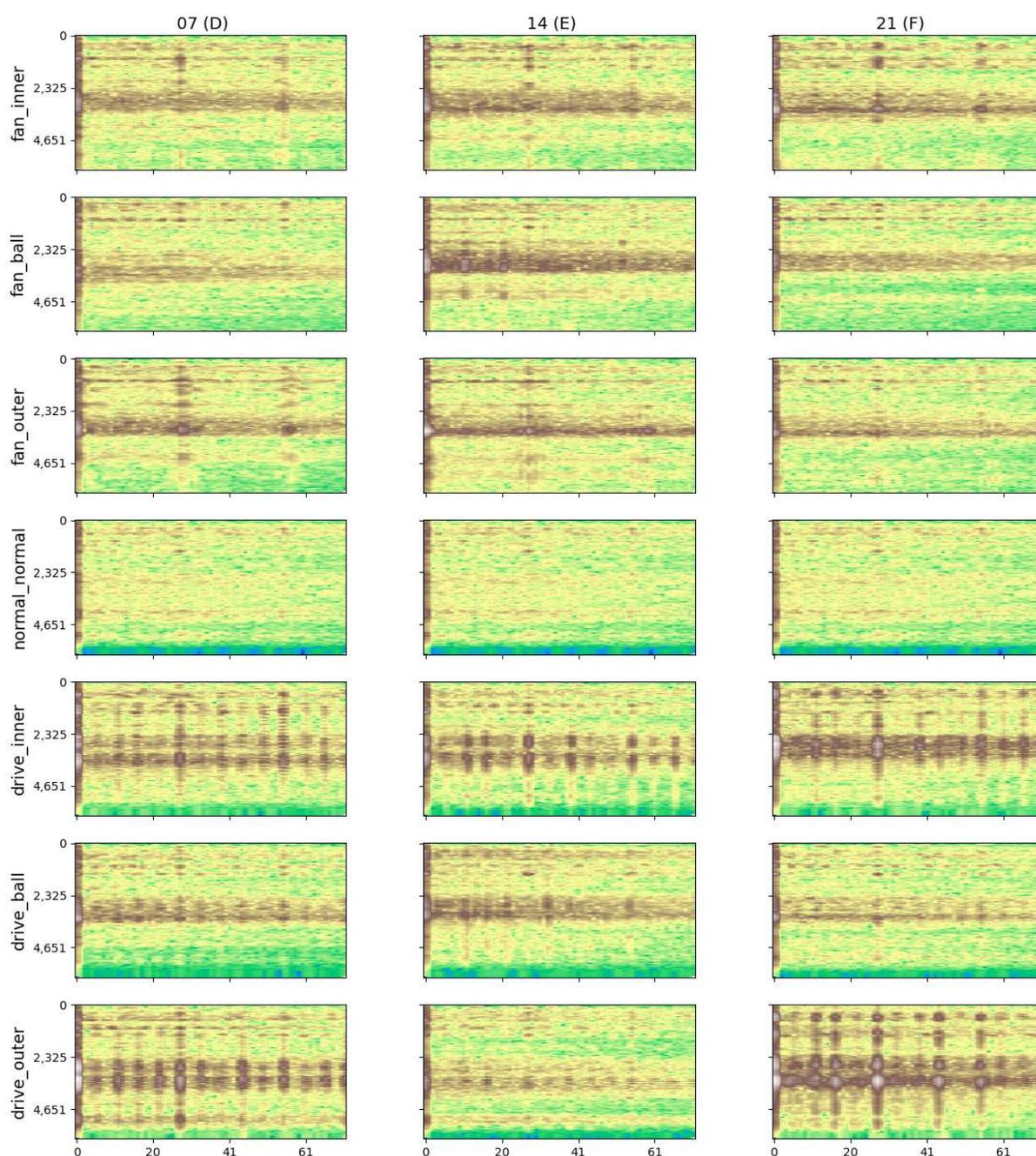
Tabela 11 – Resultados finais – otimização com espectrograma de modulação.

Loop Externo	TA	Época	Acurácia
EF	5,00E-05	4	71,08±11,32%
DF	1,00E-06	3	55,43±7,65%
DE	1,00E-06	3	54,88±5,44%
Média	-	-	60,46±7,51%

Fonte: Arquivo pessoal.

A figura 22 mostra os espectrogramas de modulação de todas as classes de defeitos (linhas) através de todos os tamanhos de defeitos, com a carga comum de 3 HP. O espectrograma de modulação representa de forma significativa diferenças entre cada classe de falha, enquanto simultaneamente destaca semelhanças intra-classe de falha dentro de cada conjunto. O conjunto de falhas do *fan end* é especialmente discernível com *features* diferenciáveis entre si, mas semelhantes em cada linha.

Figura 22 – Espectrograma de modulação das classes de falha com carga de 3 HP



Fonte: Arquivo pessoal.

5.2 AVALIAÇÃO DOS MODELOS

5.2.1 Avaliação no conjunto de teste

Com a otimização de hiperparâmetros para cada *pipeline* finalizada, fazemos um teste final para cada conjunto do loop externo da validação cruzada aninhada: Os modelos (com os hiperparâmetros definidos da seção anterior) são treinados com um conjunto de dados resultante da concatenação dos dados de treino e validação, e testados nos conjuntos de teste do loop externo. Coletando os resultados finais da acurácia de teste em cada *pipeline* otimizada, montamos a tabela 12:

Tabela 12 – Resultados finais com a Validação Cruzada Aninhada.

Entrada	Acurácia e Desvio Padrão (%)			
	EF → D	DF → E	DE → F	MÉDIA
Espectrograma	65,3±7,7	34,5±7,1	62,1±9,1	53,9±16,9
Escalograma	43,5±16,9	19,3±2,3	59,8±12,1	40,9±20,4
Cepstrograma	63,9±12,3	66,1±3,0	74,2±6,1	68,1±5,4
Espectrograma de modulação	56,4±1,7	60,3±5,1	79,1±4,4	65,2±12,1

Fonte: Arquivo pessoal.

Pode-se notar que a acurácia média final do espectrograma foi inferior a acurácia média obtida no experimento inicial com validação cruzada da pesquisa do ano passado, mostrada na tabela 3. Uma hipótese do motivo é a troca da CNN de ResNet50 para ResNet18, onde a ResNet18 é mais ágil de treinar, mas pode não ter uma capacidade de memória de *features* tão boa. Como se visa comparar o impacto das transformadas visuais na acurácia de classificação, e não as redes em si, o resultado final da tabela 12 é considerado como a real métrica da qualidade dos espectrogramas para diagnóstico de falhas no escopo deste trabalho.

Comparando a acurácia de teste média das *pipelines* – que fornece uma estimativa da performance de generalização do modelo – podemos observar que Cepstrogramas e Espectrogramas de modulação tem um grande ganho de acurácia de classificação comparado a abordagens mais tradicionais de visualização de sinais oscilatórios como o escalograma e o espectrograma. Discutimos um arrazoado do motivo:

Pode-se inferir que cada defeito de rolamento gera *features* espectrais únicas ao longo do tempo: O choque de uma bola saudável contra um defeito na trilha externa vai ter uma marca espectral no tempo (visível no espectrograma) diferente de outras falhas na trilha externa de outros rolamentos.

Ao avaliar a periodicidade dessa marca espectral no tempo, se emerge um

padrão que pode ser uniforme entre todos os rolamentos que tem essa falha. independente da profundidade, comprimento ou irregularidades internas dessa falha na trilha externa, a marca espectral dela ressoa toda vez que um elemento rolante passa por cima dela, e esse padrão periódico dentro de um espectrograma é capturado explicitamente pelo espectrograma de modulação como uma frequência modulada – que é mais consistente entre defeitos em rolamentos diferentes – ou indiretamente nas quefrecias do cepstrograma, que ressaltam periodicidades dentro do espectro. Considerando as várias *features* de cada classe de defeito que aparecem nas imagens, as CNNs treinadas adquirem heurísticas vinculando esses padrões de frequência das frequências ou períodos do instante de tempo a essas classes, aumentando sua acurácia em relação ao modelos que realizam inferências com base nos padrões da frequência no tempo.

5.2.2 Performance na estrutura anterior

Na estrutura de treino e testes proposta por Hendriks, Dumond e Knox (2022), se pode treinar e testar as pipelines elaboradas, mas a ausência de um conjunto de validação proíbe a otimização de hiperparâmetros para esta formulação de problema. Para fins de comparação das transformadas no paradigma anterior, tomamos a configuração de hiperparâmetros de entrada otimizados na seção 5.1 e para os hiperparâmetros da rede, é selecionada a taxa de aprendizado e a quantidade de épocas que obtiveram o melhor desempenho para cada tipo de entrada diferente. A tabela 13 mostra essa comparação.

Tabela 13 – Novas transformadas na estrutura anterior

Pipeline	TA	Épocas	Acurácia e Desvio Padrão (%)						Média
			D→E	D→F	E→D	E→F	F→D	F→E	
Espectrograma (Hendriks et al.)	5,00E-04	1	53,2	62,6	55,9	52,3	53,2	41,4	53,1
Espectrograma	5,00E-04	5	30,9 ±12,2	39,6 ±6,2	39,0 ±2,9	48,6 ±4,2	49,3 ±8,1	43,7 ±7,7	41,9 ±6,9
Escalograma	1,00E-04	6	16,4 ±1,0	44,4 ±5,6	20,0 ±11,9	43,0 ±4,2	58,6 ±11,9	38,7 ±16,0	36,8 ±16,0
Cepstrograma	1,00E-03	6	47,0 ±12,3	62,0 ± 6,6	61,0 ±10,0	63,4 ±18,0	62,3 ±6,0	52,1 ±10,0	57,9 ±6,8
Espectrograma de modulação	1,00E-06	4	56,0 ±3,7	56,7 ±10,3	53,8 ±7,0	70,6 ±2,9	54,2 ±4,8	49,6 ±4,2	56,8 ±7,2

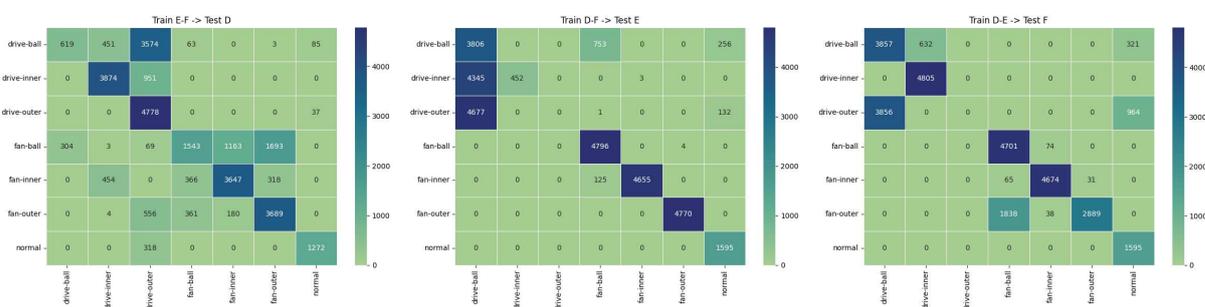
Fonte: Arquivo pessoal.

Observa-se que as *pipelines* com cepstrograma e espectrograma de validação tem uma vantagem sobre as transformadas tempo-frequência, generalizando melhor ao não ter *features* com grande variância temporal.

5.2.3 Matriz de Confusão

Para um olhar aprofundado em cima das dinâmicas de classificação, é selecionada a melhor *pipeline* encontrada – a *pipeline* com Cepstrograma – e em cima dela extraímos a matriz de confusão, rodando os 5 experimentos em cada conjunto do loop externo e somando as inferências em suas respectivas matrizes. Em uma matriz de confusão, contabilizamos cada inferência na linha de sua classe real e na coluna de sua classe predita. Um classificador perfeito tem todas as suas amostras de teste na diagonal principal da matriz. A figura 23 mostra as matrizes de confusão extraídas.

Figura 23 – Matrizes de Confusão da *pipeline* do cepstrograma



Fonte: Arquivo pessoal.

Se nota uma grande dificuldade do modelo em classificar as classes do *drive end*, especialmente referente a falhas no elemento rolante do *drive end* (classe *drive ball*). Em cada conjunto, observa-se que a maioria das amostras de *drive ball* é predominantemente categorizada como *drive inner* ou *drive ball* é categorizada corretamente, mas as outras duas classes do *drive end* são categorizadas erroneamente.

O modelo é mais coerente ao classificar falhas no *fan end*, e podemos notar que através dos conjuntos, classificações erradas pelo lado do motor que elas ocorrem são poucas, o que demonstra uma boa capacidade de discernir o lado em que as falhas ocorrem, usando apenas um acelerômetro.

6 CONCLUSÃO

Nesse trabalho, foi explorada a criação de *pipelines* de classificação de defeitos e falhas em rolamentos, compostas de uma transformação bidimensional que converte em uma imagem um sinal oscilatório de vibração (capturado de um acelerômetro na extremidade de um acionamento elétrico) e uma CNN que interpreta a imagem, vinculando o sinal bidimensional observado ao local no acionamento elétrico e parte defeituosa do rolamento que o gerou.

Em uma fundamentação teórica, detalhou-se as técnicas e propriedades do aprendizado de máquina em geral e das CNNs em particular, seguindo-se de uma introdução ao conjunto de dados utilizado para treinar as CNNs, o *CWRU Bearing Dataset* e seu uso em pesquisas de aprendizado da arte na literatura. Finalizou-se a fundamentação teórica com uma introdução às representações visuais, ou seja, as transformações bidimensionais que podem ser aplicadas a um sinal unidimensional para torná-lo de vetor em imagem.

Após apresentar o progresso da pesquisa realizado anteriormente pelo bolsista pesquisador e a equipe do laboratório no ano passado, se detalhou a metodologia usada para validar as *pipelines* com base na acurácia de predição, bem como a técnica de validação cruzada aninhada, usada para otimizar os parâmetros de entrada (inerentes às representações visuais) e parâmetros de rede (inerentes ao treino algorítmico da CNN). Finalmente, são mostrados como resultados os hiperparâmetros para cada *pipeline* e suas respectivas acurácias de validação e teste.

Os resultados obtidos indicam que as mudanças no pré-processamento dos dados de entrada usando as novas transformações visuais e a metodologia de otimização de hiperparâmetros implementada resultaram em um aumento próximo de 20% na acurácia média em relação à maior acurácia obtida por Hendriks, Dumond e Knox (2022) em seu artigo.

Melhorando a metodologia de avaliação ao adicionar repetições dos experimentos para maior diversidade estatística, além de propor uma forma de criar um conjunto de validação que não seja o próprio conjunto de testes para a nova proposta de divisão de dados trouxe uma maior robustez para a avaliação dos parâmetros das transformações e do treinamento da CNN ResNet.

No entanto é importante considerar que – como mencionado por Hendriks, Dumond e Knox (2022) e Smith e Robert B Randall (2015) em seus respectivos artigos – o *CWRU Bearing Dataset* tem algumas características de seus dados que geram dúvidas quanto à sua qualidade e se deveríamos continuar a usá-lo como *benchmark*. No trabalho de Smith e Robert B Randall (2015) por exemplo, os autores apontam que as falhas nos elementos rolantes não são possíveis de ser claramente identificadas por métodos clássicos de diagnóstico de falhas, o que pode ser um defeito do conjunto

de dados, mas também pode ser uma limitação dos métodos utilizados.

O conteúdo desenvolvido neste trabalho compõe parte de um artigo desenvolvido em conjunto pela equipe de pesquisa do laboratório, a ser publicado em breve. A pesquisa feita aqui contribui para o desenvolvimento de conjuntos de dados mais robustos e realistas para melhorar continuamente os sistemas de predição de falhas de máquinas, bem como estabelecer testes mais rigorosos de modelos de aprendizado de máquina antes dos mesmos serem colocados em produção, gerando valor para a empresa parceira.

REFERÊNCIAS

- BOUDIAF, Adel; MOUSSAOUI, Abdelkrim; DAHANE, Amine; ATOUI, Issam. A comparative study of various methods of bearing faults diagnosis using the case Western Reserve University data. **Journal of Failure Analysis and Prevention**, Springer, v. 16, n. 2, p. 271–284, 2016.
- CASE Western Reserve University Bearing Data Center Website. [S.l.: s.n.], ago. 2021. Disponível em: <https://engineering.case.edu/bearingdatacenter>.
- CASSANI, Raymundo; ALBUQUERQUE, Isabela; MONTEIRO, João; FALK, Tiago H. AMA: An Open-source Amplitude Modulation Analysis Toolkit for Signal Processing Applications. *In*: 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP). [S.l.: s.n.], 2019. P. 1–4. DOI: 10.1109/GlobalSIP45357.2019.8969210.
- CHE, Chiwei; LIN, Qiguang. Cepstrogram and its application to speech analysis. **The Journal of the Acoustical Society of America**, Acoustical Society of America, v. 96, n. 5, p. 3351–3351, 1994.
- CHEN, Yuanhang; PENG, Gaoliang; XIE, Chaohao; ZHANG, Wei; LI, Chuanhao; LIU, Shaohui. ACDIN: Bridging the gap between artificial and real bearing damages for bearing fault diagnosis. **Neurocomputing**, v. 294, p. 61–71, 2018. ISSN 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2018.03.014>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S092523121830300X>.
- HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. **Deep Residual Learning for Image Recognition**. [S.l.]: arXiv, 2015. DOI: 10.48550/ARXIV.1512.03385. Disponível em: <https://arxiv.org/abs/1512.03385>.
- HE, Qiuchen; LI, Shaobo; BAI, Qiang; ZHANG, Ansi; JING, Yang; SHEN, Mingming. A Siamese Vision Transformer for Bearings Fault Diagnosis. **Micromachines**, v. 13, p. 1656, set. 2022. DOI: 10.3390/mi13101656.
- HENDRIKS, Jacob; DUMOND, Patrick; KNOX, D.A. Towards better benchmarking using the CWRU bearing fault dataset. **Mechanical Systems and Signal Processing**, v. 169, p. 108732, 2022. ISSN 0888-3270. DOI: <https://doi.org/10.1016/j.ymsp.2021.108732>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0888327021010499>.

HOANG, Duy-Tang; KANG, Hee-Jun. Rolling element bearing fault diagnosis using convolutional neural network and vibration image. **Cognitive Systems Research**, v. 53, p. 42–50, 2019. Advanced Intelligent Computing. ISSN 1389-0417. DOI: <https://doi.org/10.1016/j.cogsys.2018.03.002>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1389041717303261>.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. ImageNet Classification with Deep Convolutional Neural Networks. In: PEREIRA, F.; BURGESS, C.J.; BOTTOU, L.; WEINBERGER, K.Q. (Ed.). **Advances in Neural Information Processing Systems**. [S.l.]: Curran Associates, Inc., 2012. Disponível em: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.

LECUN, Yann; BENGIO, Y.; HINTON, Geoffrey. Deep Learning. **Nature**, v. 521, p. 436–44, mai. 2015. DOI: 10.1038/nature14539.

LEE, Gregory R.; GOMMERS, Ralf; WASELEWSKI, Filip; WOHLFAHRT, Kai; O'LEARY, Aaron. PyWavelets: A Python package for wavelet analysis. **Journal of Open Source Software**, The Open Journal, v. 4, n. 36, p. 1237, 2019. DOI: 10.21105/joss.01237. Disponível em: <https://doi.org/10.21105/joss.01237>.

LEI, Yaguo; YANG, Bin; JIANG, Xinwei; JIA, Feng; LI, Naipeng; NANDI, Asoke K. Applications of machine learning to machine fault diagnosis: A review and roadmap. **Mechanical Systems and Signal Processing**, v. 138, p. 106587, 2020. ISSN 0888-3270. DOI: <https://doi.org/10.1016/j.ymsp.2019.106587>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0888327019308088>.

RANDALL, R.B. **Vibration-based Condition Monitoring: Industrial, Aerospace and Automotive Applications**. [S.l.]: Wiley, 2011. ISBN 9780470747858. Disponível em: <https://books.google.com.br/books?id=2v4RDQEACAAJ>.

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

SMITH, Wade A; RANDALL, Robert 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.

WIRTH, Rüdiger; HIPPE, Jochen. Crisp-dm: towards a standard process model for data mining. *In*.

YAN, Ruqiang; GAO, Robert X; CHEN, Xuefeng. Wavelets for fault diagnosis of rotary machines: A review with applications. **Signal processing**, Elsevier, v. 96, p. 1–15, 2014.

ZHANG, Wei; PENG, Gaoliang; LI, Chuanhao; CHEN, Yuanhang; ZHANG, Zhujun. A New Deep Learning Model for Fault Diagnosis with Good Anti-Noise and Domain Adaptation Ability on Raw Vibration Signals. **Sensors**, v. 17, n. 2, 2017. ISSN 1424-8220. DOI: 10.3390/s17020425. Disponível em: <https://www.mdpi.com/1424-8220/17/2/425>.

ZHU, Zhiyu; PENG, Gaoliang; CHEN, Yuanhang; GAO, Huijun. A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis. **Neurocomputing**, v. 323, p. 62–75, 2019. ISSN 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2018.09.050>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231218311238>.