



CENTRO UNIVERSITÁRIO SENAI CIMATEC
PROGRAMA DE PÓS GRADUAÇÃO EM GESTÃO E TECNOLOGIA
INDUSTRIAL
MESTRADO EM GESTÃO E DESENVOLVIMENTO INDUSTRIAL

ROBERTO MACEDO DE SOUZA

CLASSIFICAÇÃO DE FALHAS EM ATIVOS ROTATIVOS
UTILIZANDO REDE NEURAL CONVOLUCIONAL

Salvador

2020

ROBERTO MACEDO DE SOUZA

**CLASSIFICAÇÃO DE FALHAS EM ATIVOS ROTATIVOS
UTILIZANDO REDE NEURAL CONVOLUCIONAL**

Dissertação apresentada ao Programa de Mestrado em Gestão e Tecnologia Industrial no Centro Universitário SENAI CIMATEC, como requisito para obtenção do grau de Mestre em Gestão e Tecnologia Industrial.

Orientador: Prof. Dr. Herman Augusto Lepikson

Co-orientador:
Prof. Dr. Erick Giovani Sperandio Nascimento

Salvador
2020

Ficha catalográfica elaborada pela Biblioteca do Centro Universitário SENAI CIMATEC

S719c Souza, Roberto Macedo de

Classificação de falhas em ativos rotativos utilizando rede neural convolucional / Roberto Macedo de Souza. – Salvador, 2020.

84 f. : il. color.

Orientador: Prof. Dr. Herman Augusto Lepikson.

Coorientador: Prof. Dr. Erick Giovani Sperandio Nascimento.

Dissertação (Mestrado em Gestão e Tecnologia Industrial) – Programa de Pós-Graduação, Centro Universitário SENAI CIMATEC, Salvador, 2020.

Inclui referências.

1. Análise de vibração mecânica. 2. Redes neurais convolucionais. 3. Aprendizagem profunda – Gestão da manutenção. 4. Data Driven. I. Centro Universitário SENAI CIMATEC. II. Lepikson, Herman Augusto. III. Nascimento, Erick Giovani Sperandio. IV. Título.

CDD: 621.8

CENTRO UNIVERSITÁRIO SENAI CIMATEC

Mestrado Profissional em Gestão e Tecnologia Industrial

A Banca Examinadora, constituída pelos professores abaixo listados, aprova a Defesa de Mestrado, intitulada "Classificação de falhas em ativos rotativos utilizando rede neural convolucional" apresentada no dia 16 de março de 2020, como parte dos requisitos necessários para a obtenção do Título de Mestre em Gestão e Tecnologia Industrial.

Orientador:



Prof. Dr. Herman Augusto Lepikson
SENAI CIMATEC

Coorientador:



Prof. Dr. Erick Giovani Sperandio Nascimento
SENAI CIMATEC

Membro Interno:



Prof. Me. Ubatan Almeida Miranda
FIEB

Membro Externo:



Prof. Dr. Leizer Schmitman
UFBA

AGRADECIMENTOS

São tantos agradecimentos, que peço desculpas, caso não os cite.

Primeiramente, agradeço a Deus por todos os momentos nos quais Ele me ajudou a enfrentar os desafios e a persistir na conclusão de mais essa etapa árdua. Agradeço também por todas as pessoas que Ele colocou no meu caminho e que me ajudaram imensamente. Ao Senhor, gratidão eterna.

Agradeço ao professor Dr. Herman Lepikson pela orientação, ensinamentos, confiança, incentivo, direcionamento e disposição em sempre contribuir para o bom andamento da pesquisa e desenvolvimento da solução.

Agradeço aos professores Dr. Erick Sperandio e Me. Ubatan Miranda que foram essenciais para o andamento e conclusão da pesquisa. Sem o apoio incondicional deles nem e as incontáveis reuniões para delineamento e direcionamento da pesquisa, este trabalho não teria sido finalizado.

Ao amigo e parceiro de muitas horas de discussões técnicas, Wenisten Dantas, agradeço pelo grande e fundamental apoio na realização dos experimentos.

A todos da turma GETEC 10 agradeço pela amizade, encorajamento e disponibilização de informações em momentos delicados.

Agradeço aos diversos amigos da vida profissional que tanto me ensinaram, disponibilizaram diversos materiais didáticos, me aconselharam no direcionamento da pesquisa, revisaram os experimentos, e correlacionaram as respostas da solução com a visão prática.

Agradeço à minha família pelo apoio e compreensão nos diversos momentos em que fiquei ausente.

Agradeço à família composta por Capiá, D^a Maria, Maria Luiza e Jorge pelo acolhimento, carinho e boas conversas que me proporcionaram, quando residi em Recife-PE.

Agradeço à instituição SENAI CIMATEC pela disponibilização de sua infraestrutura, fundamental à realização dos experimentos.

RESUMO

Máquinas rotativas são amplamente empregadas em processos industriais, cujo uso contínuo provoca desgastes em diversos dos seus componentes, demandando a parada do ativo, para substituí-los. Em função da importância dos ativos para a continuidade operacional de uma indústria, é valiosa sua monitoração constante com o objetivo de predição de falha, realizar uma manutenção programada e minimizar a possibilidade de falha em condições não planejadas. A estratégia de manutenção preditiva, a qual realiza o monitoramento constante dos ativos através das principais variáveis, é importante para permitir o aumento da sua confiabilidade. A fim de ampliar a eficácia da manutenção preditiva, emerge a necessidade de implantar uma solução capaz de detectar e classificar falhas em estágio inicial e recomendar a equipe de manutenção a necessidade de programar a manutenção no ativo. Este trabalho propõe o uso de Rede Neural Convolutiva para classificar automaticamente as falhas de ativos que podem ser diagnosticadas através dos sinais de vibração mecânica com o objetivo de apoiar a detecção, o diagnóstico e a classificação de falhas. Os dados experimentais utilizados neste trabalho foram coletados através da base de dados denominada *Machinery Fault Database* que possui um conjunto de condições normais e sequências de falhas geradas em uma bancada de simulação de falhas, utilizando técnica *data driven*. Os sinais de vibração no domínio do tempo foram pré-processados usando Transformada Rápida de Fourier e transformados no domínio da frequência, para então serem processados no modelo desenvolvido e denominado de Manutenção Preditiva Utilizando Rede Neural Convolutiva (MP-CNN). Este trabalho utilizou os dados de três sensores de vibração unidirecionais do tipo acelerômetro instalados no mancal interno, já que este é um ponto de medição amplamente utilizado e presente em processos industriais. O valor médio da métrica *accuracy* resultante do modelo foi de 99,58%, mostrando sua capacidade de detectar e classificar falhas em ativos.

Palavras chaves: Análise de vibração mecânica, Redes Neurais Convolutivas, Aprendizagem Profunda, *Data Driven*, máquinas rotativas

ABSTRACT

Rotating machines are widely used in industrial processes, whose continuous use causes wear in several of its components, requiring the stop of the asset, to replace them. Due to the importance of the assets for the operational continuity of an industry, it is valuable to constantly monitor them in order to predict failure, perform scheduled maintenance and minimize the possibility of failure in unplanned conditions. The strategy of predictive maintenance, which performs constant monitoring of the assets through the main variables, is important to allow the increase of their reliability. In order to increase the effectiveness of predictive maintenance, the need emerges to implement a solution capable of detecting and classifying failures at an early stage and to recommend to the maintenance team the need to schedule the maintenance on assets. This work proposes the use of Convolutional Neural Network to automatically classify asset faults that can be diagnosed through mechanical vibration signals in order to support fault detection, diagnosis and classification. The experimental data used in this work were collected through the Machinery Fault Database which has a set of normal conditions and fault sequences generated in a fault simulation bench, using data driven technique. The vibration signals in the time domain were preprocessed using Fast Fourier Transform and transformed in the frequency domain, to be then processed in the model developed and called Predictive Maintenance Using Convolutional Neural Network (MP-CNN). This work used the data from three unidirectional vibration sensors of the accelerometer type installed in the internal bearing, since this is a widely used measurement point present in industrial processes. The average accuracy metric value resulting from the model was 99.58%, showing its ability to detect and classify failures in assets.

Keywords: Vibration analysis, Convolutional Neural Networks, Deep Learning, [Data Driven](#), rotating machinery

LISTA DE FIGURAS

Figura 1 – Curva P-F e custo de reparo de manutenção.....	21
Figura 2a – Sinal de vibração no domínio do tempo.....	22
Figura 2b – Sinal de vibração no domínio da frequência.....	22
Figura 3 – Representação do nível pico a pico e RMS.....	23
Figura 4a – Espectro típico de vibração mecânica provocada por desbalanceamento.....	26
Figura 4b – Gráfico no domínio do tempo provocado por desbalanceamento.....	26
Figura 5a – Espectro típico de vibração mecânica provocada por desbalanceamento.....	27
Figura 5b – Gráfico no domínio do tempo provocado por desbalanceamento.....	27
Figura 6a – Gráfico no domínio do tempo provocado por falha no rolamento em função de esfera danificada.....	30
Figura 6b – Gráfico no domínio do tempo provocado por falha no rolamento em função de gaiola danificada.....	30
Figura 6c – Gráfico no domínio do tempo provocado por falha no rolamento em função da pista externa danificada.....	30
Figura 7a – Simulador de falha em maquinário.....	31
Figura 7b – Desenho esquemático do MFS apresentando a posição de cada componente.....	31
Figura 8 – Novo paradigma na programação.....	34
Figura 9 – Fluxo de tratamento dos dados.....	36
Figura 10 – Representação de aprendizagem profunda e camadas.....	37
Figura 11 – Representação das funções de aprendizagem profunda.....	38
Figura 12 – Representação entre as modelos de Inteligência Artificial.....	38
Figura 13 – Aprendizagem supervisionada.....	41
Figura 14 – Processo de convolução.....	44
Figura 15 – Operação <i>maxpooling</i>	45
Figura 16 – Processo da camada <i>pooling</i>	45
Figura 17 – Camada Totalmente Conectada.....	45
Figura 18 – Função ReLU X Sigmoidais.....	46
Figura 19 – Ilustração da função <i>dropout</i>	47
Figura 20 – Resultado de uma função <i>Softmax</i>	48
Figura 21 – Visão geral de uma amostra de CNN.....	49
Figura 22 – Comparação da taxa de erro utilizando ReLU e $\tanh(x)$	51
Figura 23 – Comparação da taxa de erro entre DCNN e ADCNN.....	52

Figura 24 – Comparativo de convergência utilizando diferentes funções.....	55
Figura 25 – Representação simbólica do modelo proposto.....	59
Figura 26 – Estrutura das camadas do modelo CNN.....	60
Figura 27 – Resumo do modelo CNN construído para extrair características.....	61
Figura 28 – Resultado da convergência (<i>loss</i>).....	62
Figura 29 – Gráficos representando as falhas no domínio do tempo e domínio da frequência.....	63

LISTA DE TABELAS

Tabela 1 – Especificação dos componentes.....	32
Tabela 2 – Listagem dos cenários.....	33
Tabela 3 – Matriz de confusão.....	39
Tabela 4 – Função <i>Loss</i>	48
Tabela 5 – Comparação dos erros dos modelos de Inteligência Artificial.....	54
Tabela 6 – <i>Accuracy</i> na classificação de falha de rolamento utilizando o classificador LRC e diferentes combinações de algoritmos de compressão.....	56
Tabela 7 – <i>Accuracy</i> na classificação de falha de rolamento utilizando o classificador ANN e diferentes combinações de algoritmos de compressão.....	56
Tabela 8 – <i>Accuracy</i> na classificação de falha de rolamento utilizando o classificador SVM e diferentes combinações de algoritmos de compressão.....	57
Tabela 9 – Comparação entre os extratores de características e classificadores.....	58
Tabela 10 – Métricas do modelo de classificação de falhas de acordo com a severidade.....	64
Tabela 11 – Matriz de confusão – Classificação de falhas de acordo com a severidade.....	66
Tabela 12 – Métricas do modelo de classificação de falhas.....	67
Tabela 13 – Matriz de confusão – Classificação de falhas.....	68
Tabela 14 – Comparativo com outros trabalhos acadêmicos utilizando a base de dados MaFaulDa.....	68
Tabela 15 – <i>Accuracy</i> do modelo XGBoost + Transformada <i>Wavelet</i> de <i>Haar</i>	69

LISTA DE ABREVIATURAS E SIGLAS

AdaBN	<i>Adaptative Batch Normalization</i>
ADCNN	<i>Adaptive Deep CNN</i>
ADCNN	<i>Adaptative Convolutional Neural Network</i>
AM	Aprendizagem de Máquina
ANN	<i>Artificial Neural Network</i>
BPFI	<i>Ball Pass Frequency of Inner ring</i>
BPFO	<i>Ball Pass Frequency of Outer Ring</i>
BSF	<i>Ball Spin Frequency</i>
CNN	<i>Convolutional Neural Network</i>
CWRU	<i>Case Western Reserve University</i>
DCNN	<i>Deep Convolutional Neural Network</i>
DNCNN	<i>Deep Normalized Convolutional Neural Network</i>
FFT	<i>Fast Fourier Transform</i>
FN	Falso Negativo
FP	Falso Positivo
FTF	<i>Fundamental Train Frequency</i>
g	Gramma
GPU	<i>Graphics Processing Unit</i>
HPC	<i>High Performance Computer</i>
IA	Inteligência Artificial
IC	Inteligência Computacional
ILSVRC	<i>ImageNet Large-Scale Visual Recognition Challenge</i>
KNN	<i>K-nearest Neighbors</i>
MFS	<i>Machinery Fault Simulator</i>
MLP	<i>Multilayer Perceptron</i>
mm	Milímetro
MP-CNN	Manutenção Preditiva utilizando Convolutional Neural Network
PdM	<i>Predictive Maintenance</i>
RAM	<i>Random Access Memory</i>
RBFN	<i>Radial Basis Function Networks</i>
ReLU	<i>Rectified Linear Unit</i>
RMS	<i>Root Mean Square</i>
RNA	Rede Neural Artificial

RNC	Rede Neural Convolucional
ROI	<i>Return on Investment</i>
RPM	Rotações por Minuto
s	Segundo
SBM	<i>Similarity-Based Modeling</i>
SI	Sistema Internacional de Unidades
SVM	<i>Support Vector Machines</i>
SVRM	<i>Support Vector Regression Machine</i>
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo
WDCNN	<i>Deep Convolutional Neural Networks with Wide First-Layer Kernel</i>

SUMÁRIO

1. INTRODUÇÃO	15
1.1. JUSTIFICATIVA.....	16
1.2. OBJETIVO GERAL.....	17
1.3. OBJETIVOS ESPECÍFICOS	17
1.4. ESTRUTURA DA DISSERTAÇÃO	18
2. REFERENCIAL TEÓRICO	19
2.1. ESTRATÉGIAS DE MANUTENÇÃO	19
2.1.1. MANUTENÇÃO CORRETIVA	19
2.1.2. MANUTENÇÃO PREVENTIVA.....	20
2.1.3. MANUTENÇÃO PREDITIVA	20
2.2. VIBRAÇÃO MECÂNICA.....	21
2.2.1. ANÁLISE DE VIBRAÇÃO NO DOMÍNIO DO TEMPO.....	22
2.2.2. ANÁLISE DE VIBRAÇÃO NO DOMÍNIO DA FREQUÊNCIA – ANÁLISE ESPECTRAL.....	24
2.2.3. ANÁLISE DE HARMÔNICOS	25
2.2.4. CAUSAS DE VIBRAÇÃO.....	25
a) Desbalanceamento.....	26
b) Desalinhamento	26
c) Falha de rolamento	27
2.3. BASE DE DADOS - MAFAULDA.....	30
2.3.1. FALHAS SIMULADAS	32
2.4. APRENDIZAGEM DE MÁQUINA.....	34
2.5. APRENDIZAGEM PROFUNDA	36
2.5.1. ACCURACY, PRECISION, RECALL E F1-SCORE	39
2.5.2. PROCESSO DE APRENDIZAGEM.....	40
a) Aprendizagem supervisionada.....	40
b) Aprendizagem não supervisionada	41
2.5.3. REDES NEURAS CONVOLUCIONAIS (CONVOLUCIONAL NEURAL NETWORK - CNN)	41
a) Camada de convolução.....	43
b) Camada <i>Pooling</i>	44
c) Camada Totalmente Conectada (<i>Fully Connected</i>).....	45
d) Normalização de batches (<i>Batch Normalization</i>)	46
e) Função de ativação da camada de convolução	46
f) Dropout	46
g) Função <i>softmax</i>	47
h) Função custo (<i>loss</i>)	48
3. TRABALHOS RELACIONADOS	50
3.1. CLASSIFICAÇÃO DE IMAGENS.....	50

3.2.	CLASSIFICAÇÃO DE FALHA.....	51
3.3.	PRÉ-PROCESSAMENTO DOS DADOS	55
3.4.	CONCLUSÃO.....	58
4.	MÉTODO PROPOSTO	59
4.1.	TRATAMENTO DOS DADOS	59
4.2.	DESENVOLVIMENTO EXPERIMENTAL	60
5.	RESULTADOS E DISCUSSÕES	62
6.	CONCLUSÃO	71
6.1.	SUGESTÕES	72
	REFERÊNCIA	73
	GLOSSÁRIO	77
	APÊNDICE A – PUBLICAÇÃO E SUBMISSÃO DE ARTIGOS.....	84
	APÊNDICE B – DEPÓSITO DE PATENTE.....	85

1. INTRODUÇÃO

O uso de técnicas não invasivas para o monitoramento de ativos¹, como a análise de vibração mecânica, é importante para identificar as suas condições de funcionamento, detectar falhas, identificar as causas e estimar o tempo quando o ativo poderá falhar.

Com o objetivo de otimizar o monitoramento dos ativos, as indústrias demandam soluções que possam ser mais assertivas nas recomendações de manutenção. Portanto, é possível desenvolver soluções que possam colaborar com a equipe de manutenção e informar com antecedência a necessidade de programar a manutenção do ativo, informar os componentes necessários a serem substituídos e minimizar a possibilidade de paradas não programadas.

A evolução da monitoração das condições e desempenho dos ativos, além da possibilidade de detectar falhas, têm sido possíveis em virtude do desenvolvimento de novas tecnologias, da rápida queda no custo dos sensores, da possibilidade de armazenamento de grande quantidade de dados (*big data*), do aumento da capacidade de processamento do computador e do desenvolvimento de soluções que possam analisar os dados para prever, diagnosticar ou classificar falhas. (LU *et al.*, 2017)

Com o desenvolvimento de técnicas baseadas em *data driven*, *big data*, mineração de dados, inteligência artificial e modelos de diagnóstico de falhas, baseados em aprendizagem de máquina e aprendizagem profunda, soluções inteligentes têm sido aplicadas na detecção, diagnóstico e prognóstico de falhas em ativos. (LU *et al.*, 2017)

Aprendizagem de máquina (AM) é um subcampo específico da Inteligência Artificial (IA) que visa desenvolver programas (chamados modelos) automaticamente a partir da exposição a dados de treinamento. (CHOLLET, 2018)

Soluções desenvolvidas através de técnicas de aprendizagem de máquinas no diagnóstico de falhas do motor de indução ganharam popularidade ao longo dos anos. No entanto, as soluções utilizando AM enfrentam um grande desafio para expandir o campo de aplicabilidade em razão da necessidade de extração das características estatísticas (curtose, média, desvio padrão, fator de crista) como forma de pré-tratamento dos dados. O tratamento prévio dos dados provoca o excesso de trabalho para gerar as principais características dos dados, a fim de que o modelo possa obter um bom desempenho dos classificadores, conforme métricas a serem

¹ O termo “ativo” no decorrer dessa dissertação será utilizado conforme definição estabelecida na norma ISO 55000 que significa “Um bem, uma coisa ou uma entidade, que tem um valor potencial ou real para uma organização”. Na dissertação será utilizado para exprimir equipamentos rotativos, equipamentos industriais ou máquinas rotativas

estabelecidas. Por outro lado, a aprendizagem profunda utiliza um modelo baseado na representação de características e abre um novo horizonte em que os descritores de características sejam extraídos dos sinais brutos, pelo próprio modelo. (CHATTOPADHYAY *et al.*, 2018)

A aprendizagem profunda está ganhando popularidade em função da possibilidade de identificar as principais características (*features*) das variáveis através dos dados brutos, o que provoca um menor tempo de processamento e pode gerar modelos que consigam prever, diagnosticar e classificar as falhas com grande precisão e obter uma maior *accuracy*. (CHATTOPADHYAY *et al.*, 2018)

A aprendizagem profunda é cada vez mais utilizada em diversas soluções como visão computacional, som e reconhecimento facial. No campo de diagnóstico de falhas de motores, o uso do sinal bruto e técnicas de *data driven*, estão sendo empregados para realizar a modelagem dos dados e evitar a necessidade de equações matemáticas para modelar o equipamento. Aprendizagem profunda, além de permitir o uso dos dados brutos e técnicas de *data driven*, está ampliando o campo de aplicação, como diagnóstico de falha de ativos, porque não requer trabalho extra para geração das características do sinal, evitando a conversão de dados brutos em características estatísticas que são exigidas em aprendizagem de máquina. (ZHAO *et al.*, 2018)

1.1. JUSTIFICATIVA

Máquinas rotativas são amplamente empregadas nas indústrias. São ativos importantes, cujo uso contínuo acarreta desgaste nos seus componentes, e necessitam de um monitoramento constante para evitar que o desgaste implique na necessidade de paradas não programadas e aumente os custos operacionais².

A necessidade de otimização dos custos operacionais, do aumento da confiabilidade dos ativos e a demanda de produtos com qualidade dentro dos limites estabelecidos, além da segurança operacional, provocam nas indústrias a necessidade de implementação de soluções capazes de realizar a detecção de falhas de ativos em estágio inicial e de gerar informações para a equipe de manutenção. Com essa gama de dados a equipe poderá realizar a programação da manutenção do ativo, reduzindo o número de paradas operacionais não programadas, os custos de manutenção e o tempo de parada do ativo.

² São as despesas variáveis que mantêm uma empresa em funcionamento e que podem variar conforme a produção, no caso da manutenção de ativos.

A vibração mecânica é utilizada por ser um sinal importante na análise de falha em diversos equipamentos rotativos. A partir da análise de vibração, é possível acompanhar a severidade, a condição do equipamento, assim como identificar as possíveis causas da falha.

Uma das possibilidades para classificação da causa da falha do ativo é através da análise dos sinais de vibração, que é facilitada quando analisada através dos gráficos no domínio da frequência e correlaciona-los com a velocidade de rotação do eixo do ativo. As falhas possuem características próprias que conseguem ser analisadas dentro do domínio da frequência.

Com o objetivo de implementar melhorias e aumentar a confiabilidade no ambiente industrial, verificou-se a possibilidade de desenvolver uma solução que pudesse realizar o diagnóstico e a classificação da falha utilizando sinais de vibração, em um menor tempo de resposta na classificação de falhas, realizar a classificação em estágio inicial além de não necessitar do fator humano para classificar a falha.

Determinou-se a métrica *accuracy* como parâmetro de aceitação do modelo. Foi determinado o valor mínimo de 99%, conforme o item 1.2, em função da leitura de artigos científicos e todos que utilizaram a base de dados MaFaulDa apresentaram *accuracy* inferior a 99%.

1.2. OBJETIVO GERAL

Desenvolver um modelo utilizando métodos de rede neural convolucional para classificar falhas de ativos, utilizando sinais de vibração mecânica.

O sistema deverá ser capaz de classificar as falhas de desbalanceamento, desalinhamento e de rolamento³ (esfera, gaiola e pista externa), utilizando sinais de vibração mecânica e apresentar *accuracy* maior que 99%.

1.3. OBJETIVOS ESPECÍFICOS

Serão seguidas as etapas abaixo para permitir alcançar o objetivo geral:

- Elaborar um estudo de prospecção e revisão utilizando informações científicas contidas nas bases de dados;
- Identificar os modelos de aprendizagem profunda que possam realizar a classificação de falha;
- Modelar e testar o sistema utilizando a base de dados *Machinery Fault Database* (MaFaulDa);

³ Optou-se por usar, neste trabalho, o termo rolamento como um componente mecânico que tem a função de transmitir uma ação de rotação a um determinado objeto.

- Realizar a otimização dos hiperparâmetros do modelo que seja capaz de classificar falhas causadas por desbalanceamento, desalinhamento ou rolamento e que possam ser identificadas através dos sinais de vibração mecânica;
- Avaliar o desempenho do modelo com dados conhecidos e certificar que a métrica *accuracy* seja superior a 99%.

1.4. ESTRUTURA DA DISSERTAÇÃO

O capítulo dois é constituído de referencial teórico, introdução e descrição sobre os fundamentos teóricos que suportam o desenvolvimento dessa dissertação: estratégia de manutenção (corretiva, preventiva e preditiva); vibração mecânica; base de dados MaFaulDa; aprendizagem de máquina e aprendizagem profunda.

O capítulo três apresenta os estudos sistematizados relacionados ao tema, o estado-da-arte, os resultados obtidos em cada pesquisa e a contribuição para este trabalho. Não será possível realizar um estudo comparativo, visto que alguns trabalhos somente realizaram estudos qualitativos.

O capítulo quatro refere-se ao método utilizado no desenvolvimento da pesquisa, o hardware e os softwares utilizados no desenvolvimento do modelo.

O capítulo cinco consiste nos resultados e compara-os com os trabalhos que utilizaram a mesma base de dados.

O capítulo seis apresenta a conclusão com os elementos mais relevantes obtidos, bem como as limitações da proposta e sugestões de trabalhos futuros.

2. REFERENCIAL TEÓRICO

O principal objetivo deste capítulo é contextualizar o trabalho que será realizado, apresentando um embasamento teórico que possa contribuir no entendimento da solução que será apresentada no capítulo quatro.

Será apresentada nesse capítulo uma descrição formal sobre as estratégias de manutenção, vibração mecânica, a base de dados MaFaulda, aprendizagem de máquina e aprendizagem profunda, que subsidiaram a abordagem proposta neste trabalho de dissertação.

2.1. ESTRATÉGIAS DE MANUTENÇÃO

A fim de alcançar um desempenho de disponibilidade de ativos estabelecido pelas metas das empresas, cada vez mais elas estão empreendendo esforços para melhorar a qualidade, a produtividade e reduzir os custos. Parte destes esforços tem incluído uma estratégia de manutenção que permite o prolongamento da vida útil do ativo, melhora a disponibilidade e mantém o ativo em condições adequadas. (SWANSON, 2001)

A implementação do plano de manutenção permite definir a melhor estratégia de acordo com a classificação de criticidade do ativo. Os principais critérios utilizados para classificação de criticidade dos ativos são: segurança do processo, possibilidade de afetar a qualidade do produto final, regime de trabalho do ativo, efeito da falha sobre interrupção do processo, frequência de falha e custo de manutenção. Utilizando esses critérios, conforme Márquez (2007), é possível determinar, através da Matriz de Criticidade, juntamente com a Análise Qualitativa de Criticidade, o nível de criticidade e definir o tipo de manutenção (corretiva, preventiva ou preditiva) requerida para o ativo.

Conforme Márquez (2007), a definição de estratégia de manutenção busca sempre dois critérios:

- Os mais baixos custos de manutenção;
- A maior disponibilidade dos ativos.

2.1.1. Manutenção Corretiva

É uma estratégia de manutenção reativa que espera pela falha do ativo, antes que seja realizada qualquer ação de manutenção. Essa estratégia é utilizada quando a falha do ativo não resulta em infortúnios humanos, monetários e operacionais. (SWANSON, 2001)

Trata-se da atuação para correção da falha ou redução do desempenho do ativo. Em função da possibilidade de paradas inesperadas do ativo, poderá ocasionar um maior tempo de parada do processo, altos custos de estoques de peças sobressalentes e baixa confiabilidade do ativo.

2.1.2. Manutenção Preventiva

É a estratégia de manutenção aplicada quando o componente do ativo é substituído de acordo com determinado padrão de uso, por exemplo: tempo e ciclos de uso do ativo.

A manutenção preventiva obedece a um padrão previamente definido, algumas vezes pelo fabricante, que estabelece paradas periódicas com a finalidade de substituir o componente.

Este tipo de manutenção depende da probabilidade estimada de que o equipamento falhe no intervalo especificado e envolve a programação de paradas regulares do ativo. Essa estratégia de manutenção pode reduzir as falhas antes que elas ocorram, mas também leva a um aumento dos custos de manutenção, uma vez que os componentes são substituídos quando isso não é necessariamente exigido. (LACEY, 2017)

O trabalho realizado pode incluir lubrificação do equipamento, substituição de componentes, limpeza e ajuste. O ativo também pode ser inspecionado para detectar sinais de deterioração durante os trabalhos de manutenção preventiva.

Os benefícios que podem ser obtidos através da implementação da manutenção preventiva, dentre outros, são a redução da probabilidade de avarias nos ativos e o prolongamento da sua vida útil. As desvantagens são: necessidade de interromper a produção em intervalos programados para executar a manutenção e o aumento de custos.

2.1.3. Manutenção Preditiva

A manutenção preditiva (PdM) é frequentemente referida como manutenção baseada em condições, que é uma estratégia de monitorar a condição do ativo à medida que ele está funcionando.

A monitoração das condições operacionais do ativo tem como objetivo acompanhar o efeito de possível desgaste dos componentes e realizar uma parada programada. Desta forma, a manutenção pode ser planejada com a possibilidade de substituir os componentes que apresentem sinais de deterioração e no momento mais oportuno. (LACEY, 2017).

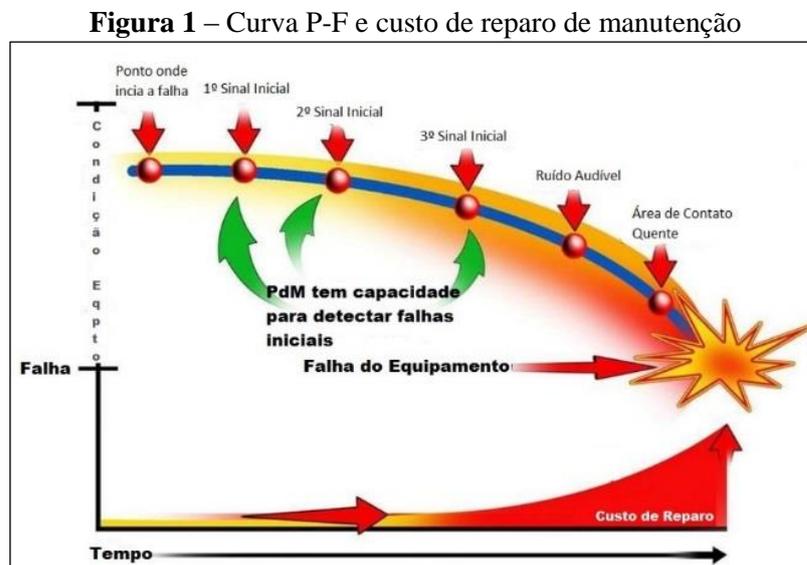
Os equipamentos de monitoração são usados para medir a condição física do equipamento, através das variáveis como temperatura, pressão, vazão, vibração, ruído, lubrificação e corrosão. Quando um destes indicadores atinge um determinado nível, são realizados trabalhos para restaurar o ativo. (SWANSON, 2001)

A implementação deste tipo de estratégia de manutenção requer investimentos na aquisição de sensores, sistemas de monitoramento e controle e treinamentos.

Tal como na manutenção preventiva, a manutenção preditiva reduz a probabilidade de falhas dos ativos. O benefício adicional vem da possibilidade de realizar a manutenção de forma programada e antecipada, reduzindo os custos de manutenção.

Dados do Departamento de Energia dos EUA indicam que a manutenção preditiva é extremamente eficaz na redução de custos. Efetivando uma estratégia de manutenção preditiva haverá um aumento de dez vezes no Retorno Sobre o Investimento (ROI), 25 a 30% de redução nos custos de manutenção, 70 a 75% de redução em paradas inesperadas. Em contraposição a manutenção corretiva custa 3 a 4 vezes mais que a manutenção preventiva. (LAVI, 2018)

A Figura 1 retrata o aumento relativo de custo de reparo quando não ocorre a antecipação de predição de falha.



2.2. VIBRAÇÃO MECÂNICA

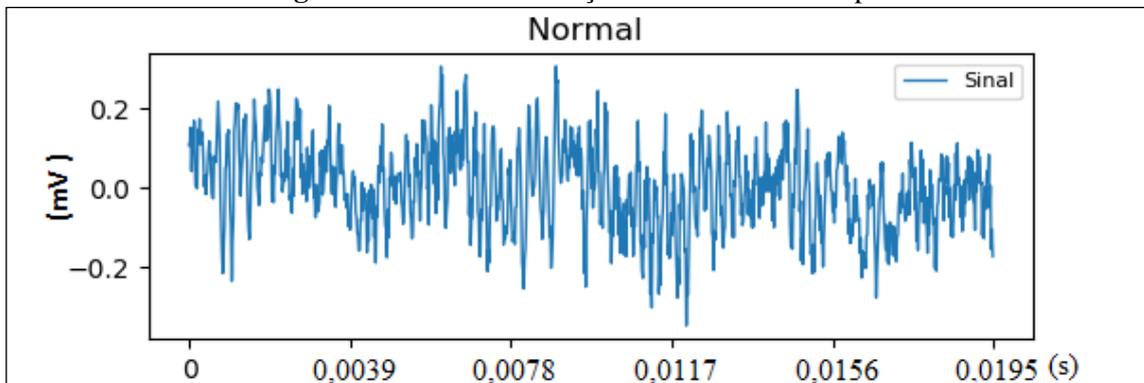
A monitoração de ativos através da análise de vibração mecânica é uma das técnicas mais utilizadas na estratégia de manutenção preditiva, em função de permitir determinar a condição do ativo e gerar informações que possibilitam diagnosticar os componentes que mostram sinais de deterioração. Portanto, através desta técnica é possível aumentar a confiabilidade do ativo e reduzir os custos operacionais. (LACEY, 2017)

Os sensores de vibração captam os sinais e disponibilizam os dados para que sejam tratados e apresentados em formato de gráficos, tabelas e gerarem informações estatísticas, que possam auxiliar os especialistas na interpretação dos dados.

Os gráficos gerados no domínio do tempo são bastante utilizados pelos especialistas para identificar a tendência de falha do equipamento e obter informações estatísticas, tais como valores de pico, pico a pico, *Root Mean Square* (RMS), curtose, média, desvio padrão. Embora

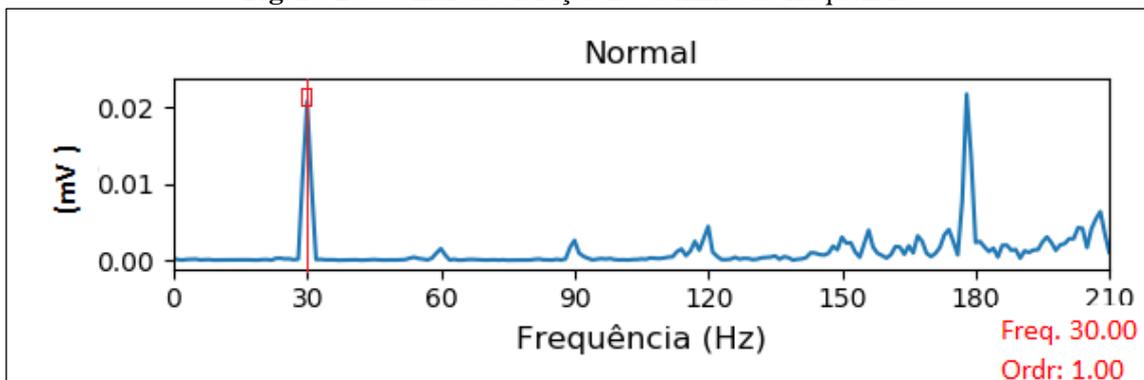
os gráficos no domínio do tempo sejam muito utilizados, existe a possibilidade de utilizar os gráficos no domínio da frequência para auxiliar na classificação da falha de vibração. As figuras 2a e 2b apresentam gráficos no domínio do tempo e no domínio da frequência, respectivamente.

Figura 2a – Sinal de vibração no domínio do tempo



Fonte: O autor, 2020

Figura 2b – Sinal de vibração no domínio da frequência



Fonte: O autor, 2020

2.2.1. Análise de Vibração no Domínio do Tempo

Uma das maneiras de extrair informações dos sinais de vibração no domínio do tempo e acompanhar as condições de funcionamento dos ativos é através das funções estatísticas e as principais são: média, raiz quadrada média (RMS), desvio padrão e variância. (ZHAO *et al.*, 2018)

a - Média

O valor médio de vibração pode ser calculado pela equação 2.1 e permite representar os valores médios do sinal sobre um período de tempo.

$$y(i\Delta t) = \frac{1}{N} \sum_{r=0}^{N-1} x(i\Delta t + rM\Delta t) \quad (\text{Eq. 2.1})$$

Em que:

$x(i\Delta t)$ são os valores do sinal original

$y(i\Delta t)$ é o valor médio do i -ésimo valor do sinal

M é o número de pontos por período

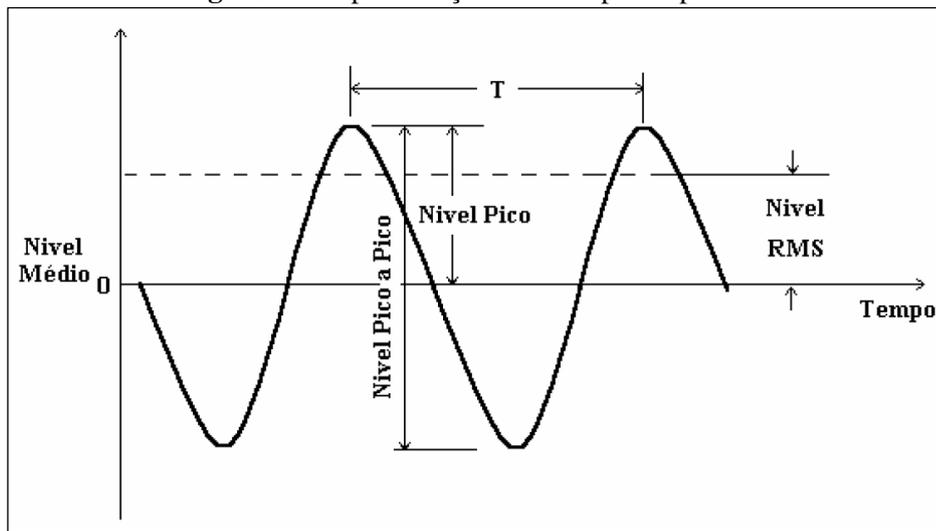
N é o número de períodos

Δt é o intervalo de aquisição.

b – Quantificação do nível de vibração

A amplitude de vibração é a característica que descreve a severidade da falha, podendo ser quantificada de diversas maneiras. A Figura 3 apresenta uma relação entre os níveis de pico a pico, nível de pico e o nível RMS de um sinal harmônico. (MEOLA, 2005)

Figura 3 – Representação do nível pico a pico e RMS



Fonte: (MEOLA, 2005)

Outros parâmetros que podem ser extraídos dos sinais de vibração no domínio do tempo são representados através das equações 2.2 a 2.4

$$\text{Nível de pico} = \frac{1}{2} (\max(y_k) - \min(y_k)) \quad (\text{Eq. 2.2})$$

$$\text{Nível RMS} = \sqrt{\left(\frac{1}{N} \sum_{K=1}^N (y_k)^2\right)} \quad (\text{Eq. 2.3})$$

$$\text{Desvio Padrão } (\sigma) = \sqrt{\left(\frac{1}{N-1} \sum_{K=1}^N (y_k - \bar{y})^2\right)} \quad (\text{Eq. 2.4})$$

O valor de pico a pico indica a excursão máxima da onda. Um valor importante para definir quando um componente do ativo atinge valores críticos, considerando-se máximas tensões ou folgas mecânicas. (MEOLA, 2005)

O valor RMS é o parâmetro para mensuração de nível mais relevante, por considerar o histórico da onda no tempo, evidenciando a severidade da energia contida no sinal, ou seja, a capacidade destrutiva desta vibração. (GARCIA, 2005)

2.2.2. Análise de Vibração no Domínio da Frequência – Análise Espectral

Quando os dados são analisados no domínio da frequência, a amplitude da assinatura de vibração demonstra um sinal da gravidade do problema e a frequência onde ocorre o pico pode indicar a fonte da falha. Vibrações anômalas são frequentemente o principal sinal de falha do ativo e as principais falhas são: desbalanceamento, desalinhamento, rolamento em desintegração e danos na engrenagem. (KUMAR *et al.*, 2018).

A análise espectral desempenha um papel importante na detecção e diagnóstico de falhas do ativo. No domínio do tempo é difícil, para os especialistas, identificar as contribuições individuais de falha, como desbalanceamento, desalinhamento, rolamentos, engrenagens, entre outras. No domínio da frequência, as falhas são mais fáceis de serem identificadas e podem ser mais facilmente relacionadas com as fontes de vibração, em função das assinaturas características de falha, conforme será explanado no capítulo 2.2.4.

A conversão dos dados no domínio do tempo para o domínio da frequência é realizada através da Transformada Rápida de Fourier (FFT).

Fourier demonstrou que a equação periódica no domínio do tempo pode ser representada por uma série de senos e cossenos, dadas pela equação 2.5

$$f(t) = \frac{A_0}{2} + \sum_{i=1}^{\infty} C_i e^{j(\frac{2\pi i}{T}t + \Phi_i)} \quad (\text{Eq. 2.5})$$

Em que:

$$C_i = \sqrt{(A_i)^2 + (B_i)^2}$$

$$A_i = \frac{2}{T} \int_0^T X(t) \cos\left(\frac{2\pi i}{T}t\right) dt$$

$$B_i = \frac{2}{T} \int_0^T X(t) \sin\left(\frac{2\pi i}{T}t\right) dt$$

$$\Phi_i = \text{tg}^{-1}\left(\frac{B_i}{A_i}\right)$$

$$j = -1$$

A_0 é o nível médio do sinal

A_i e B_i são os coeficientes de Fourier

A transformada de Fourier é apresentada através da equação 2.6.

$$f(\omega) = \int_{-\infty}^{+\infty} f(t) \cos \omega t dt + i \int_{-\infty}^{+\infty} f(t) \sin \omega t dt \quad (\text{Eq. 2.6})$$

A representação dos sinais de vibração no domínio da frequência resulta num gráfico similar a Figura 4a. Este tipo de gráfico recebe o nome de espectro de frequência e a análise recebe o

nome de análise espectral que é muito empregada para análise de assinatura de falha, pelos seguintes motivos:

1 – Uma função harmônica no domínio do tempo é representada por uma grande quantidade de pontos, enquanto que no domínio da frequência, bastam apenas três informações, sendo: a magnitude do termo cosseno, do termo seno e a frequência;

2 – As características do mecanismo de geração do sinal são normalmente mais perceptíveis no domínio da frequência, tais como: frequências características do mecanismo, ressonância do sistema mecânico, faixa de frequência na qual se concentram as vibrações, modulações em amplitude, entre outros.

A conversão dos dados no domínio do tempo para o domínio da frequência é realizada em dispositivos de coleta e processamento de dados de vibração através da Transformada Rápida de Fourier (FFT).

2.2.3. Análise de Harmônicos

Conforme será apresentado no capítulo 2.2.4, as causas de vibração possuem características típicas e podem ser identificadas com maior assertividade quando são analisadas utilizando gráficos que apresentem as informações no domínio da frequência.

A análise de harmônicos ou múltiplos da rotação relacionados com a velocidade de rotação é tratada como ordem. O harmônico na mesma frequência da velocidade de rotação é taxado como de primeira ordem; o harmônico referente a duas vezes a frequência de rotação é taxado como de segunda ordem e assim sucessivamente.

Observando a amplitude dos harmônicos e relacionando-a com a frequência fundamental, pode-se chegar a um diagnóstico da condição de funcionamento ou alteração em curso na máquina, tais como:

1 – Desbalanceamento: resulta, no gráfico de espectro, em um pico de amplitude na primeira ordem (frequência fundamental);

2 – Desalinhamento: gera um pico de amplitude na segunda ordem;

3 – Rolamentos: a harmônica gerada depende do tipo de falha e do modelo do rolamento.

Nos sistemas dedicados para análise de vibração, os gráficos podem ser analisados no domínio do tempo ou no domínio da frequência.

2.2.4. Causas de Vibração

Através do gráfico no domínio da frequência é possível analisar a ordem de vibração, a amplitude do sinal, assim como se esses valores ultrapassam os limites aceitáveis para determinado ativo, conforme estabelecido na norma ISO-10816 (ISO, 2015). Através dessas

informações, juntamente com a velocidade rotacional do eixo do ativo, é possível identificar a causa e severidade da falha.

A seguir serão listadas as principais falhas dos ativos que podem ser detectadas através da análise de vibração e serão escopo de análise deste trabalho.

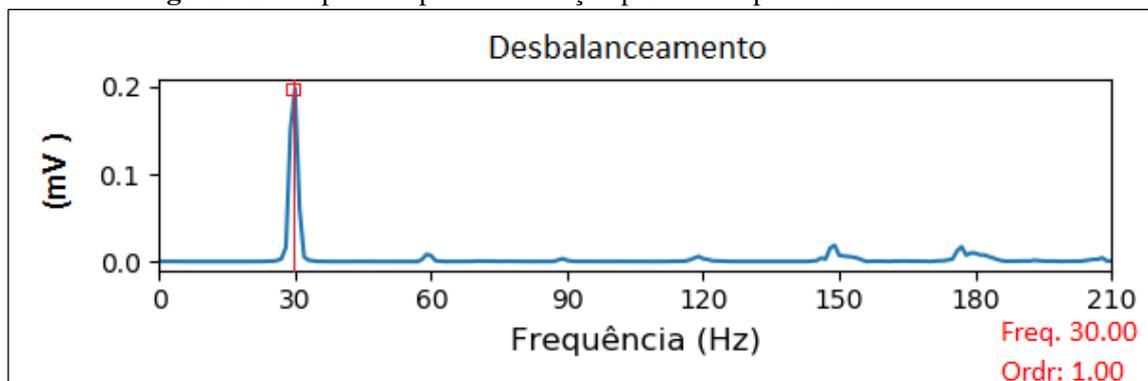
a) Desbalanceamento

As principais causas que podem provocar o desbalanceamento são: alteração no equilíbrio das forças radiais que atuam sobre o eixo do ativo, balanceamento errado, perda de parte do rotor por qualquer razão, depósito de material estranho em alguma das pás do rotor, empeno do eixo ou excentricidade entre os componentes acoplados.

A vibração resultante de um desbalanceamento puro, quando analisada no domínio da frequência, é reproduzida com um pico na própria frequência de rotação do eixo do ativo (ordem 1X RPM). A amplitude é proporcional a quantidade de desbalanceamento.

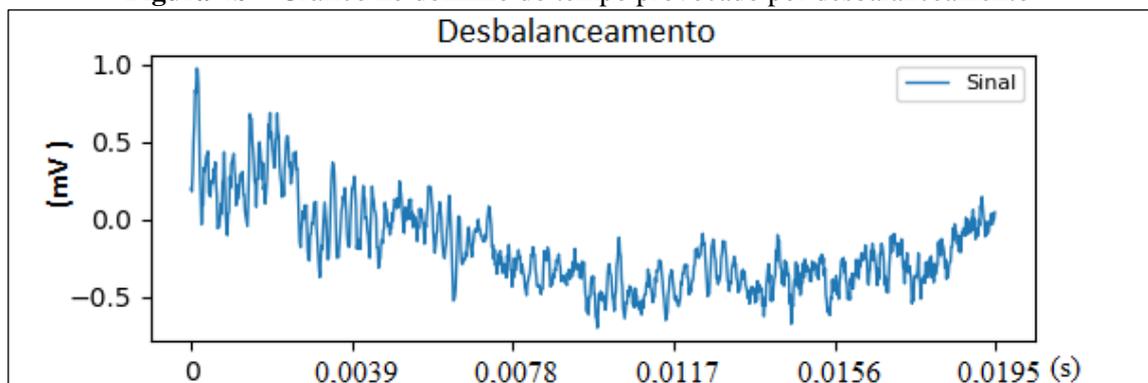
As Figuras 4a e 4b apresentam gráficos representativos da falha de desbalanceamento no domínio da frequência e no domínio do tempo, respectivamente.

Figura 4a – Espectro típico de vibração provocado por desbalanceamento



Fonte: O autor, 2020

Figura 4b – Gráfico no domínio do tempo provocado por desbalanceamento



Fonte: O autor, 2020

b) Desalinhamento

A vibração associada à falha de desalinhamento tem uma característica peculiar, pois o acoplamento do eixo possui a capacidade de absorver o desalinhamento e por essa razão a

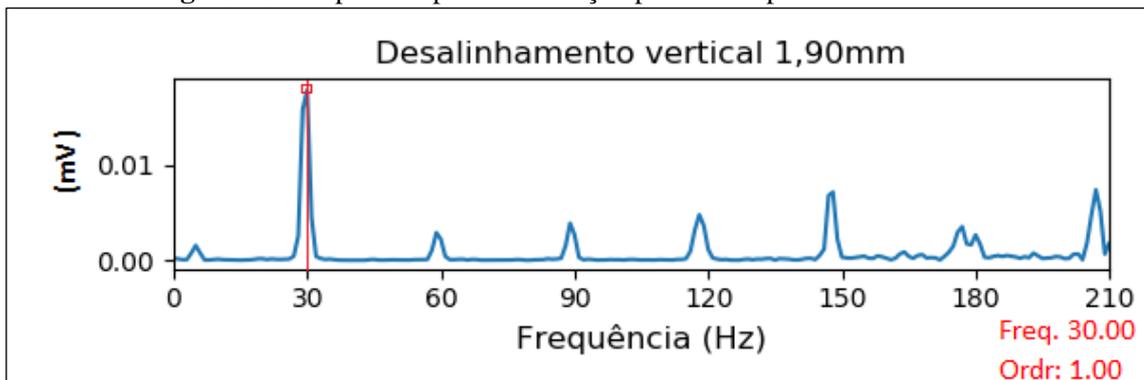
vibração só se manifesta quando as forças provocadas pelo desalinhamento são transmitidas ao rotor e mancais sob a forma de pré-carga. (FILHO, 2013)

Os principais problemas sob o aspecto de pré-carga em ativos são: desalinhamento angular, paralelo ou ambos e desalinhamento entre mancais ou engrenagens que podem ser provocados por falhas de montagem, falha na fixação do ativo na base e parafusos de fixação folgados.

O desalinhamento é reproduzido no espectro principalmente no componente de segunda, terceira e pode chegar à quarta ordem. Em algumas situações, o valor de pico pode ser observado na mesma frequência de velocidade de rotação do eixo do ativo e, quando ocorre, poderá ocasionar dúvidas quanto à causa de vibração, se é desalinhamento ou desbalanceamento.

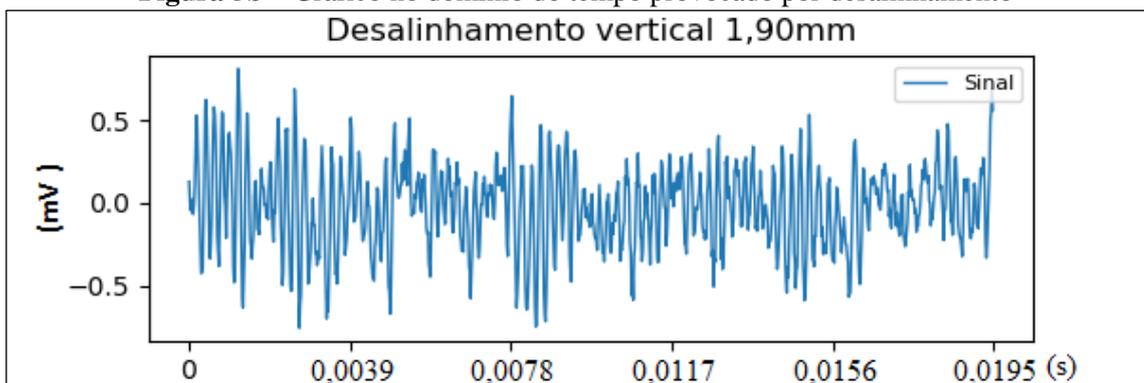
As Figuras 5a e 5b representam os sinais de desalinhamento no domínio da frequência e no domínio do tempo, respectivamente.

Figura 5a – Espectro típico de vibração provocado por desalinhamento



Fonte: O autor, 2020

Figura 5b – Gráfico no domínio do tempo provocado por desalinhamento



Fonte: O autor, 2020

c) Falha de rolamento

Muitos rolamentos falham prematuramente em serviço em função de contaminação; lubrificação deficiente; encaixe/ajuste inadequado; estocagem inadequada; temperaturas, desalinhamento e/ou desbalanceamento extremo. Todos esses fatores levam a um aumento de

vibração no rolamento e o monitoramento da condição tem sido usado por muitos anos para detectar a degradação antes que eles falhem catastroficamente. (LACEY, 2017)

Na maioria das situações, a vibração do rolamento não pode ser medida diretamente e a assinatura da vibração do rolamento é modificada pela estrutura do ativo. A medição é ainda mais complicada quando ocorrem vibrações simultâneas geradas por outras partes do ativo, tais como: motores elétricos, engrenagens, correias, sistemas hidráulicos, ressonâncias estruturais, entre outros. (LACEY, 2017)

Embora os sistemas e manuais dos fabricantes tragam as frequências características dos rolamentos, caso seja necessário, é possível calcular estas frequências através das equações 2.7 a 2.11.

$$f_{c/o} = \frac{fr}{2} \left[1 - \frac{d}{D} \cos \alpha \right] \quad (\text{Eq. 2.7})$$

$$f_{c/i} = \frac{fr}{2} \left[1 + \frac{d}{D} \cos \alpha \right] \quad (\text{Eq. 2.8})$$

$$f_{b/o} = Z * f_{c/o} \quad (\text{Eq. 2.9})$$

$$f_{b/i} = Z * f_{c/i} \quad (\text{Eq. 2.10})$$

$$f_b = \frac{D}{2d} fr \left[1 - \left(\frac{d}{D} \cos \alpha \right)^2 \right] \quad (\text{Eq. 2.11})$$

Em que:

$f_{c/o}$ = Frequência de giro da gaiola (FTF)

$f_{c/i}$ = Frequência fundamental da gaiola em relação ao anel interno

$f_{b/o}$ = Frequência de passagem das esferas na pista externa (BPFO)

$f_{b/i}$ = Frequência de passagem das esferas na pista interna (BPFI)

f_b = Frequência de giro das esferas (BSF)

f_r = Frequência rotacional do anel interno

D = Diâmetro do círculo de inclinação

d = Diâmetro das esferas

Z = Quantidade de esferas

α = Ângulo de contato

Geralmente, as frequências características do rolamento não são múltiplos inteiros do anel interno de frequência de rotação, o que ajuda a distingui-las de outras fontes de vibração. (LACEY, 2017)

O processo de degradação de um rolamento pode se iniciar na pista externa ou interna, em um dos componentes rolantes (rolos ou esferas) ou na gaiola, alastrando-se depois para os demais componentes.

Ao medir as frequências geradas por um rolamento, é possível identificar não só a existência de um problema, mas também a sua causa. Embora possa ser necessário apenas identificar que um rolamento está começando a se deteriorar, uma análise mais detalhada do sinal de vibração pode dar indícios sobre o que causou o problema em primeiro lugar. (LACEY, 2017)

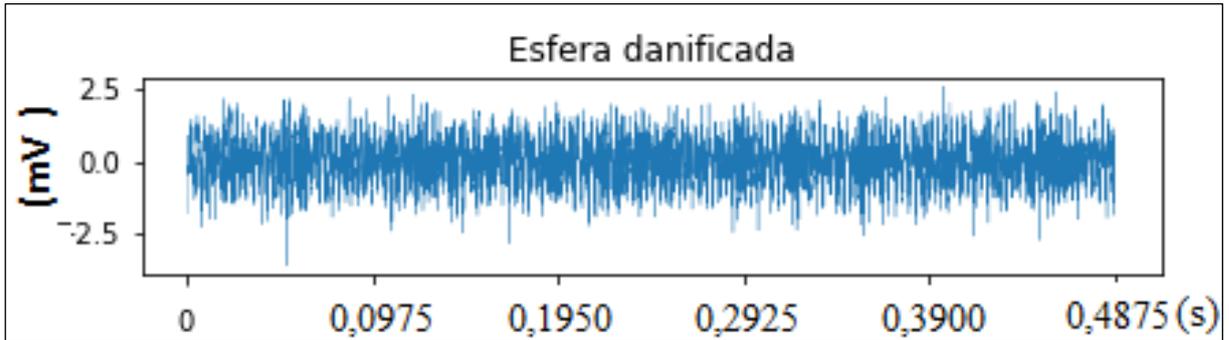
Quando ocorre o processo de falha de um único componente do rolamento, de uma esfera, por exemplo, pode ser visualizado, através do gráfico no domínio da frequência, no espectro de aceleração, um pico na frequência de passagem das esferas. Conforme a severidade da falha aumenta, ocorre um aumento nas frequências características e bandas laterais, seguido por uma queda na amplitude e um aumento de ruído nas bandas laterais com considerável vibração na frequência de rotação do eixo. (LACEY, 2017)

Um rolamento com falha nas pistas internas e externas pode gerar vibração na frequência correspondente ao dobro da frequência de rotação do eixo. Como a falha entra em contato com a gaiola, ele pode frequentemente modular outras frequências, quais sejam: frequência de falha de esfera, frequência de passagem de esfera (f_b) ou frequência de rotação do eixo e aparecem como uma banda lateral. A frequência de rotação da gaiola ($f_{c/o}$) pode ser gerada em uma gaiola desgastada ou danificada. (LACEY, 2017)

Nem sempre é possível medir com precisão a amplitude do sinal de vibração provocado pela falha de rolamento em função que cada ativo terá massa, rigidez e amortecimento diferentes. Mesmo ativos idênticos podem ter sistemas de amortecimento diferentes e isso pode afetar a amplitude do sinal de falha do rolamento defeituoso mesmo que tenha dimensão semelhante. (LACEY, 2017)

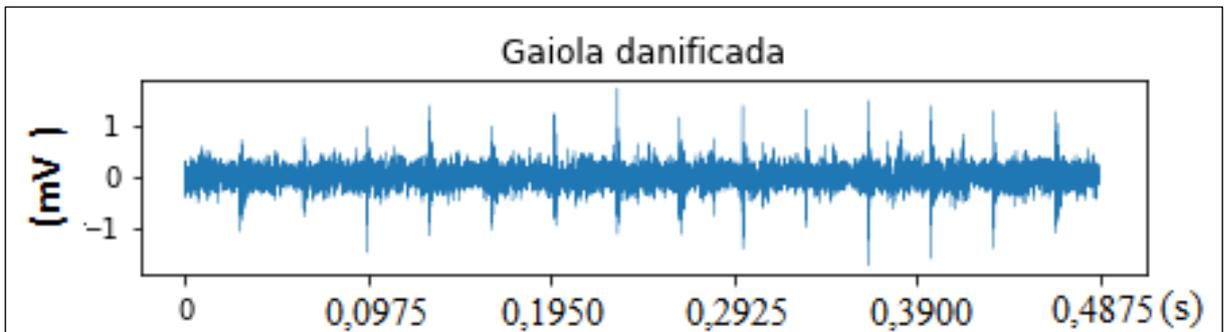
As condições de falha da pista externa têm um sinal bastante periódico; a falha da esfera pode ou não ser periódica, dependendo de várias razões, incluindo o nível de dano da esfera, a carga do rolamento e também a pista que a esfera descreve dentro da própria pista. A falha da gaiola gera uma distorção aleatória, que também depende do grau de dano e da carga do rolamento, conforme pode ser analisada através das Figuras 6a a 6c.

Figura 6a – Gráfico no domínio do tempo provocado por falha no rolamento em função de esfera danificada



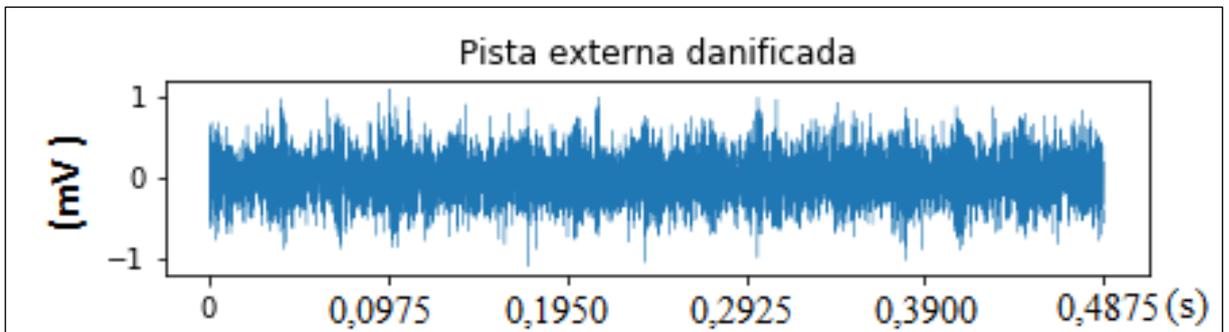
Fonte: O autor, 2020

Figura 6b – Gráfico no domínio do tempo provocado por falha no rolamento em função de gaiola danificada



Fonte: O autor, 2020

Figura 6c – Gráfico no domínio do tempo provocado por falha no rolamento em função da pista externa danificada



Fonte: O autor, 2020

2.3. BASE DE DADOS - MAFAULDA

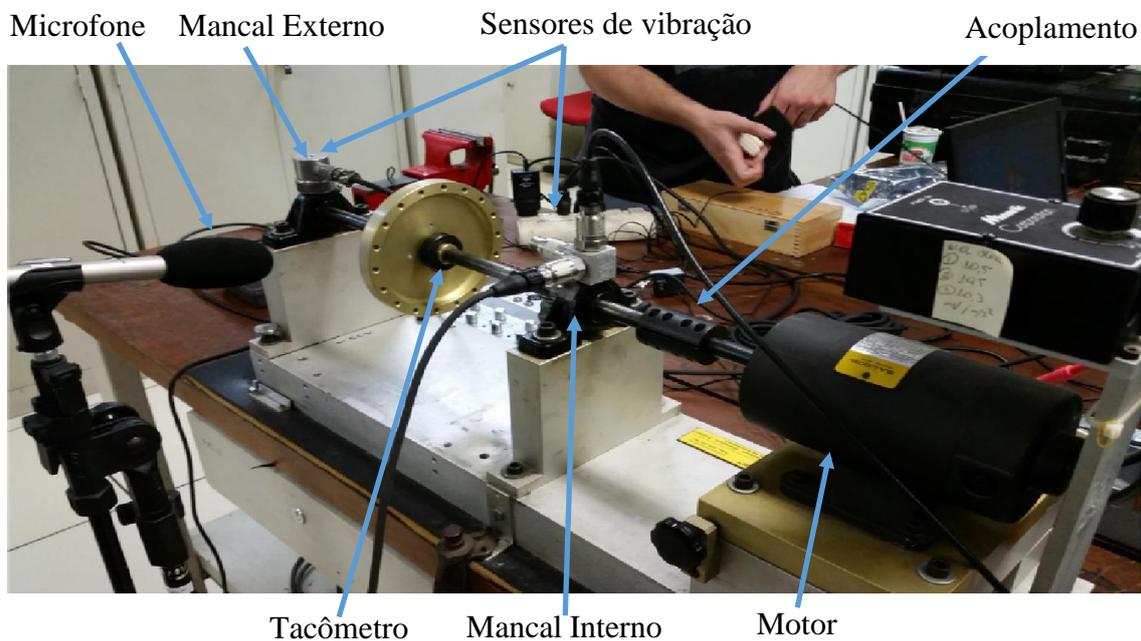
Em virtude da dificuldade de obter dados gerados por ativos instalados em ambiente industrial, no desenvolvimento do modelo, utilizou-se a base de dados de domínio público, denominada MaFaulDa, que possui dados gerados em condição de funcionamento normal assim como em falhas. (COPPE - UFRJ, 2018)

É uma base de dados que inclui falhas de desalinhamento, desbalanceamento e rolamento, captadas através de um acelerômetro do tipo triaxial (IMI Sensors - 604B31), três acelerômetros unidirecional (IMI Sensors - 601A01), um microfone e um tacômetro.

Os acelerômetros foram instalados para captarem os sinais de vibração na direção axial, radial e tangencial. Os acelerômetros unidirecionais foram instalados no mancal interno, conforme posição 4 da Figura 7b e o acelerômetro triaxial foi instalado no mancal externo, conforme posição 8 da Figura 7b.

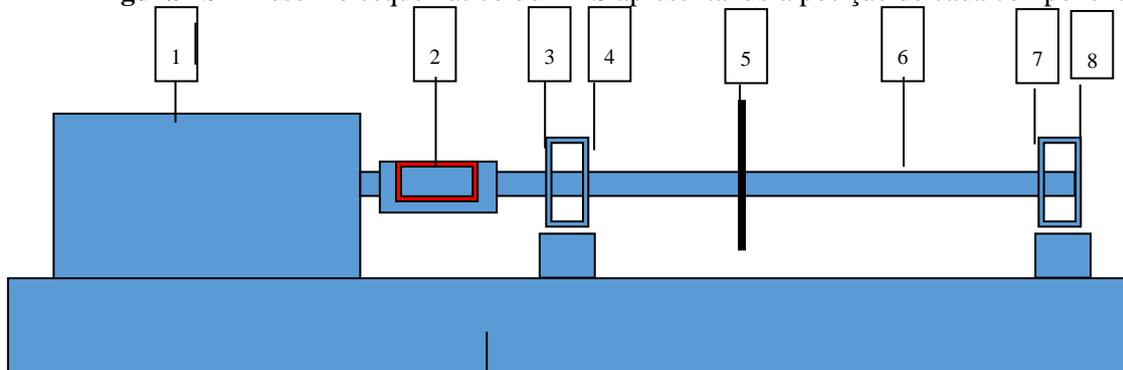
Os dados foram gerados através de um MFS ([Machinery Fault Simulator](#)), conforme Figuras 7a e 7b, que simula a dinâmica de motores com dois rolamentos de suporte de eixo. Através deste simulador é possível implementar diferentes tipos de falhas e obter os dados através dos respectivos tipos de sensores, utilizando técnicas de *data driven*.

Figura 7a – Simulador de falha em maquinário



Fonte: (Marins e colaboradores)

Figura 7b – Desenho esquemático do MFS apresentando a posição de cada componente



Posição: 1. Motor ; 2. Acoplamento ; 3, 7. Rolamentos ; 4, 8. Sensor de vibração; 5. Disco; 6. Eixo

Fonte: O autor, 2020

As principais características dos instrumentos, equipamentos e sensores utilizados no MFS estão listadas na Tabela 1.

Tabela 1 – Especificação dos componentes

Especificação	Modelo	Valor
Motor		¼ CV
Faixa de frequência		700 – 3600 rpm
Eixo	Diâmetro	16 mm
	Comprimento	520 mm
Sensor de vibração (Acelerômetro)	601A01 (Três sensores uniaxiais)	Fabricante: IMI Sensors
		Sensibilidade: 10,2 mV/(m/s ²)
		Faixa de Freq.: 0,27 a 10 000 Hz
		Faixa de medição: ±490 m/s ²
	604B31 (Um sensor triaxial)	Fabricante: IMI Sensors
		Sensibilidade: 10,2 mV/(m/s ²)
		Faixa de Freq.: 0,5 a 5 000 Hz
		Faixa de medição: ±490 m/s ²
Tacômetro	MT-190	Fabricante: Monarch Instrument
Microfone	SM81	Fabricante: Shure
Rolamento		Número de esferas: 8
		Diâmetro da esfera: 0,7145 cm
		Diâmetro da gaiola: 2,8519 cm
		FTF: 0,375 rpm
		BPFO: 2,998 rpm
		BPFI: 5,002 rpm
		BSF: 1,871 rpm
Sistema de aquisição de dados	NI 9234	Taxa de amostragem: 51,2 kHz

Fonte: (COPPE - UFRJ, 2018)

2.3.1. Falhas Simuladas

Através do [MFS](#) foram realizados testes, simulando diferentes tipos de falhas e severidades, e os dados foram armazenados na base de dados MaFaulDa, conforme as classes abaixo:

- a) Operação normal: corresponde à geração de dados dos ensaios sem aplicar qualquer tipo de falha. Estes dados foram gerados através de um conjunto de 49 diferentes velocidades, variando entre 737 e 3 686 rpm com passos de 60 rpm.
- b) Desbalanceamento: corresponde à geração de dados dos ensaios com a introdução de diferentes cargas assimétricas instaladas no disco, conforme item 5 da imagem 7b, tais como 6, 10, 15, 20, 25, 30 e 35g. Os dados dos ensaios foram compostos através de 333 diferentes conjuntos de dados e incluídos na base de dados. Não foram obtidos dados

quando se aplicou carga igual ou superior a 30 g e velocidade superior a 3 300 rpm porque todo o conjunto MFS vibrava.

- c) Desalinhamento horizontal paralelo: refere-se à geração de dados dos ensaios com o deslocamento horizontal do eixo do motor em 0,5; 1,0; 1,5 e 2,0 mm. Estes dados foram gerados através de um conjunto de 197 diferentes conjuntos de dados e incluídos na base de dados.
- d) Desalinhamento vertical paralelo: corresponde à geração de dados dos ensaios realizados com o deslocamento vertical do eixo do motor em 0,51; 0,63; 1,27; 1,4; 1,78 e 1,9 mm. Estes dados foram gerados através de um conjunto de 301 diferentes conjuntos de dados e incluídos na base de dados.
- e) Falhas nos rolamentos: refere-se à criação de uma base de dados com os testes realizados em três rolamentos defeituosos. Para cada rolamento foi gerado falhas na pista externa, na gaiola ou na esfera. As experiências foram realizadas utilizando um rolamento defeituoso de cada vez no mancal interno, conforme posição 4 na Fig. 7b, e no mancal externo, conforme posição 8 na Fig. 7b. Em razão da transmissão de vibração ser insuficiente, as falhas nos rolamentos eram praticamente imperceptíveis quando não havia desbalanceamento. Portanto, para induzir um efeito perceptível, três massas de 6, 20 e 35 g foram adicionadas, individualmente. Estes dados foram gerados através de um conjunto de 558 cenários para falhas no mancal interno e 513 cenários de falhas para o mancal externo e incluídos na base de dados.

A Tabela 2 resume os cenários gerados durante os experimentos.

Tabela 2 – Listagem dos cenários

Classe	Sub-classe		Quantidade
Normal			49
Desbalanceamento			333
Desalinhamento horizontal paralelo			197
Desalinhamento vertical paralelo			301
Rolamento	Mancal Interno	Esferas	186
		Gaiola	188
		Pista externa	184
	Mancal Externo	Esferas	137
		Gaiola	188
		Pista externa	188
Total			1.951

Fonte: O autor, 2020

Durante a simulação de falha, os sinais dos sensores foram capturados com uma taxa de amostragem de 51,2 kHz, através do sistema de aquisição de dados (NI 9234), por um tempo máximo de cinco segundos, compondo uma matriz de dados, para cada cenário de falha, de 250 000 linhas por oito colunas, correspondente aos sinais de um tacômetro, um microfone, um acelerômetro triaxial e três acelerômetros unidirecional.

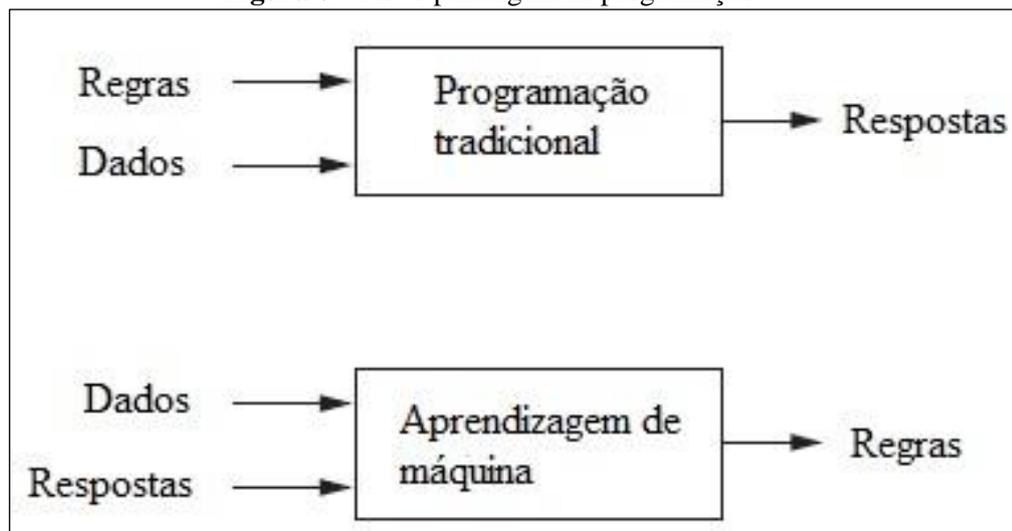
2.4. APRENDIZAGEM DE MÁQUINA

A aprendizagem de máquina apresenta um novo paradigma de programação. Na programação computacional tradicional, o código é gerado e executado para que os dados sejam processados de acordo com as regras estabelecidas e gerem respostas. Já na aprendizagem de máquina, os dados de entradas são processados juntamente com as respostas esperadas e o algoritmo gera as regras, sendo que essas regras podem ser reutilizadas em novos dados para gerarem respostas. (CHOLLET, 2018)

A Figura 8 representa graficamente esta alteração de programação dos algoritmos tradicionais e aprendizagem de máquina.

A aprendizagem de máquina está intimamente relacionada com as funções estatísticas, entretanto difere em pontos importantes. Ao contrário das funções estatísticas, que são aplicadas em pequena quantidade de dados, a aprendizagem de máquina possibilita o uso de grande quantidade de dados em que o uso de análise estatística clássica seria impraticável. (CHOLLET, 2018)

Figura 8 – Novo paradigma na programação



Fonte: Adaptado de CHOLLET, 2018

A aprendizagem de máquina é capaz de examinar grande quantidade de dados em busca de padrões e gerar um modelo que permite reconhecer esses padrões em novos dados. Esse modelo poderá ser compartilhado em outras soluções que podem fazer chamadas para os modelos

gerados, sem necessitar de realizar um novo modelo. Em outras palavras, aprendizagem de máquina pode ajudar a criar soluções mais inteligentes. (CHAPPELL, 2018)

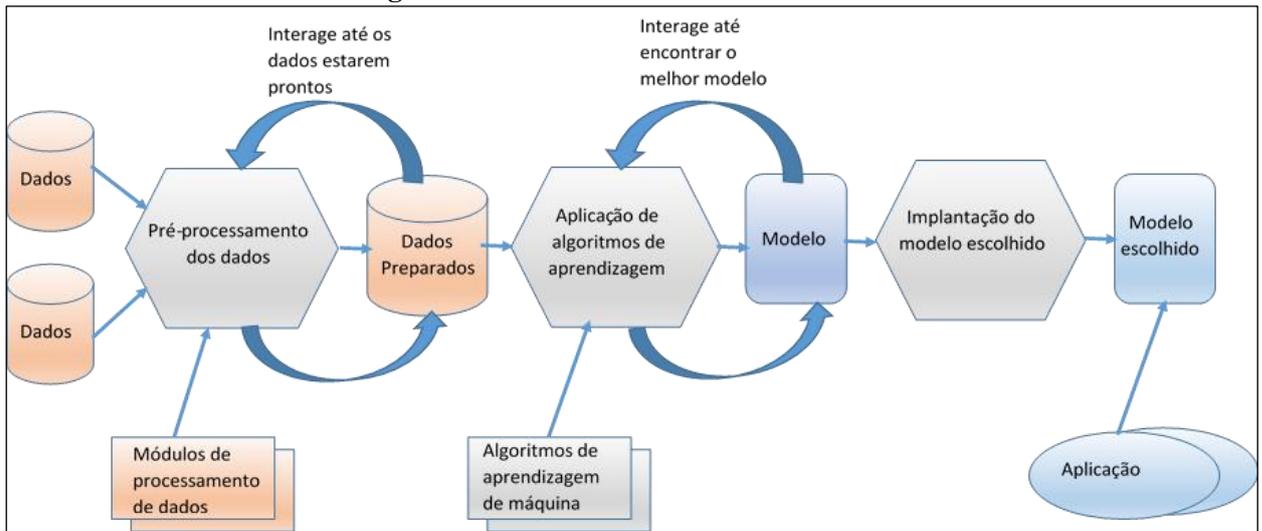
O processo de aprendizagem/treinamento (*learning*) no contexto de aprendizagem de máquina corresponde a um processo automático de busca pela melhor representação dos dados de entrada em determinada solução.

A aprendizagem de máquina pode ser empregada para resolver soluções de regressão, classificação e prognósticos. Portanto, soluções utilizando aprendizagem de máquinas estão sendo bastante utilizadas em sistemas de análise preditiva. (CHOLLET, 2018)

Resumidamente, aprendizagem de máquina é a busca por representações úteis dos dados de entrada, dentro de um espaço predefinido de possibilidades, usando a orientação de um sinal de feedback. Esta ideia simples permite resolver uma gama notavelmente ampla de tarefas intelectuais, desde o reconhecimento de voz até a condução autónoma de automóveis. (CHOLLET, 2018)

A Figura 9 apresenta o fluxo de tratativa dos dados para que o modelo de aprendizagem de máquina possa ser identificado. Como tudo começa com a análise dos dados, é muito importante que estes sejam os mais fidedignos possíveis e que retratem a realidade. Portanto, estes necessitam ser criteriosamente selecionados e pré-processados, através de módulos próprios para esta finalidade e que possam extrair os dados repetidos e conflitantes, entre outros. Esta etapa pode exigir uma grande carga de processamento e várias interações são realizadas para selecionar os dados mais representativos.

Uma vez que os dados estão prontos, eles são utilizados em algoritmos próprios de aprendizagem de máquina, tais como: regressão, árvore de decisão, assim como abordagens mais complexas que utilizam *two-class boosted decision tree* e *multiclass decision jungle*, entre outros. Com o objetivo de identificar o melhor modelo, os cientistas de dados, especialistas em aprendizagem de máquina, realizam testes e ajustes nos hiperparâmetros para identificar os algoritmos que podem modelar a solução da melhor forma. Então, o algoritmo será aplicado na etapa de testar o modelo. Caso o modelo apresente os valores das métricas que atendam aos limites desejados, ele será utilizado como o modelo escolhido; caso não os atinja, o modelo será refeito até que seja identificado um capaz de atingir a métrica desejada. O modelo após testado, identificado sua eficiência e aprovado, poderá ser compartilhado em diversas soluções.

Figura 9 – Fluxo de tratamento dos dados

Fonte: Adaptado de CHAPPELL, 2018

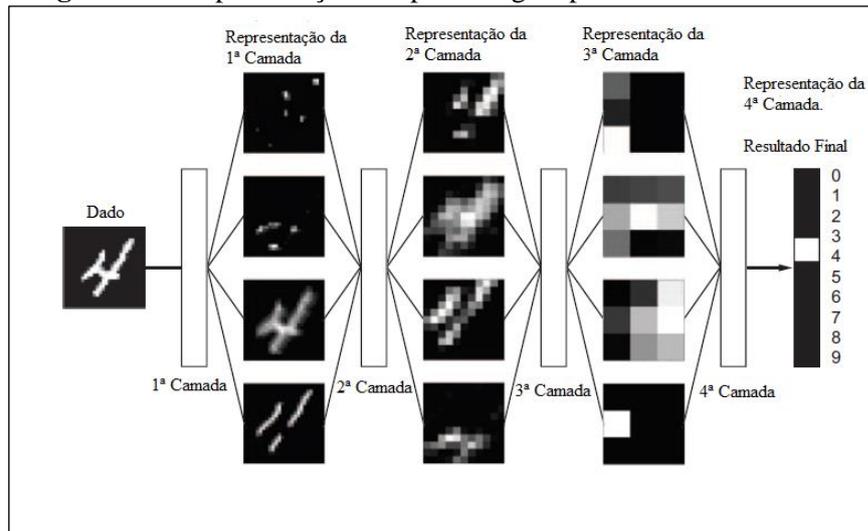
Diversos algoritmos de aprendizagem de máquina podem ser utilizados para classificação, tais como: [Logistic Regression](#), [K-nearest Neighbors \(KNN\)](#), [Support Vector Machines \(SVM\)](#), [MLP Neural Networks](#), [Multiclass Decision Jungle](#), [Two-Class Boosted Decision Tree](#), entre outros. Para utilizar como regressão, outros algoritmos podem ser utilizados: [Bayesian Linear Regression](#), [Boosted Decision Tree Regression](#), [Neural Network Regression](#), [Linear Regression](#), [Gaussian Process Regression](#). (SOBIE, FREITAS e NICOLAI, 2017)

2.5. APRENDIZAGEM PROFUNDA

A aprendizagem profunda é um subcampo da aprendizagem de máquinas. É uma abordagem das representações de aprendizagem (*learning representations*) a partir de dados que são enfatizados na aprendizagem de sucessivas camadas de representações cada vez mais significativas. O termo “aprofundamento” da aprendizagem profunda não é uma referência a qualquer tipo de compreensão mais profunda alcançada pela abordagem. Pelo contrário, representa a ideia de sucessivas camadas de representações. A quantidade de camadas que contribuem para um modelo é chamada de profundidade do modelo. Outras abordagens à aprendizagem de máquinas tendem a concentrar-se em aprender com uma ou duas camadas de representações dos dados. Por isso, às vezes, são chamados de aprendizagem rasa (*shallow learning*). (CHOLLET, 2018)

A aprendizagem profunda pode ser imaginada como um processo de destilação da informação em múltiplos estágios, em que a informação passa por filtros sucessivos e sai cada vez mais purificada, conforme pode ser representado através da Figura 10.

Figura 10 – Representação de aprendizagem profunda e camadas



Fonte: Adaptado de CHOLLET, 2018

Analisando mais especificadamente a camada e como processa os dados de entrada, tem-se que a parametrização de uma camada é armazenada nos pesos das respectivas camadas, que são por vezes chamados de parâmetros de uma camada. Portanto, o processo de aprendizagem significa encontrar um conjunto de valores para os pesos de todos os níveis da camada de uma rede, de tal forma que a rede possa identificar os melhores valores para mapear corretamente os dados de entrada com as respostas esperadas. Uma rede neural profunda pode conter dezenas de milhões de parâmetros. Encontrar o valor correto para todos eles é uma tarefa árdua, principalmente que a modificação do valor de um parâmetro poderá afetar o comportamento de todos os outros e o resultado da rede. (CHOLLET, 2018)

Para controlar os valores dos parâmetros e as alterações a serem realizadas, faz-se necessário observar o comportamento da rede, através de resultados que a rede pode apresentar. Uma das formas de analisar o resultado da rede é através da função *loss* que calcula o quão longe a resposta está do valor real e calcula uma pontuação de distância, capturando quão bem a rede se saiu neste caso específico. (CHOLLET, 2018)

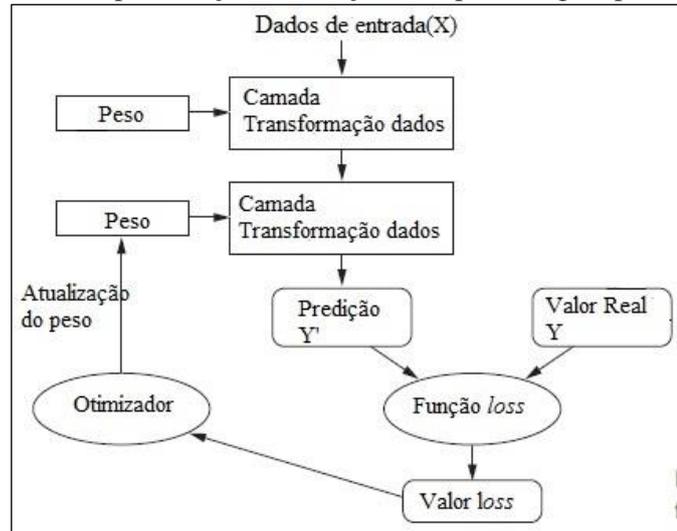
Uma das principais funções implementadas na aprendizagem profunda é a possibilidade de utilizar a pontuação da distância entre o valor real e a predição como um sinal de *feedback* para ajustar o valor dos pesos, em uma direção que irá baixar o valor do *loss*. Este ajuste é tarefa do otimizador, que implementa o algoritmo de retropropagação de erro (*backpropagation*) que é o algoritmo central na aprendizagem profunda. (CHOLLET, 2018)

Inicialmente, os pesos das camadas de rede recebem valores aleatórios, de modo que o diagrama de rede apenas implementa uma série de transformações aleatórias. Por conseguinte, o resultado pode estar longe do que deveria ser e o valor de *loss* pode ser muito alto, mas

conforme a quantidade de dados que são apresentados, os pesos são ajustados e o valor de *loss* diminui. O processo descrito acima é definido como *loop* de treino que realiza repetições em quantidade suficientes até atingir um número de *loss* satisfatório, representando saídas que estão o mais próximo possível dos valores reais, gerando consequentemente uma rede treinada.

A Figura 11 representa simbolicamente as principais funções utilizadas em aprendizagem profunda, conforme apresentado nos parágrafos acima.

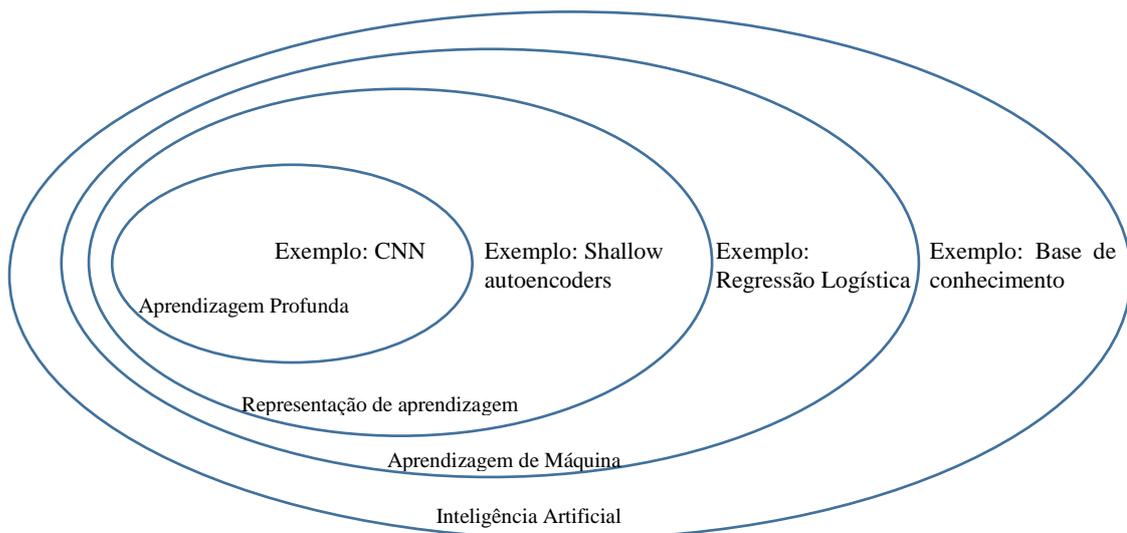
Figura 11 – Representação das funções de aprendizagem profunda



Fonte: Adaptado de CHOLLET, 2018

A Figura 12 apresenta o Diagrama de Venn e mostra a relação entre as modelos de Inteligência Artificial.

Figura 12 – Representação entre os modelos de Inteligência Artificial



Fonte: Adaptado de BENGIO, 2009

2.5.1. Accuracy, Precision, Recall e F1-Score⁴

Como forma de identificar o desempenho do modelo, são utilizadas métricas para comparar a eficácia entre os modelos testados. Para o cálculo das métricas são utilizadas as classificações binárias geradas pelo próprio modelo.

Uma classificação binária pode ser realizada em quatro categorias: falso negativo (FN), falso positivo (FP), verdadeiro negativo (VN) e verdadeiro positivo (VP). Um falso negativo é uma predição de falso quando deveria ter sido verdadeira. No contrário, falso positivo é quando uma predição de verdade deveria ter sido falsa. Verdadeiros positivos e verdadeiros negativos ocorrem quando um modelo está correto com sua predição. (DAVIS e GOADRICH, 2006)

Os falsos positivos (no caso de detecções de falhas) manifestam-se como sendo de baixa precisão e são indesejáveis, pois resultam numa manutenção indevida e/ou geram uma diminuição da confiança dos operadores humanos no sistema. (SOBIE, FREITAS e NICOLAI, 2017)

As quatro categorias para um problema de classificação de duas classes são frequentemente apresentadas numa matriz de confusão, conforme a Tabela 3.

Tabela 3 - Matriz de confusão

	Predição Positiva	Predição Negativa
Atual Positivo	VP	FN
Atual Negativo	FP	VN

Fonte: O autor, 2020

Accuracy é uma métrica importante da eficácia do algoritmo, no entanto, outras métricas adicionais para classificação binária, como *precision*, *recall* e *F1-Score*, são igualmente importantes, particularmente no domínio do monitoramento de condições que, neste trabalho, terá como objetivo realizar a classificação da falha.

Accuracy é uma métrica de avaliação que mostra quantas previsões de um modelo preditivo estão corretas (Goodfellow, Bengio e Courville, 2016). É definida como a razão entre os verdadeiros (positivos e negativos) e o total de amostras, conforme apresentada na equação 2.12.

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN} \quad \text{Eq. 2.12}$$

⁴ Os termos *accuracy*, *precision*, *F1-Score* e *Recall* não serão traduzidos para português em função da interpretação ambígua e ter significado específico na área de Inteligência Artificial, conforme definição nessa dissertação.

Precision é a fração de detecções reportada pelo modelo que estão corretas (Goodfellow, Bengio e Courville, 2016). É uma métrica importante para determinar a classificação de FP que quanto maior, menor será o valor calculado da métrica *precision*, conforme apresentada na equação 2.13.

$$Precision = \frac{VP}{VP + FP} = \frac{VP}{Total\ de\ Predição\ de\ Positivos} \quad \text{Eq. 2.13}$$

Recall é a fração de eventos verdadeiros (Verdadeiro Positivo) que foram detectados pelo modelo (Goodfellow, Bengio e Courville, 2016), conforme apresentada na equação 2.14.

$$Recall = \frac{VP}{VP + FN} = \frac{VP}{Total\ Atual\ Positivo} \quad \text{Eq. 2.14}$$

F1-Score é uma métrica que visa analisar o equilíbrio entre as métricas *precision* e *recall*, combinando-as para gerar um único indicador, conforme apresentada na equação 2.15.

$$F1 - Score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad \text{Eq. 2.15}$$

Recall, *precision* e *accuracy* não devem ser analisados separadamente, ao invés disso, eles devem ser considerados em conjunto na avaliação do modelo de destino. Um valor elevado de *precision* significa que todas as variáveis retornadas foram corretamente classificadas, mas é possível que se tenham perdido variáveis relevantes; enquanto que um resultado elevado de *recall* significa que todas as variáveis relevantes foram encontradas, mas pode haver uma abundância de resultados inúteis. *Accuracy* é a exatidão geral do modelo sendo considerado, que leva em conta não apenas as classificações verdadeiras positivas, mas também os valores verdadeiros negativos. (NASCIMENTO, 2012 p. 47)

2.5.2. Processo de aprendizagem

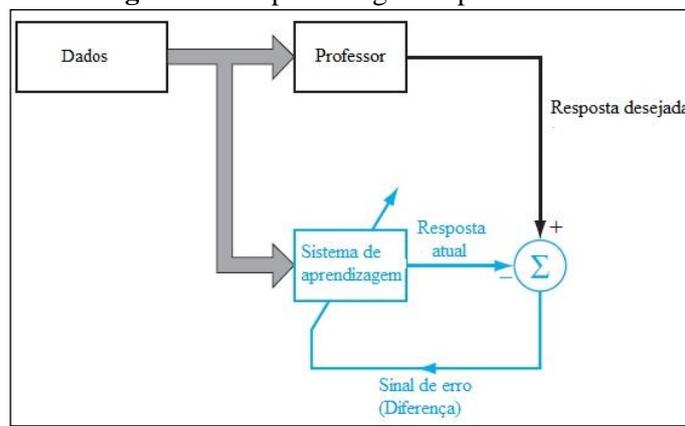
Assim como a forma de aprendizado do ser humano, as redes neurais também podem ser submetidas a processo similar. Em um sentido amplo, é possível categorizar o processo de aprendizagem das redes neurais através da aprendizagem com um professor (aprendizagem supervisionada) e aprender sem um professor (aprendizagem não supervisionada e aprendizagem com reforço).

a) Aprendizagem supervisionada

A forma de aprendizagem supervisionada é a base da aprendizagem de correção de erros. A Figura 13 apresenta um diagrama que ilustra o processo de aprendizagem supervisionada constituído por um sistema de retroalimentação (*feedback*) de ciclo fechado. Pode-se imaginar esse processo de aprendizagem como um professor que tem conhecimento do ambiente, sendo esse conhecimento representado por um conjunto de entrada-saída. O ambiente é, no entanto, desconhecido da rede neural. Em virtude do conhecimento incorporado, o professor é capaz de fornecer à rede neural uma resposta desejada para aquele conjunto de dados de treinamento. Os

parâmetros da rede são ajustados sob a influência combinada do conjunto de dados de treino e do sinal de erro. O erro é definido como a diferença entre a resposta desejada e a resposta real da rede. Este ajuste é realizado de forma iterativa com o objetivo de fazer com que a rede neural emule o professor. A identificação da assertividade do processo de aprendizagem ocorre quando o erro é mínimo e pode ser calculado através do Erro Quadrático Médio ou Soma dos Erros Quadráticos. Desta forma, o conhecimento do ambiente disponível para o professor é transferido para a rede neural e armazenado sob a forma de pesos sinápticos (hiperparâmetros). (HAYKIN, 2009)

Figura 13 – Aprendizagem supervisionada



Fonte: Adaptado de HAYKIN, 2009

Resumidamente, a aprendizagem supervisionada consiste em aprender a mapear dados de entrada para dados de saída conhecidos. Geralmente, quase todas as soluções de aprendizagem profunda que estão no centro das atenções nestes dias pertencem a esta categoria, tais como reconhecimento de caracteres, reconhecimento de voz, classificação de imagens, tradução de idiomas, classificação de objetos. (CHOLLET, 2018)

b) Aprendizagem não supervisionada

Este ramo da aprendizagem de máquina consiste em encontrar transformações interessantes dos dados de entrada sem a ajuda de rótulos de saída, para fins de visualização de dados, compressão de dados ou para melhor compreender as correlações presentes nos dados em questão. Aprendizagem não-supervisionada muitas vezes é um passo necessário para entender melhor um conjunto de dados antes de tentar resolver um problema de aprendizagem supervisionada. A redução da dimensão e o agrupamento (*clustering*) são categorias bem conhecidas de aprendizagem não supervisionada. (CHOLLET, 2018)

2.5.3. Redes Neurais Convolucionais (Convolutional Neural Network - CNN)

É um tipo de rede neural para processamento de dados que tem uma topologia conhecida, em forma de tabelas. Exemplos incluem dados de séries temporais que são adquiridos em

intervalos regulares de tempo e apresenta-se no formato de 1D (uma dimensão) e dados de imagem que pode apresenta-se no formato 2D (duas dimensões). O nome Rede Neural Convolutacional indica que emprega uma operação matemática chamada convolução que é um tipo especializado de operação linear. Portanto, Rede Neural Convolutacional são redes neurais que usam a convolução no lugar de multiplicação geral da matriz em pelo menos uma das suas camadas. (Goodfellow, Bengio e Courville, 2016)

Normalmente, a operação usada em uma Rede Neural Convolutacional não corresponde exatamente a definição de convolução usada em outros campos, como engenharia ou matemática pura. (Goodfellow, Bengio e Courville, 2016)

As equações de convolução são expressas através das equações 2.16 e 2.17.

$$s(t) = \int_{a=-\infty}^{\infty} x(a)w(t-a)da \quad \text{Eq. 2.16}$$

$$s(t) = (x * w)(t) \quad \text{Eq. 2.17}$$

Onde: w representa os pesos (*kernel*), x representa o valor de entrada, t representa o tempo, a função $s(t)$ é referenciada como *feature map*.

Analisando a equação 2.16, w precisa ser zero para todos os argumentos negativos a fim de evitar que a equação analise tempo no futuro, o que não é o objetivo da CNN. (Goodfellow, Bengio e Courville, 2016)

Em aplicações de aprendizagem de máquinas, a entrada (x) é uma matriz de dados multidimensional e o *kernel* (w) é um conjunto multidimensional de parâmetros que são adaptados e aprendidos pelo algoritmo de aprendizagem. Cada elemento de entrada (x) e de pesos (w) deve ser explicitamente armazenado separadamente, assume-se que estas funções são zero, exceto ao conjunto finito de pontos para os valores armazenados. Resulta que no algoritmo, pode-se implementar a soma infinita como um somatório sobre um número finito de elementos da matriz.

Como a função de convolução é cumulativa e pode ser utilizada em mais de um eixo cartesiano de cada vez, como é o caso do processamento de imagens em 2D, a equação 2.18 descreve a equivalência e melhor representa a função de convolução em aplicações de aprendizagem de máquina. (Goodfellow, Bengio e Courville, 2016)

$$S_{(i,j)} = (x * w)_{(i,j)} = \sum_m \sum_n x(i-m, j-n)w(m, n) \quad \text{Eq. 2.18}$$

Convolução aproveita duas ideias importantes que ajudam a melhorar o sistema de aprendizagem da máquina:

i) Interações esparsas – as camadas de rede neural tradicional realizam a multiplicação matricial por uma matriz de parâmetros com um parâmetro separado que descreve a interação entre cada unidade de entrada e saída. Isto significa que cada unidade de saída interage com cada unidade de entrada. As redes convolucionais, no entanto, têm interações esparsas (também chamadas de conectividade esparsa ou pesos esparsos), tornando o *kernel* menor do que a entrada. Isto significa que é necessário armazenar menos parâmetros, o que reduz os requisitos de memória do modelo e melhora a sua eficiência estatística. Significa também que o cálculo da saída requer menos operações. (Goodfellow, Bengio e Courville, 2016)

ii) Compartilhamento de parâmetros – Em uma rede neural tradicional, cada elemento da matriz de peso é usado uma única vez quando se calcula a saída de uma camada. Em uma Rede Neural Convolucional, cada membro do *kernel* é usado em cada posição da entrada. A partilha de parâmetros utilizada pela operação de convolução significa que em vez de aprender um conjunto separado de parâmetros para cada local, aprende-se um conjunto de parâmetros para o modelo. Consequentemente, reduz-se os requisitos de armazenamento dos parâmetros do modelo. (Goodfellow, Bengio e Courville, 2016)

CNN é composta por uma sequência de camadas, sendo que cada uma tem função específica para o tratamento e propagação dos dados. As principais camadas são: camada de entrada, camadas de filtros (convolução e *pooling*) e uma camada de classificação (totalmente conectada). A camada de filtros tem como função extrair os padrões (*features*) da camada de entrada ou anterior e a camada de classificação, como o próprio nome diz, tem a função de realizar a classificação dos dados, é do tipo [*Multilayer Perceptron \(MLP\)*](#) e é composta pela camada totalmente conectada.

Uma camada típica de uma rede convolucional consiste em três fases. Na primeira fase, a camada executa o processo de convolução em paralelo para produzir um conjunto de ativações lineares. Na segunda fase, cada ativação linear é executada através de uma função de ativação não linear, tal como a função ReLU (*Rectified Linear*). Na terceira fase, utiliza-se uma função *pooling* para modificar a saída para a próxima camada. (Goodfellow, Bengio e Courville, 2016)

As camadas e funções que compõem uma Rede Neural Convolucional são:

a) Camada de convolução

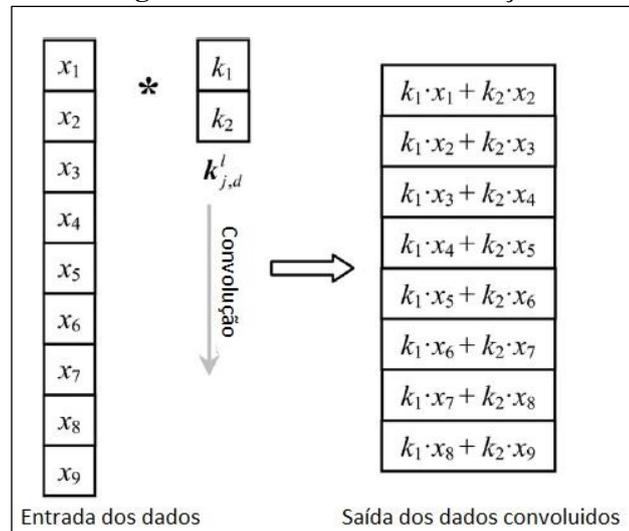
Embora o processo de convolução tenha sido explicado acima, neste tópico será explorado a camada de convolução e seu funcionamento.

Essa camada é responsável por realizar a convolução dos dados de entrada, utilizando os filtros, e encaminhá-los para a camada de ativação com o objetivo de gerar as características. (ZHANG *et al.*, 2017)

Os atributos dos dados de entrada são extraídos através dessa camada.

A Figura 14 apresenta o processo de convolução.

Figura 14 – Processo de convolução



Fonte: Adaptado de Jia e colaboradores

A expressão matemática utilizada nessa camada foi representada através da equação 2.18.

b) Camada *Pooling*

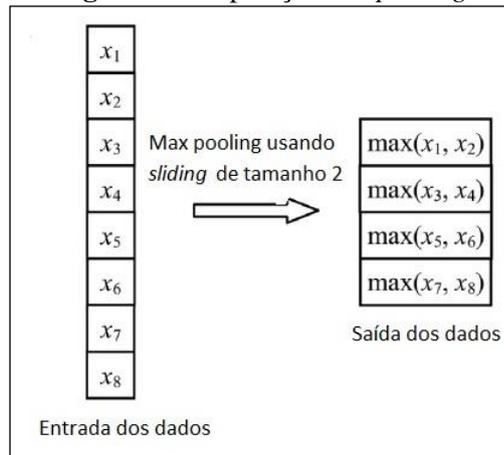
É configurada para atuar após a camada de convolução.

A função mais comumente empregada nesta camada é a *maxpooling* com o objetivo de reduzir a quantidade de parâmetros da CNN, reduzir a dimensão dos dados resultante da camada de convolução e acelerar o processo de computação. O tamanho do passo de deslizamento (*sliding*) é configurado na função para percorrer toda extensão do volume do dado e quanto maior, maior será a incerteza de identificação do dado. (JIA *et al.*, 2018)

A função *pooling* sintetiza as respostas em torno do tamanho definido pela janela e é possível utilizar menos camadas de *pooling* do que camadas de convolução. A camada *pooling* é capaz de agrupar regiões espaçadas entre k pontos em vez de 1 ponto de diferença. Isto melhora a eficiência computacional da rede porque a camada seguinte tem aproximadamente k vezes menos entradas para processar. Esta redução no tamanho da entrada, na camada seguinte, resulta em maior eficiência estatística e requisitos reduzidos de memória para o armazenamento dos parâmetros. (Goodfellow, Bengio e Courville, 2016)

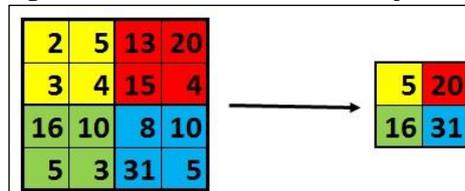
A forma mais comum da camada *pooling* consiste em substituir os valores de uma região por seu valor máximo (*maxpooling*), conforme Figura 15 e 16. A operação apresentada através da Figura 16, representa uma redução do volume de entrada de dados 4×4 para 2×2 , em virtude de utilizar o passo de deslizamento de 2.

Figura 15 – Operação *maxpooling*



Fonte: Adaptado de (Jia e colaboradores)

Figura 16 – Processo da camada *pooling*



Fonte: O autor, 2020

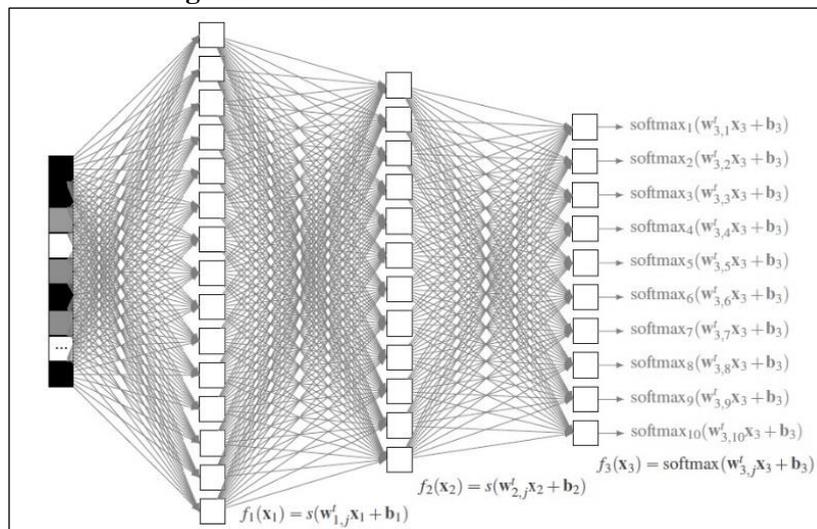
c) Camada Totalmente Conectada (*Fully Connected*)

É configurada após as camadas de convolução e *pooling*, sendo a última camada da rede.

Os dados de saída da camada *pooling* são encaminhados para esta camada com o objetivo de classificar os dados e propagar o sinal. Essas camadas são similares a uma Rede Neural Artificial (*Artificial Neural Network– ANN*) convencional que utiliza funções de ativação do tipo *softmax* na última camada de saída.

A Figura 17 representa as camadas de uma rede de aprendizagem profunda e a última camada representa a camada totalmente conectada.

Figura 17 – Camada Totalmente Conectada



Fonte: Vieira, Razente e Barioni, 2017

d) Normalização de batches (*Batch Normalization*)

É responsável por reduzir a covariância e acelerar o processo de treinamento.

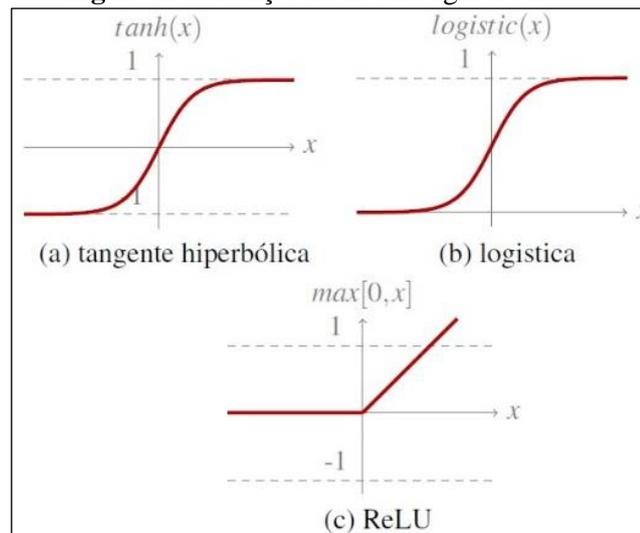
Tem efeito regularizador, normalizando as ativações da camada anterior em cada batch de entrada, mantendo a ativação média próxima a zero (centralizada) e o desvio padrão das ativações próximo a 1. (Vieira, Razente e Barioni, 2017)

e) Função de ativação da camada de convolução

É responsável por encontrar uma função não-linear que represente o sinal de entrada, normalizar o peso (k) e acelerar o processo de convergência do modelo. (Krizhevsky, Sutskever e Hinton (2012) e Zhang e colaboradores (2017))

A função ReLU tem sido mais empregada porque as funções sigmoidais saturam a partir de um determinado ponto, enquanto a ReLU é uma função identidade para valores positivos. As funções sigmoidais comprimem a saída para um intervalo curto e a função ReLU cancela todos os valores negativos, sendo linear para os positivos, conforme apresentado na Figura 18.

Figura 18 – Função ReLU X Sigmoidais



Fonte: Vieira, Razente e Barioni

A equação 2.19 descreve a função ReLU.

$$a^{l(i,j)} = f(z^{l(i,j)}) = \max\{0, z^{l(i,j)}\} \quad \text{Eq. 2.19}$$

Em que:

$z^{l(i,j)}$ é o valor de saída da camada de normalização por batelada

$a^{l(i,j)}$ é a ativação de $z^{l(i,j)}$

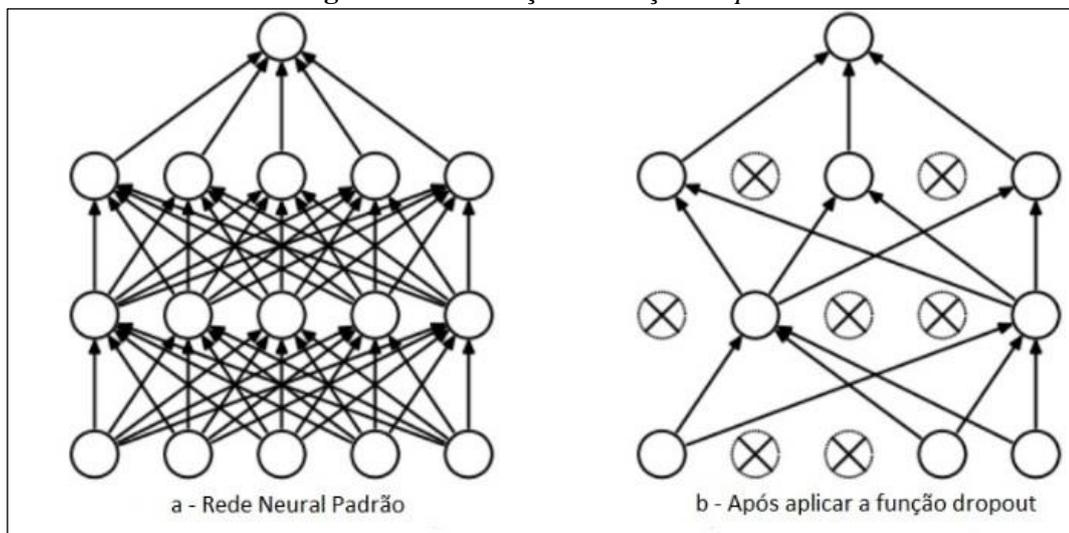
f) Dropout

É uma função para minimizar *overfitting*, na fase de treinamento, em que aleatoriamente desativa com probabilidade p (parâmetro ajustável) a ativação de neurônios. (Vieira, Razente e Barioni, 2017)

Dropout momentaneamente (em um lote de dados de entrada) desliga parte dos neurônios em uma camada para que eles não contribuam com qualquer informação ou aprendam qualquer informação durante essas atualizações, e o ônus recai sobre outros neurônios ativos para que possam aprender e reduzir o erro. O efeito é que a rede se torna menos sensível aos pesos específicos dos neurônios. Isso, por sua vez, resulta em uma rede que é capaz de melhorar a generalização e é menos provável que se sobreponha aos dados de treinamento. (BROWNLEE, 2016)

As Figuras 19a e 19b ilustram, respectivamente, uma rede neural sem a função de *dropout* e com a função *dropout*, com um ajuste de 0,5, ou seja, probabilidade de 50%.

Figura 19 – Ilustração da função *dropout*



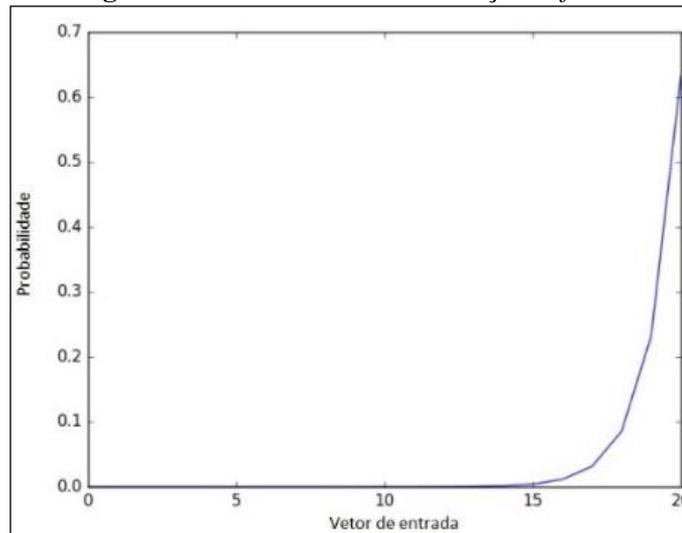
Fonte: Adaptado (JERKIC, 2017)

g) Função *softmax*

É a função de ativação utilizada na Camada Totalmente Conectada para que a saída possa ser interpretada como probabilidades e permite atribuir um exemplo a classe com maior probabilidade. (Vieira, Razente e Barioni, 2017)

A função *Softmax* calcula a distribuição de probabilidades do evento sobre 'n' eventos diferentes. A principal vantagem do uso da *Softmax* é o intervalo de probabilidades de saída que será de 0 a 1 e a soma de todas as probabilidades será igual a um.

A Figura 20 apresenta a propriedade fundamental da função *Softmax*. O valor com maior probabilidade será o resultado da função *Softmax*.

Figura 20 – Resultado de uma função *Softmax*

Fonte: Adaptado (POLAMURI, 2017)

h) Função custo (*loss*)

É uma métrica para avaliar quão bem o algoritmo realiza a predição (\hat{y}).

Existem diferentes tipos de funções que podem ser utilizadas de acordo com a aplicação, conforme Tabela 4.

Tabela 4 – Função *Loss*

Aplicação	Função
Regressão	<i>Mean Squared Error</i>
	<i>Mean Squared Logarithmic Error</i>
	<i>Mean Absolute Error</i>
Classificação Binária	<i>Binary Cross-Entropy</i>
	<i>Hinge Loss</i>
	<i>Squared Hinge Loss</i>
Classificação Multi Classes	<i>Multi-Class Cross-Entropy</i>
	<i>Sparse Multiclass Cross-Entropy</i>
	<i>Kullback Leibler Divergence</i>

Fonte: O autor, 2020

A Equação 2.20 representa a função *cross-entropy*, considerando um único exemplo cuja distribuição de probabilidade de classes real é y e a predição é dada por $f(x) = \hat{y}$. (Vieira, Razente e Barioni, 2017)

$$l^{ce} = - \sum_j y_j * \log(\hat{y}_j + \varepsilon) \quad \text{Eq. 2.20}$$

Onde: y_j representa a classe real; \hat{y}_j representa a predição; $\varepsilon \ll 1$, é uma variável para evitar $\log(0)$.

Caso as previsões desviem muito dos resultados reais (VP ou VN), a função poderá calcular um custo maior e não apresentar uma convergência. Caso contrário, quando as previsões estiverem identificando corretamente os resultados reais, a função poderá calcular um custo menor e apresentar uma convergência. O resultado da função *cross-entropy* é minimizada, quando valores menores representam um modelo em que a previsão está próxima do valor real. Um modelo que prevê previsão perfeita ($y=\hat{y}$) tem uma *cross-entropy* de 0,0. (Vieira, Razente e Barioni, 2017)

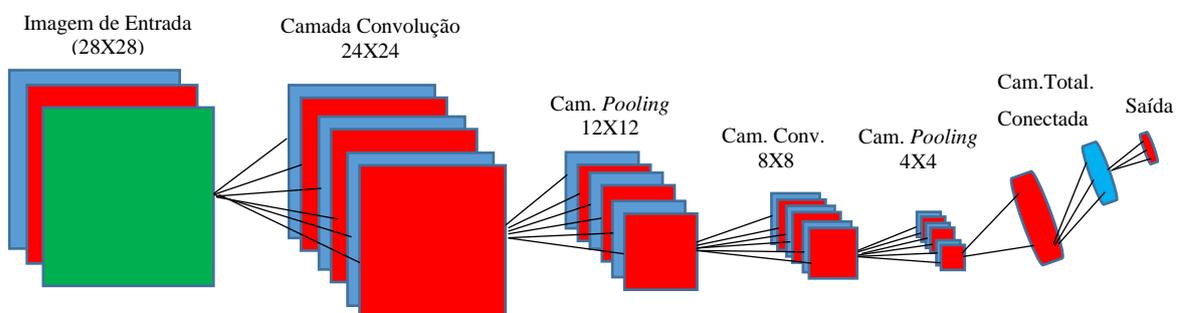
Objetivando reduzir o custo (*loss*) do modelo, pode-se utilizar alguns otimizadores da função, dentre eles destaca-se o [gradiente descendente](#) (GD) que basicamente computa derivadas parciais de forma a encontrar, para cada parâmetro do modelo, qual modificação dos parâmetros permite minimizar a função. (Goodfellow, Bengio e Courville, 2016)

Em função dos itens apresentado acima, pode-se resumir o resultado de uma Rede Neural Convolutiva como uma informação da probabilidade dos dados de entrada pertencerem a uma classe na qual a rede foi treinada.

A Figura 21 ilustra um modelo CNN-2D com uma camada de entrada que aceita imagens de 28 x 28 pontos. Cada camada de convolução, após a camada de entrada, alterna-se com as camadas *pooling*.

A camada de entrada é uma camada passiva que aceita uma imagem de entrada e atribui seus canais de cor (R, G e B) como os mapas de recursos de seus três neurônios. Com a propagação para a frente sobre um número suficiente de camadas de *pooling*, eles são detalhados para um escalar (1-D) na saída da última camada. A camada totalmente conectada é do tipo [Multilayer Perceptron \(MLP\)](#) e *feedforward* que tem a camada de saída estimando a classificação.

Figura 21 – Visão geral de uma amostra de CNN



Fonte: Adaptado de Ince e colaboradores

3. TRABALHOS RELACIONADOS

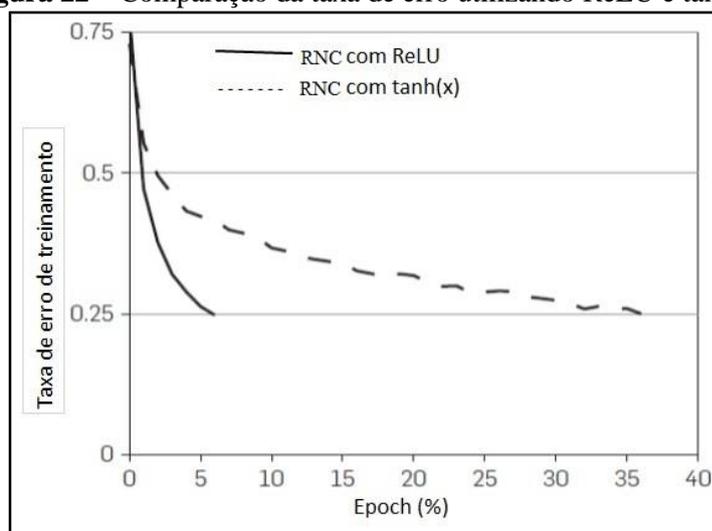
Nesse capítulo são analisadas e comentadas as referências teóricas que servirão de base de conhecimento para o desenvolvimento, estruturação e elaboração da pesquisa. Todas as referências apresentam um breve resumo, os objetivos e a correlação com o tema pesquisado.

Na área acadêmica, existem diversos pesquisadores desenvolvendo soluções com o uso de modelos de Inteligência Artificial para prognóstico, diagnóstico e classificação de falhas de ativos através de sinal de vibração. Atualmente, com o desenvolvimento de soluções utilizando aprendizagem profunda, autores estão utilizando a CNN para diagnosticar e classificar as falhas de ativos, conforme apresentados a seguir.

3.1. CLASSIFICAÇÃO DE IMAGENS

Krizhevsky, Sutskever e Hinton (2012) foram os primeiros pesquisadores que utilizaram CNN para classificação de imagens 2-D em uma competição utilizando a base de dados *ImageNet Large-Scale Visual Recognition Challenge* (ILSVRC). Essa base de dados, em 2012, possuía 15 milhões de imagens rotuladas e cadastradas em 22 000 categorias diferentes. O concurso realiza a classificação dos modelos através da incerteza na classificação das imagens em top-1 e top-5. A taxa de erro em top-5 é a fração das imagens de teste para a qual a etiqueta correta não está entre as cinco etiquetas consideradas mais prováveis pelo modelo. Os autores utilizaram a função *dropout* e a função *Rectified Linear Unit* (ReLU), até então pouco conhecida e utilizada, segundo os pesquisadores. A função ReLU tem a capacidade de ser mais rápida, se comparada com a função *tanh(x)*, conforme apresentado na Figura 22. A linha sólida representa o uso da ReLU em CNN que atinge uma taxa de erro de 25%, ou seja, aproximadamente seis vezes mais rápido que uma rede equivalente utilizando a função *tanh(x)*. A rede foi instalada em duas placas gráficas (GPU), em paralelo, e demorou entre 5 e 6 dias para realizar a etapa de treinamento. Embora os pesquisadores não tenham participado do concurso em 2010, eles realizaram um comparativo do modelo desenvolvido como o melhor resultado apresentado. A taxa de erro para o top-1 e top-5 foram, respectivamente, 37,5% e 17,0%, enquanto o melhor resultado e que ganhou o concurso foi 47,1% e 28,2%, respectivamente, ou seja, o modelo CNN, desenvolvido pelos pesquisadores, apresentou uma taxa de erro inferior ao melhor modelo apresentado, constatando a superioridade do modelo CNN.

Através deste trabalho, sendo um dos primeiros a ser publicado utilizando CNN para classificação de imagens, foi possível demonstrar a possibilidade dos modelos utilizando CNN em classificar imagens com taxa de erro inferior aos trabalhos apresentados.

Figura 22 – Comparação da taxa de erro utilizando ReLU e tanh(x)

Fonte: Adaptado de Krizhevsky, Sutskever e Hinton (2012)

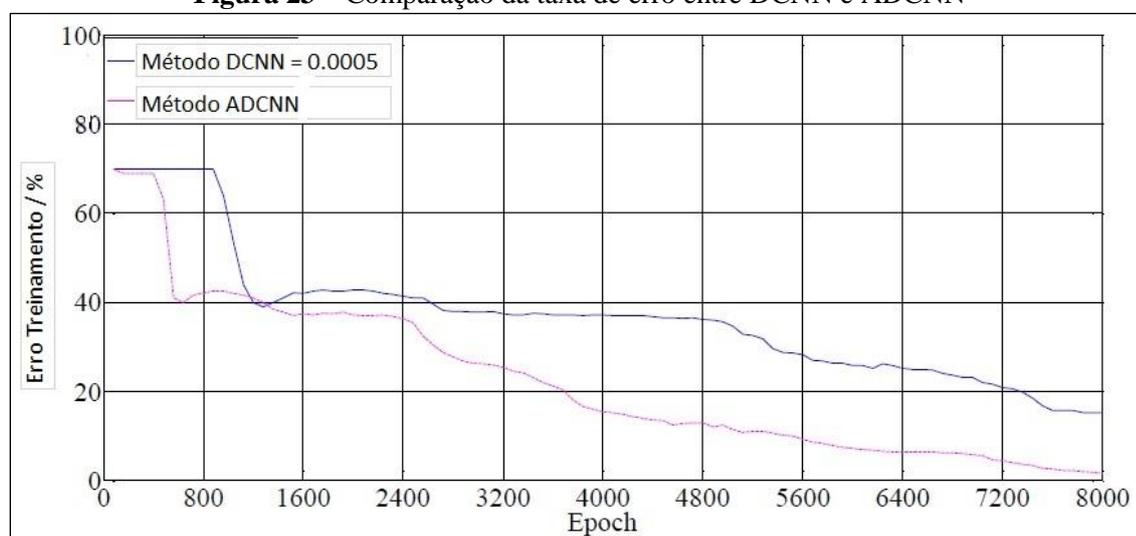
3.2. CLASSIFICAÇÃO DE FALHA

No trabalho apresentado por Janssens e colaboradores (2016) realizou-se um estudo que consiste no desenvolvimento de uma solução que possa aprender, sem intervenção humana, as características dos dados brutos de falha dos ativos. Durante o experimento, falhas de rolamentos foram analisadas, tais como falhas em pistas externas e degradação da lubrificação, mancais de rolamento não saudáveis, além de desbalanceamento do rotor. Para verificar a generalização do algoritmo, foram realizados testes em diferentes tipos de rolamentos e condições de falha. Ao final do artigo, dois sistemas são comparados, utilizando os mesmos dados, e observa-se que a *accuracy* obtida pelo sistema desenvolvido pelo autor, utilizando [feature-learning](#) e CNN, em detrimento de outro sistema que utiliza [feature-learning](#) e o classificador [Random Forest](#), atingiram o valor de 93,61% e 87,25%, respectivamente.

No trabalho apresentado por Guo, Chen e Shen (2016) foi proposto o uso da CNN para diagnosticar falhas nos rolamentos e determinar a gravidade. Através de estudos realizados, os autores concluíram que os problemas na extração das características ([features](#)) automaticamente, sem aumentar significativamente a demanda por conhecimento dos ativos, e na maximização da *accuracy*, permaneciam sem solução. Concluíram também que poucas pesquisas têm usado CNN para diagnóstico de falhas. Portanto, propuseram o modelo *Adaptive Convolutional Neural Network* (ADCNN) para classificar falhas de rolamento e determinar a sua gravidade, com a habilidade de extrair as principais características dos dados de forma automática, sem intervenção manual. Para o desenvolvimento do modelo, foi acrescentada uma taxa de aprendizado adaptativa com o objetivo de extrair as características mais expressivas e obter o melhor resultado de reconhecimento. Segundo os autores, a taxa de

aprendizado adaptativa é capaz de prevenir o [gradiente vanishing](#). Os experimentos foram investigados utilizando a base de dados pública *Case Western Reserve University* (CWRU), gerados através de testes realizados na bancada simulando falhas de rolamento, tais como: pista interna, pista externa e falha de esferas. Foram realizados testes comparando o modelo ADCNN com DCNN e SVRM. A primeira comparação foi realizada em função da velocidade de convergência e da taxa de erro, onde se comparam os modelos ADCNN e DCNN, constatando, conforme a Figura 23, que o modelo ADCNN encontra uma convergência mais rápida que CNN durante as primeiras 1000 [epochs](#) com uma validação de erro de 10% menor. Realizando a comparação entre o modelo ADCNN e SVRM, os autores identificaram um desempenho no modelo ADCNN 10% melhor que no modelo SVRM.

Figura 23 – Comparação da taxa de erro entre DCNN e ADCNN



Fonte: Guo, Chen e Shen (2016)

No trabalho apresentado por Zhang e colaboradores (2017), o modelo desenvolvido foi denominado de *Deep Convolutional Neural Networks with First-layer Kernels* (WDCNN) e, segundo os autores, é capaz de: utilizar os sinais de vibração, no domínio do tempo, sem realizar tratamento prévio, filtragem ou cálculos estatísticos; apresentar um bom domínio da capacidade de adaptação em diferentes cenários, utilizando o modelo AdaBN (*Adaptive Batch Normalization*); além de poder ser utilizado em ambientes com grande nível de ruído no sinal de vibração. Para o desenvolvimento da solução, os autores realizaram modificações no modelo CNN, ajustando os parâmetros dos filtros ([kernel](#)) da primeira camada de convolução com valores (64 X 1) e as demais camadas de convolução utilizaram valores mais baixos (3 X 1); e também implementou o modelo denominado AdaBN. Para comprovar a capacidade de adaptação do modelo, os autores acrescentaram sinal de ruído aos dados da base de dados utilizando a função [White Gaussian Noise](#). Então, foi possível comprovar a eficácia do modelo

em classificar a falha, mesmo que não tivesse sido treinado com o acréscimo de ruído aos dados. O modelo foi configurado com dez camadas de filtragem (convolução e *pooling*), uma camada totalmente conectada utilizando a função *softmax*. Com essas modificações foi possível atingir *accuracy* média de 95,9%, utilizando a base de dados CWRU.

No trabalho apresentado por Marins e colaboradores (2018), realizou-se um estudo comparativo do Modelo Baseado em Similaridade ([Similarity-Based Modeling - SBM](#)) utilizando as bases de dados MaFaulDa e CWRU. A *accuracy* obtida utilizando a base de dados MaFaulDa e CWRU foi de 98,5% e 98,9%, respectivamente.

Para o modelo [SBM](#) ser utilizado é necessário extrair as características estatísticas (média, desvio padrão, variância, entropia e curtose) dos dados, o que requer um trabalho extra na geração do modelo [SBM](#). Esse modelo foi testado na mesma base de dados (MaFaulDa) em que os testes deste trabalho foram realizados e, portanto, servirá de referência para buscar um modelo com *accuracy* superior a 98,5%.

No trabalho apresentado por Zhao e colaboradores (2018), foi fornecida uma revisão bibliográfica sobre os modelos baseados em aprendizagem profunda e utilizados em sistemas de monitoramento de ativos. O trabalho informa que o uso do sinal bruto em soluções utilizando aprendizagem profunda é uma nova área de diagnóstico em falhas de rolamentos, engrenagens, desalinhamento, desbalanceamento, em compressores, bombas hidráulicas e que podem ser orientados por [data driven](#). A grande vantagem de poder utilizar os dados brutos do sinal de vibração é que não necessita da conversão destes em características estatísticas, tais como desvio padrão, curtose e médias para então poder gerar o modelo.

Foi apresentada neste trabalho uma visão geral sobre modelos baseados em aprendizagem profunda em sistemas de monitoramento de ativos e seu impacto nas tecnologias de ponta (estado-da-arte). O trabalho realizou uma comparação entre os modelos de aprendizagem profunda ([Random Forest](#), [Neural Network](#), [Autoencoder](#), [Denoising Autoencoder](#) e CNN) e apresentou os erros de cada modelo, demonstrando que, dentre os modelos apresentados, CNN obteve o menor erro médio absoluto.

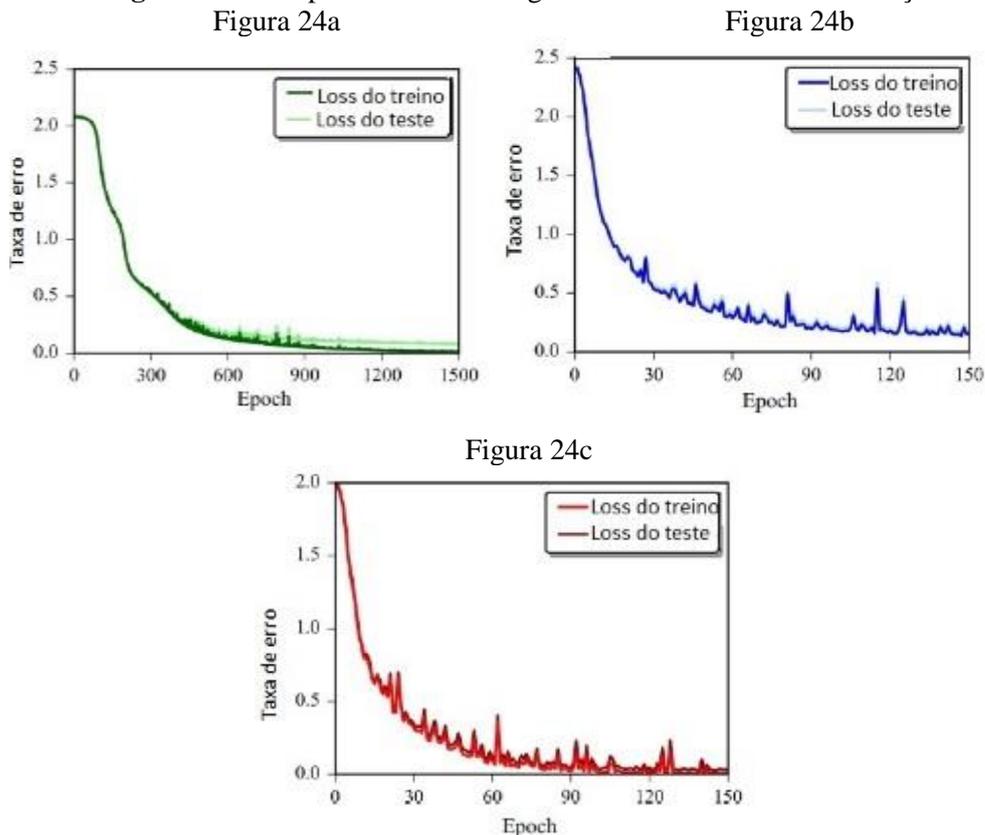
Os modelos e respectivos erros médios são apresentados na Tabela 5.

Tabela 5 - Comparação dos erros dos modelos de Inteligência Artificial

Modelo	Erro médio absoluto	Erro médio quadrático
<i>Linear SVR</i>	13,7	248,9
<i>Random Forest</i>	14,4 ± 0,2	289 ± 5,9
<i>Neural Network</i>	11,5 ± 1,2	191,6 ± 26,8
<i>Auto-encoder</i>	11,2 ± 1,9	185 ± 41,9
<i>DBN</i>	11,7 ± 1,3	194,6 ± 34,2
<i>CNN</i>	11,0 ± 1,3	197,2 ± 30,9

Fonte: adaptado Zhao e colaboradores

O trabalho apresentado por Jia e colaboradores (2018) teve como objetivo demonstrar um modelo CNN que os autores denominaram de *Deep Normalized Convolutional Neural Network* (DNCNN) e que apresentou boa *accuracy* mesmo quando utilizou poucos dados na fase de treinamento do modelo. O autor cita que esse experimento foi realizado em virtude da dificuldade em obter dados que representem falha do ativo, enquanto os dados que representam o funcionamento normal existem em maior quantidade. O trabalho apresenta detalhadamente o funcionamento das diferentes camadas do modelo CNN, assim como testes comparativos aplicando as funções [sigmoid](#), ReLU e a estratégia da normalização do peso (*weight normalization*), que segundo os pesquisadores tem a capacidade de melhorar a convergência do processo de otimização. O modelo utilizando a função de [sigmoid](#) apresentou uma convergência próxima de 1500 *epochs*; utilizando ReLU, foi próxima de 150 *epochs* e, no modelo desenvolvido (DNCNN), foi próxima de 120 *epochs*, conforme Figuras 24a, 24b e 24c, respectivamente.

Figura 24 - Comparativo de convergência utilizando diferentes funções

3.3. PRÉ-PROCESSAMENTO DOS DADOS

Diversas pesquisas estão sendo realizadas empregando aprendizagem de máquina para classificação de falha em ativos rotativos.

No trabalho apresentado por Ahmed e Nandi (2018) os autores desenvolveram um *framework* com o objetivo de classificar as falhas de rolamentos com uma quantidade limitada de dados, utilizando uma combinação de métodos para encontrar a melhor taxa de compressão dos dados (*Compressive Sampling (CS)*) baseado na taxa de amostragem de compressão, o extrator de características (*feature extraction*) e o classificador da falha. O framework tem como objetivo tratar os dados em diferente etapas. Na primeira etapa, utiliza duas representações de compressibilidades do sinal de vibração (FFT e transformada de Wavelet), com o objetivo de otimizar a quantidade de dados. Na segunda etapa, foram utilizadas técnicas com o objetivo de selecionar as principais características (*feature selection*) dos dados através das três técnicas baseadas em similaridades (Fisher Score (FS), Laplacian Score (LS), Relief-F), uma técnica de correlação (coeficiente de correlação Pearson (PCC)) e o método Chi-2. Na terceira etapa foram utilizados os classificadores (Regressão Logística (LRC), *Artificial Neural Network (ANN)* e SVM).

O método de *Compressive Samplig (CS)* baseia-se no fato de sinais terem representações dispersas em algum domínio e poderem ser convertidos em outro domínio com melhor representatividade. É o caso dos sinais de vibração que apresentam representação dispersa no domínio do tempo, podem ser transformados para o domínio da frequência e apresentar uma melhor representatividade. A CS baseia-se em dois conceitos: (1) a taxa de dispersão dos sinais e (2) a matriz de medição para ser utilizada na compressão dos sinais originais com base nas suas representações esparsas.

As tabelas 6, 7 e 8 apresentam as *accuracies* dos modelos quando os dados foram gerados utilizando uma base de dados do autor.

O parâmetro k representa o número de características selecionadas (*feature selection*) e o parâmetro α representa a taxa de amostragem (0,1; 0,2; 0,3 e 0,4) de *Compressive Samplig (CS)*. As *accuracies* com valor superior a 99% estão em negrito.

Tabela 6 – *Accuracy* na classificação de falha de rolamento utilizando o classificador LRC e diferentes combinações de algoritmos de compressão

Sparsifying method		FFT				WT			
Sampling rate (α)		0.1	0.2	0.3	0.4	0.1	0.2	0.3	0.4
<i>Classifier: LRC</i>									
Method	k								
CS-FS	60	94.3 ± 5.6	98.7 ± 0.7	98.8 ± 0.5	98.1 ± 0.9	92.2 ± 7.5	96.4 ± 3.9	97.2 ± 2.8	98.0 ± 2.2
	120	99.7 ± 0.4	99.8 ± 0.3	99.8 ± 0.2	99.9 ± 0.4	96.3 ± 3.8	96.2 ± 4.0	99.2 ± 0.8	99.9 ± 0.1
	180	99.9 ± 0.1	99.9 ± 0.2	99.9 ± 0.1	100.0 ± 0.0	98.9 ± 1.2	97.8 ± 2.2	99.7 ± 0.3	99.9 ± 0.2
CS-LS	60	95.8 ± 0.9	93.4 ± 1.2	95.2 ± 3.7	97.4 ± 2.8	91.9 ± 8.2	93.3 ± 6.9	95.1 ± 4.8	96.6 ± 3.5
	120	99.5 ± 0.3	99.8 ± 0.6	99.4 ± 0.6	99.9 ± 0.1	92.8 ± 7.2	94.2 ± 5.7	96.6 ± 3.4	98.0 ± 2.1
	180	99.8 ± 0.2	99.9 ± 0.1	99.9 ± 0.1	99.9 ± 0.1	95.4 ± 4.6	95.5 ± 4.4	96.5 ± 3.7	98.9 ± 1.1
CS-Relief-F	60	99.5 ± 0.3	99.3 ± 0.6	99.5 ± 0.2	99.4 ± 0.3	94.1 ± 5.1	93.7 ± 6.4	95.8 ± 3.4	96.6 ± 3.5
	120	99.8 ± 0.3	99.9 ± 0.2	99.9 ± 0.1	99.9 ± 0.1	96.4 ± 3.2	95.9 ± 4.2	98.3 ± 1.8	99.9 ± 0.1
	180	99.9 ± 0.1	99.9 ± 0.1	100.0 ± 0.0	100.0 ± 0.0	98.4 ± 1.6	99.3 ± 0.7	100.0 ± 0.0	100 ± 0.0
CS-PCC	60	98.5 ± 1.3	99.3 ± 0.5	98.8 ± 0.4	98.4 ± 0.9	94.1 ± 5.7	94.3 ± 4.6	97.4 ± 2.3	98.6 ± 1.5
	120	99.8 ± 0.3	99.5 ± 0.2	99.8 ± 0.2	99.9 ± 0.1	95.0 ± 4.9	95.0 ± 4.9	98.1 ± 1.7	98.9 ± 1.3
	180	99.9 ± 0.1	99.9 ± 0.2	99.9 ± 0.2	99.9 ± 0.1	98.8 ± 1.2	99.0 ± 1.1	98.8 ± 1.2	99.4 ± 0.6
CS-Chi-2	60	98.0 ± 2.2	98.7 ± 1.4	98.8 ± 1.2	98.8 ± 1.2	95.1 ± 4.5	96.7 ± 2.7	98.3 ± 2.8	98.9 ± 1.2
	120	99.5 ± 0.5	99.5 ± 0.4	99.5 ± 0.3	99.5 ± 0.5	96.8 ± 2.9	97.9 ± 3.2	96.6 ± 3.4	99.9 ± 0.1
	180	99.9 ± 0.1	99.9 ± 0.1	100.0 ± 0.0	99.9 ± 0.1	99.5 ± 0.4	99.3 ± 0.7	100.0 ± 0.0	100.0 ± 0.0

Tabela 7 – *Accuracy* na classificação de falha de rolamento utilizando o classificador ANN e diferentes combinações de algoritmos de compressão

Sparsifying method		FFT				WT			
Sampling rate (α)		0.1	0.2	0.3	0.4	0.1	0.2	0.3	0.4
<i>Classifier: ANN</i>									
Method	k								
CS-FS	60	93.8 ± 6.3	98.3 ± 0.8	99.0 ± 1.1	98.9 ± 0.4	91.6 ± 8.3	97.1 ± 2.8	98.9 ± 1.2	97.7 ± 2.3
	120	99.2 ± 1.3	99.7 ± 0.7	99.7 ± 0.6	99.6 ± 0.9	97.2 ± 2.8	98.3 ± 1.7	99.0 ± 0.9	99.3 ± 0.7
	180	99.8 ± 0.5	99.8 ± 0.3	99.7 ± 0.3	100.0 ± 0.0	98.2 ± 1.9	99.1 ± 0.9	99.3 ± 0.8	99.7 ± 0.3
CS-LS	60	98.6 ± 0.9	97.3 ± 2.1	97.9 ± 2.2	95.8 ± 3.3	92.9 ± 7.0	93.7 ± 6.3	94.4 ± 5.5	97.1 ± 2.7
	120	99.9 ± 0.2	99.4 ± 0.7	99.9 ± 0.1	99.9 ± 0.1	95.3 ± 4.8	95.2 ± 4.8	96.9 ± 3.2	98.3 ± 1.7
	180	99.9 ± 0.1	99.9 ± 0.2	99.9 ± 0.1	99.9 ± 0.1	96.8 ± 2.9	95.9 ± 4.1	97.8 ± 2.1	98.9 ± 1.0
CS-Relief-F	60	99.7 ± 0.6	99.7 ± 0.3	99.2 ± 0.9	99.6 ± 0.3	94.6 ± 5.4	94.4 ± 5.7	93.2 ± 6.7	97.1 ± 2.7
	120	99.9 ± 0.2	99.7 ± 0.4	99.6 ± 0.9	99.7 ± 0.4	97.2 ± 2.8	97.0 ± 3.1	99.5 ± 0.5	99.8 ± 0.2
	180	99.9 ± 0.3	100 ± 0.0	100.0 ± 0.0	99.9 ± 0.3	99.1 ± 0.9	99.9 ± 0.1	100.0 ± 0.0	100.0 ± 0.0
CS-PCC	60	99.4 ± 0.3	99.4 ± 0.6	99.2 ± 0.9	99.5 ± 0.5	94.9 ± 5.3	95.7 ± 4.3	98.5 ± 1.6	94.4 ± 5.5
	120	99.8 ± 0.3	99.7 ± 0.4	99.9 ± 0.3	99.9 ± 0.1	96.4 ± 3.6	96.3 ± 3.8	99.0 ± 0.9	98.5 ± 1.4
	180	99.7 ± 0.5	100 ± 0.0	100 ± 0.0	99.9 ± 0.2	99.3 ± 0.7	99.6 ± 0.3	99.3 ± 0.7	99.9 ± 0.1
CS-Chi-2	60	96.9 ± 2.0	95.2 ± 3.9	97.8 ± 0.9	98.3 ± 0.9	95.8 ± 3.9	96.3 ± 3.7	98.5 ± 2.7	98.9 ± 1.2
	120	99.2 ± 0.9	99.4 ± 0.8	99.0 ± 1.1	99.3 ± 0.8	98.8 ± 0.4	98.1 ± 2.9	99.5 ± 0.5	99.9 ± 0.1
	180	99.9 ± 0.3	99.9 ± 0.3	99.6 ± 0.6	99.9 ± 0.2	99.2 ± 0.7	99.9 ± 0.1	100.0 ± 0.0	100.0 ± 0.0

Tabela 8 – Accuracy na classificação de falha de rolamento utilizando o classificador SVM e diferentes combinações de algoritmos de compressão

Sparsifying method		FFT				WT			
Sampling rate (α)		0.1	0.2	0.3	0.4	0.1	0.2	0.3	0.4
<i>Classifier: SVM</i>									
Method	k								
CS-FS	60	85.8 ± 8.4	89.5 ± 5.1	74.8 ± 5.6	69.9 ± 2.6	91.5 ± 8.4	95.2 ± 4.7	95.9 ± 3.7	95.7 ± 3.9
	120	89.6 ± 11.4	92.3 ± 6.2	98.4 ± 1.6	97.9 ± 3.9	94.9 ± 4.9	94.8 ± 5.2	97.5 ± 2.3	97.9 ± 2.4
	180	97.4 ± 5.2	99.1 ± 0.9	99.3 ± 0.7	99.8 ± 1.2	95.8 ± 3.9	96.3 ± 3.8	98.8 ± 1.4	99.4 ± 0.7
CS-LS	60	92.7 ± 7.0	92.1 ± 5.1	92.0 ± 0.7	95.3 ± 4.7	92.6 ± 7.4	92.2 ± 7.7	91.9 ± 8.3	92.7 ± 6.9
	120	98.7 ± 1.4	98.9 ± 1.1	99.3 ± 0.8	99.7 ± 0.3	91.4 ± 6.9	93.5 ± 6.4	94.7 ± 5.2	95.7 ± 4.2
	180	99.2 ± 0.8	99.2 ± 0.7	99.2 ± 0.9	99.9 ± 0.1	94.4 ± 5.4	93.2 ± 6.9	95.3 ± 4.6	97.8 ± 2.3
CS-Relief-F	60	85.8 ± 13.7	77.9 ± 1.5	68.4 ± 16.1	67.8 ± 16.4	94.7 ± 5.3	94.2 ± 5.9	92.7 ± 7.4	94.2 ± 4.8
	120	89.3 ± 8.4	90.9 ± 7.8	83.9 ± 12.2	78.5 ± 9.7	95.3 ± 4.7	95.5 ± 4.4	97.5 ± 2.7	98.8 ± 1.3
	180	95.6 ± 5.8	96.2 ± 4.8	88.0 ± 9.4	96.8 ± 4.5	97.7 ± 2.1	98.8 ± 1.2	99.3 ± 0.6	99.0 ± 1.2
CS-PCC	60	78.3 ± 14.0	73.2 ± 7.4	69.4 ± 13.1	65.1 ± 15.4	71.5 ± 17.8	79.9 ± 17.8	85.8 ± 13.2	94.7 ± 5.3
	120	93.2 ± 7.5	93.8 ± 7.9	93.3 ± 8.5	79.9 ± 2.2	78.2 ± 16.2	82.2 ± 15.8	88.2 ± 10.9	92.4 ± 7.5
	180	97.7 ± 5.1	96.7 ± 5.7	97.8 ± 2.8	99.3 ± 0.7	83.5 ± 13.7	87.4 ± 11.6	92.7 ± 7.6	99.3 ± 0.7
CS-Chi-2	60	64.2 ± 7.1	68.3 ± 7.6	63.7 ± 9.3	69.1 ± 18.4	94.3 ± 3.8	95.2 ± 3.2	97.7 ± 3.4	98.2 ± 2.8
	120	83.2 ± 9.0	76.9 ± 9.3	74.8 ± 11.9	71.4 ± 7.1	96.2 ± 3.7	97.1 ± 3.4	98.5 ± 1.7	98.8 ± 1.2
	180	95.2 ± 5.2	94.9 ± 6.3	90.7 ± 7.0	82.3 ± 14.4	99.6 ± 0.3	99.6 ± 0.2	99.8 ± 0.2	99.9 ± 0.2

Comparando os valores de *accuracy* apresentados nas Tabelas 6, 7 e 8 é possível concluir que: O uso de FFT apresenta melhor resultado que o uso da transformada de Wavelet; LRC e ANN têm a capacidade de alcançar alta *accuracy* com diferentes valores da taxa de amostragem (α) e um número de características selecionadas (k) para todas as combinações consideradas de seleção de características (*feature selection*).

Conforme comentário dos autores, os resultados mostram que a estrutura proposta atinge uma alta precisão de classificação com uma quantidade limitada de dados, utilizando várias combinações de métodos, que superam os resultados publicados recentemente.

Ince *et al.*, 2016 apresentaram o trabalho de classificação de falhas de motores causadas por falhas de rolamento utilizando o sinal de corrente elétrica. O trabalho comparou as técnicas de processamentos dos sinais do tipo FFT e Wavelet com três classificadores: Multi-Layer Perceptron (MLP), Radial Basis Function Networks (RBFN), e Support Vector Machines (SVM). O trabalho também comparou a aplicação do classificador desenvolvido durante a pesquisa, o qual denominou de Adaptive 1D CNN, e que não empregou os extratores de características. Os autores informam que o método proposto não requer qualquer forma de transformação, extração de características e pós-processamento. Ele pode trabalhar diretamente sobre os dados brutos, ou seja, o sinal de corrente do motor, para detectar as anomalias.

A Tabela 9 apresenta as métricas obtidas através dos experimentos realizados e é possível concluir que o uso da transformada de Wavelet, como extrator de características, apresenta métricas com valores mais altos, demonstrando maior acurácia do modelo, quando comparado com o uso de FFT.

Tabela 9 – *Comparação entre os extratores de características e classificadores*

Método	Detecção de falha		
	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>
Adaptativo 1D CNN	97,40	97,80	97,00
Wavelet - MLP	97,90	97,00	98,90
Wavelet - RBFN	99,80	100,00	99,70
Wavelet - SVM	99,20	100,00	98,30
FFT - MLP	92,70	90,80	95,10
FFT- RBFN	92,50	90,80	94,60
FFT - SVM	84,20	85,00	82,90

Fonte: INCE e colaboradores, 2016

3.4. CONCLUSÃO

Especificamente na utilização de aprendizagem profunda, através das pesquisas realizadas e do estado-da-arte apresentado, o modelo CNN ganha destaque por apresentar um nível de *accuracy* próximo ao estabelecido no objetivo geral deste trabalho, e pelo seu uso estar sendo ampliado no meio industrial em diferentes soluções, principalmente na área de manutenção preditiva.

Em função da análise de trabalhos acadêmicos que apresentam o estado-da-arte com o uso de sinais de vibração mecânica aplicados aos modelos de aprendizagem profunda como forma de classificar falhas, é possível constatar uma tendência no uso de aprendizagem profunda por apresentar maior *accuracy* e requerer menos trabalhos para geração do modelo.

É possível concluir que existem possibilidades de melhorias no desenvolvimento de modelos que apresentem uma maior *accuracy* na classificação de falhas de ativos, tais como: utilizar os dados no domínio da frequência como forma de otimizar o processamento dos dados; obter um modelo com menor número de *loss* e *epoch*; e permitir que o modelo classifique falhas de rolamento, desbalanceamento e desalinhamento, utilizando base de dados desbalanceadas.

Através dos trabalhos apresentados por Ahmed e Nandi (2018), e Ince e colaboradores (2016) demonstram que os extratores de características podem ser utilizados e a escolha do melhor tem de ser avaliado para cada experimento.

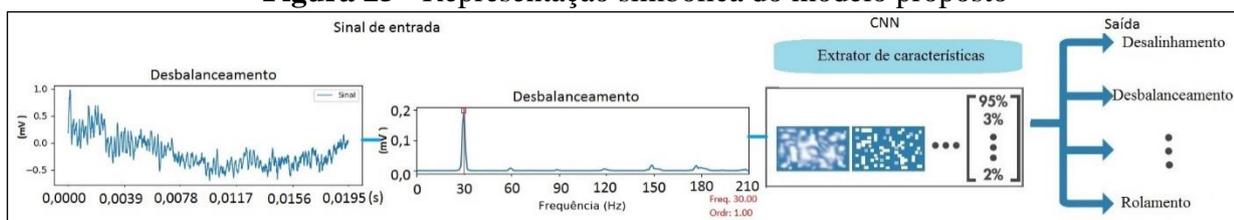
4. MÉTODO PROPOSTO

Este capítulo apresenta a metodologia utilizada para o desenvolvimento do modelo Manutenção Preditiva utilizando Rede Neural Convolutacional (MP-CNN), desenvolvido neste trabalho, que tem como principal função realizar a classificação de falhas de desalinhamento (horizontal e vertical), desbalanceamento e de rolamentos em ativos utilizando aprendizagem profunda e o modelo CNN.

Para o desenvolvimento da solução foi utilizada a base de dados pública (MaFaulDa); o framework, Tensorflow; bibliotecas Keras e Panda e programação em Python. Neste capítulo também serão apresentados os experimentos, arquiteturas utilizadas, além dos hiperparâmetros finais.

Considerando as características da CNN, foi proposto e desenvolvido um sistema de classificação de falha em ativos, conforme a Figura 25, no qual se tem como entrada do sistema o sinal de vibração no domínio do tempo, que é transformado para o domínio da frequência e a CNN é capaz de extrair as principais características do sinal e fornecer a classificação e severidade da falha.

Figura 25 - Representação simbólica do modelo proposto



Fonte: O autor, 2020

4.1. TRATAMENTO DOS DADOS

Durante o processo de modelagem, diferentemente dos trabalhos anteriores (Viana (2016), Ribeiro (2017), Marins (2018), Martins (2018) e Rocha (2018)) que utilizaram todos os dados dos sensores, o presente trabalho utilizou os dados dos três sinais de vibração do tipo acelerômetro unidirecional, o que equivale a um acelerômetro triaxial, instalados no mancal interno, conforme posição 4 na Figura 7b. Portanto, não foram utilizados os sinais do sensor de vibração do tipo acelerômetro triaxial instalado no mancal externo, conforme posição 8 da Figura 7b, assim como o microfone e o tacômetro. Foi decidido testar essa configuração por ser bastante utilizada em processos industriais, além de reduzir custo de aquisição e instalação de sensor. Posteriormente, para cada dado do sensor de vibração, foi extraída a FFT correspondente e utilizada como vetor de entrada para as tarefas de treinamento, validação e teste. A FFT não recebeu tratamento de normalização dos dados. A extração de recursos oferece uma grande vantagem ao reduzir drasticamente o tamanho do vetor de entrada em comparação

com o número de amostras na série temporal. Além disso, esta etapa do processo permite a utilização de um vetor com maior capacidade de discriminação das classes de falhas do que o obtido com as séries temporais originais.

4.2. DESENVOLVIMENTO EXPERIMENTAL

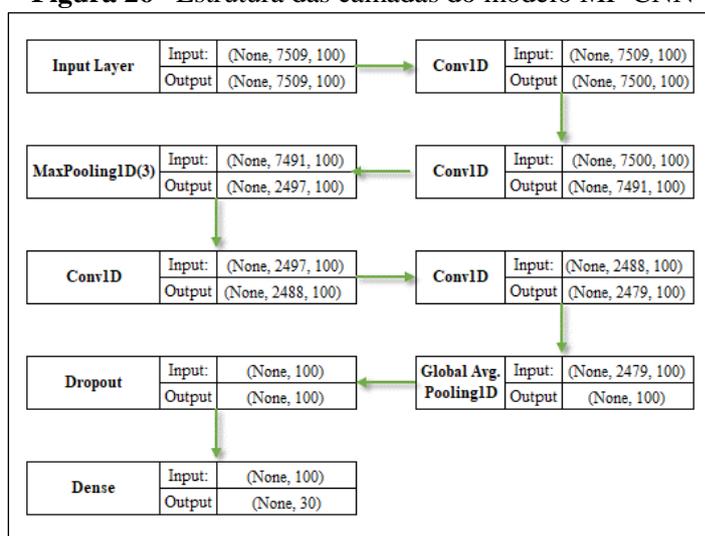
Neste experimento, o modelo foi gerado utilizando um Computador de Alto Desempenho (HPC), usando um node com dois processadores Xeon(R) Gold 6148 @ 2,40 GHz com um total de 40 threads e 187 GB de memória RAM.

Foi utilizado o Tensorflow (TENSORFLOW), uma biblioteca de código aberto para aprendizagem de máquina e aprendizagem profunda, para realizar a parametrização e desenvolvimento do modelo.

Foram utilizadas a linguagem de programação Python, as bibliotecas pandas e keras.

A arquitetura do modelo MP-CNN que obteve o melhor desempenho foi desenvolvida conforme Figuras 26 e 27 utilizando: uma camada de entrada; quatro camadas de convolução, utilizando a função de ativação ReLU; uma camada *pooling*, utilizando a função *Maxpooling*; uma camada *pooling*, utilizando a função *Global Average Pooling*; e uma camada totalmente conectada, utilizando a função *softmax*.

Figura 26– Estrutura das camadas do modelo MP-CNN



Fonte: O autor, 2020

Figura 27– Resumo do modelo MP-CNN construído para extrair características

Layer (type)	Output Shape	Param #
conv1d_5 (Conv1D)	(None, 7500, 100)	3100
conv1d_6 (Conv1D)	(None, 7491, 100)	100100
max_pooling1d_2 (MaxPooling1D)	(None, 2497, 100)	0
conv1d_7 (Conv1D)	(None, 2488, 100)	100100
conv1d_8 (Conv1D)	(None, 2479, 100)	100100
global_average_pooling1d_2 (GlobalAveragePooling1D)	(None, 100)	0
dropout_2 (Dropout)	(None, 100)	0
dense_2 (Dense)	(None, 30)	3030
Total params: 306,430		
Trainable params: 306,430		
Non-trainable params: 0		

Fonte: O autor, 2020

Na camada de entrada (*Input Layer*) foi utilizada a função *input_shape*, como o objetivo de definir o formato de entrada, a qual foi parametrizada pelo tamanho do vetor da matriz do arquivo utilizado, e pela dimensão do vetor igual a três que corresponde aos três sensores de vibração instalados no mancal interno, ou seja, *input_shape=(tamanho_fft, 3)*; as camadas de convolução foram parametrizadas com o filtro igual a 100, o tamanho do [kernel](#) igual a 10 e a função de ativação utilizada foi ReLU, ou seja, *Conv1D(100, 10, activation='relu')*; a camada *pooling* foi parametrizada com o tamanho de janela igual a três, ou seja, *Maxpooling1D(3)*; um *dropout* de 0,5 foi selecionado como parâmetro de regularização *Dropout(0.5)*; a camada totalmente conectada foi parametrizada com um número de classes igual a 30, correspondente à quantidade de falhas, e à função de ativação *softmax*, ou seja, *Dense(num_classes, activation='softmax')*.

A função de ativação ReLU juntamente com a função *dropout* foi utilizada para evitar o aparecimento de *leakage gradient*, nos processos de treinamento e validação. A função *dropout* também foi utilizada com objetivo de atingir a convergência mais rápida. Para as tarefas de treinamento e validação, 65% dos dados foram selecionados aleatoriamente da base de dados, enquanto 35% dos dados foram designados para testar o modelo. A classe 'train_test_split' da biblioteca sklearn foi usada para dividir a base de dados randomicamente em tarefas de treinamento/validação e teste. O *loss* foi calculado utilizando a função objetiva da entropia transversal esparsa e categórica ([Sparse Multiclass Cross-Entropy](#)). Utilizando essa configuração de parâmetros, o modelo foi gerado em 1 h e 55 min.

5. RESULTADOS E DISCUSSÕES

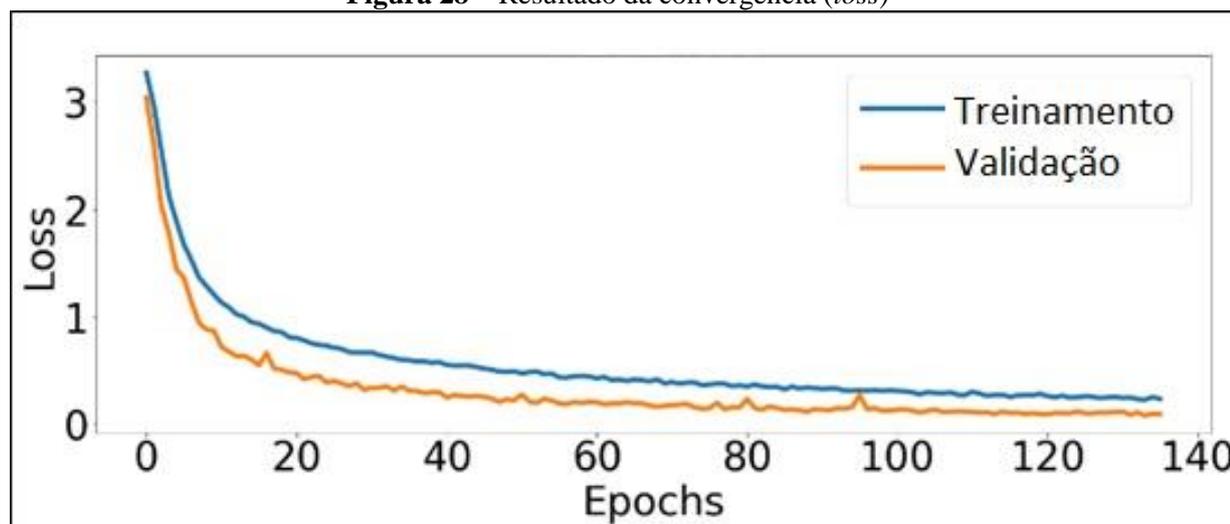
Este capítulo apresenta os resultados, analisa-os e compara-os com trabalhos anteriores que utilizaram a mesma base de dados (MaFaulDa).

Dois experimentos de classificação de falhas foram usados. O primeiro experimento usou as classes de falha com diferentes níveis de severidade, conforme Tabela 11, e, o segundo, agrupou as severidades individuais nos seus respectivos grupos de falha, conforme Tabela 13. A *accuracy* do modelo, para cada experimento de classificação, foi de 97,26% e 99,58%, respectivamente.

A seguir, são apresentados os comentários referente ao primeiro experimento de classificação utilizando as classes e classificando as falhas de acordo com os níveis de severidade.

Embora o número de vezes (*epochs*) para treinar e validar o modelo tenha sido fixado em 300 *epochs*, foi possível avaliar a convergência em estado estacionário com 135 *epochs*, conforme apresentado na Figura 28.

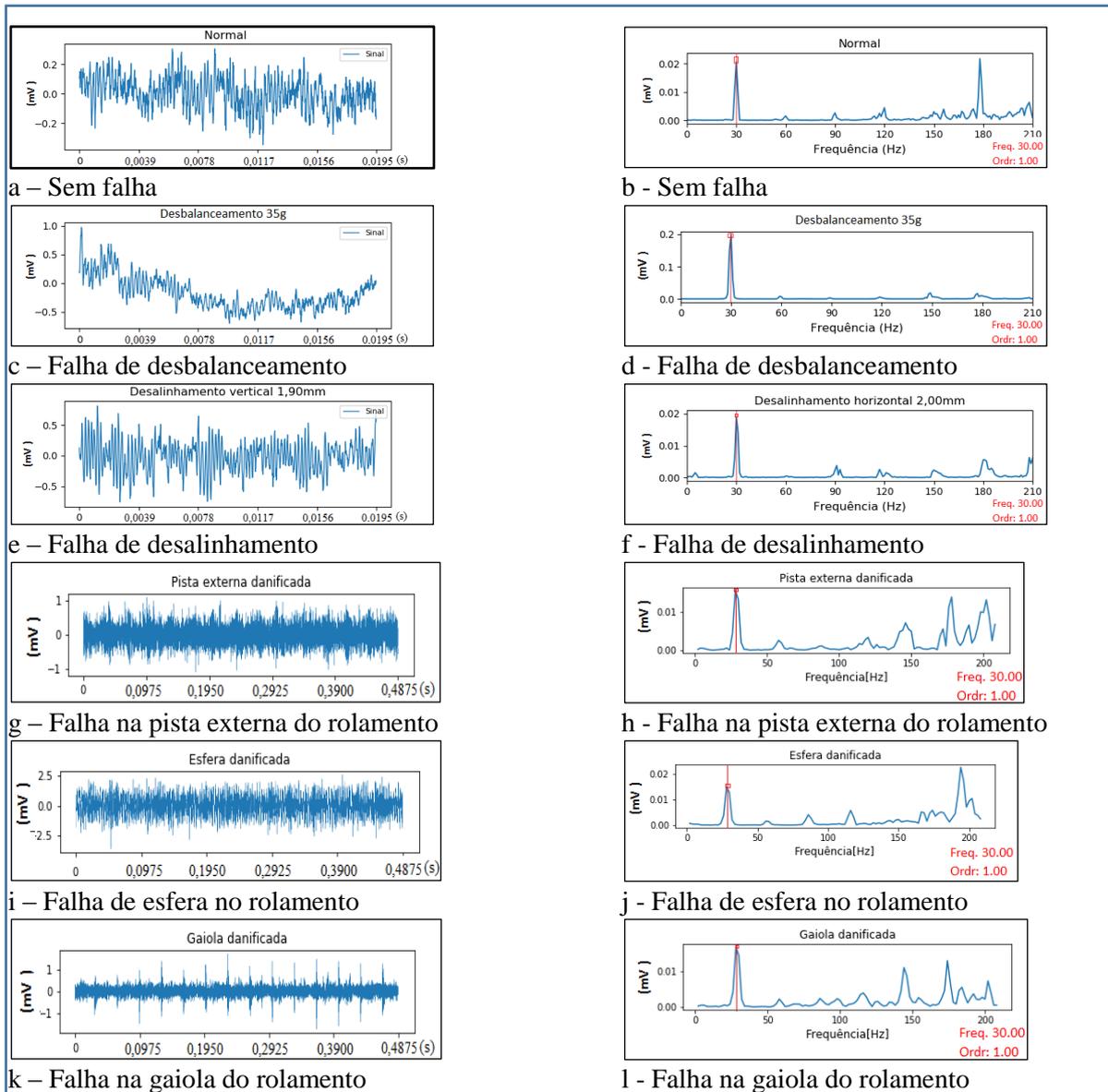
Figura 28 – Resultado da convergência (*loss*)



Fonte: O autor, 2020

A conversão dos dados do domínio do tempo para o domínio da frequência, através de FFT, permitiu otimizar o tempo para treinamento, validação e teste do modelo. Em função da conversão dos dados do domínio do tempo para o domínio da frequência foi possível otimizar a quantidade de dados representativos para cada acelerômetro e extrair as características de cada falha. A figura 29 representa os dados no domínio do tempo e no domínio da frequência dos diferentes tipos de falhas e não apresenta toda a quantidade de dados disponível na base de dados MaFaulDa, mas parte desse. Através desses gráficos é possível verificar a redução da quantidade de dados e a extração das características das falhas, utilizando o gráfico no domínio da frequência.

Figura 29 – Gráficos representando as falhas no domínio do tempo e domínio da frequência



Gráficos no domínio do tempo: a, c, e, g, i, k Gráficos no domínio da frequência: b, d, f, h, j, l

Fonte: O autor, 2020

Após a conversão dos dados do domínio do tempo para o domínio da frequência, os dados foram apresentados ao modelo MP-CNN.

A Tabela 10 apresenta as métricas de avaliação (*accuracy*, *precision*, *recall* e *F1-Score*) do modelo MP-CNN utilizadas para classificar as falhas conforme o nível de severidade. No caso de desalinhamento horizontal de 1,0 mm, o resultado das métricas foi de 100%, o que representa que não houve FP nem FN, ou seja, todas estas falhas foram classificadas corretamente. No caso do desalinhamento vertical de 1,4 mm as métricas foram 0,840; 0,980; 0,854 e 0,913, respectivamente, comprovando que houveram FP e FN, ou seja, houveram casos em que as falhas não foram diagnosticadas corretamente. O modelo MP-CNN calcula as métricas através

de funções específicas implementadas no modelo, mas também é possível realizar os cálculos manualmente, conforme equações 2.12 a 2.15, respectivamente.

Tabela 10 – Métricas do modelo de classificação de falhas de acordo com a severidade

Accuracy média: 97,258%				
	<i>Accuracy</i>	<i>Precision</i>	<i>recall</i>	<i>F1-score</i>
Operação sem falha	1,000	1,000	1,000	1,000
Desalinhamento horizontal: 0.5mm	0,994	1,000	0,994	0,997
Desalinhamento horizontal: 1.0mm	1,000	1,000	1,000	1,000
Desalinhamento horizontal: 1.5mm	0,993	1,000	0,993	0,996
Desalinhamento horizontal: 2.0mm	1,000	1,000	1,000	1,000
Desalinhamento vertical: 0.51mm	0,900	0,986	0,910	0,946
Desalinhamento vertical: 0.63mm	0,960	0,960	1,000	0,980
Desalinhamento vertical: 1.27mm	0,966	1,000	0,966	0,983
Desalinhamento vertical: 1.40mm	0,840	0,980	0,854	0,913
Desalinhamento vertical: 1.78mm	0,944	0,956	0,987	0,971
Desalinhamento vertical: 1.90mm	0,787	0,787	1,000	0,881
Desbalanceamento em: 6g	0,994	1,000	0,994	0,997
Desbalanceamento em: 10g	1,000	1,000	1,000	1,000
Desbalanceamento em: 15g	0,994	1,000	0,994	0,997
Desbalanceamento em: 20g	1,000	1,000	1,000	1,000
Desbalanceamento em: 25g	0,986	0,986	1,000	0,993
Desbalanceamento em: 30g	0,976	0,988	0,987	0,987
Desbalanceamento em: 35g	0,977	0,977	1,000	0,988
Falha na esfera: 0g	0,994	1,000	0,994	0,997
Falha na esfera: 6g	1,000	1,000	1,000	1,000
Falha na esfera: 20g	1,000	1,000	1,000	1,000
Falha na esfera: 35g	0,992	0,992	1,000	0,996
Falha na gaiola: 0g	1,000	1,000	1,000	1,000
Falha na gaiola: 6g	0,977	1,000	0,977	0,988
Falha na gaiola: 20g	0,984	1,000	0,984	0,992
Falha na gaiola: 35g	0,992	0,992	1,000	0,996
Falha na pista externa: 0g	0,994	0,994	1,000	0,997
Falha na pista externa: 6g	0,994	0,994	1,000	0,997
Falha na pista externa: 20g	0,994	0,994	1,000	0,997
Falha na pista externa: 35g	0,961	0,961	1,000	0,980
Média	0,973	0,985	0,988	0,986

Fonte: O autor, 2020

Outra forma de avaliação do modelo é através da matriz de confusão, conforme Tabela 11. As últimas colunas representam as métricas (*accuracy*, *precision*, *recall* e *F1-Score*) individuais, as células na diagonal principal, na cor verde, correspondem às amostras que foram classificadas corretamente e as células que estão na cor amarelo foram classificados como falso positivo ou falso negativo. Por fim, as células nas últimas linhas, na cor vermelho, corresponde ao valor médio das métricas.

Os resultados foram os seguintes:

- A classificação Falha na Gaiola mostrou uma *accuracy* média de 98,8% em função da classificação equivocada da solução, gerando FP e FN, o que reflete um problema prático quando o analista de vibração pode ser induzido a classificar a falha equivocadamente. Nem sempre é fácil identificar esse tipo de falha como um problema isolado e nem sempre é relacionado somente ao problema na gaiola do rolamento. Problemas de rolamentos (falha na esfera, gaiola ou pista externa), na prática, geralmente não ocorrem isoladamente, e a falha pode ocorrer junto com desbalanceamento e/ou desalinhamento. Conforme explanado no item 2.2.4.c, foram adicionadas diferentes massas para que a falha no rolamento pudesse ser mais perceptível. Portanto, eles geralmente aparecem juntos no espectro de vibração. Na prática, após o rolamento ser substituído, é realizado um novo teste de desbalanceamento/desalinhamento para garantir que o problema foi resolvido.
- Com relação à classificação da gravidade da falha de desalinhamento vertical, a *accuracy* média obtida foi de 90,2%. O modelo foi capaz de provar que o desalinhamento maior que 0,63 mm para a configuração (tamanho, massa, potência do motor) deste ativo foi muito alto. Portanto, neste caso, quanto maior o desalinhamento, menores são os valores de *precision* e *recall* da falha em função da possibilidade de outras falhas sobrepostas.
- O sistema demonstrou *accuracy* média superior a 99% em várias classificações de severidade, como normal (sem falha), desalinhamento horizontal, falha da esfera e falha da pista externa.

Em relação ao segundo experimento realizado, quando foram classificados os tipos de falhas, agrupando as severidades nos seus respectivos grupos, temos abaixo os resultados.

As métricas do segundo experimento estão listadas na Tabela 12, na qual é possível analisar a *accuracy*, *precision*, *recall* e *F1-Score* para cada tipo de falha. A *accuracy* média do experimento foi de 99,58%. Esse resultado demonstrou que é possível o uso da aprendizagem profunda aplicando o modelo MP-CNN com o objetivo de classificar as falhas analisadas através do sinal de vibração.

Tabela 12 – Métricas do modelo de classificação de falhas

Accuracy média: 99,58%				
	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Operação sem falha	1,000	1,000	1,000	1,000
Desalinhamento Horizontal	0,997	1,000	0,997	0,998
Desalinhamento Vertical	0,998	0,998	1,000	0,999
Desbalanceamento	1,000	1,000	1,000	1,000
Falha na esfera	0,998	1,000	0,998	0,999
Falha na gaiola	0,989	0,998	0,989	0,993
Falha na pista externa	0,987	0,987	1,000	0,993
Média	0,996	0,998	0,998	0,998

Fonte: O autor, 2020

Através da matriz de confusão, conforme Tabela 13, é possível concluir que:

- As operações normais (sem falha), desalinhamento horizontal, desbalanceamento e falha de esfera e falha na gaiola apresentaram valores iguais a 100% de *accuracy*, demonstrando que não houve classificação de FP e FN.
- A operação normal, desalinhamento vertical, desbalanceamento e falha na pista externa apresentaram um *recall* de 100%, demonstrando que não houve classificação de FN.
- Embora a classificação da severidade para as falhas de desalinhamento vertical, no primeiro experimento, tenha obtido uma *accuracy* média de 90,2%. Quando realizou o segundo experimento, agrupando as severidades, o modelo obteve uma *accuracy* de 99,8%, demonstrando a capacidade do modelo em classificar corretamente o tipo de falha, independente da severidade.

Tabela 13 – Matriz de confusão – Classificação de falhas

		PREDIÇÃO						Métricas			
		Normal	Desalinhamento Horiz. (mm)	Desalinhamento Vert. (mm)	Desbalanceamento (g)	Falha na esfera (g)	Falha na gaiola	Falha na pista externa	Accuracy	Precision	Recall
REAL	Normal	174	0	0	0	0	0	1,000	1,000	1,000	1,000
	Desalinhamento Horizontal	0	681	1	0	0	1	0,997	1,000	0,997	0,998
	Desalinhamento Vertical	1	1	1039	0	0	0	0,998	0,998	1,000	0,999
	Desbalanceamento	0	0	0	1170	0	0	1,000	1,000	1,000	1,000
	Falha na esfera	0	0	0	0	646	0	0,998	1,000	0,998	0,999
	Falha na gaiola	0	0	0	1	0	679	0,989	0,998	0,989	0,993
	Falha na pista externa	0	0	0	0	2	6	0,987	0,987	1,000	0,993
		Média						0,996	0,998	0,998	0,998

Fonte: O autor, 2020

Em comparação com resultados apresentados através de trabalhos acadêmicos utilizando a mesma base de dados (MaFaulDa), que foi utilizada para treinar, validar e testar esse experimento, verificou-se que este atingiu a *accuracy* média de 99,58%. Portanto, o modelo MP-CNN atingiu um nível excelente, ao analisar as métricas, e superior aos trabalhos publicados anteriormente, conforme comparação realizada através da Tabela 14.

Tabela 14 – Comparativo com outros trabalhos acadêmicos utilizando a base de dados MaFaulDa

Referência	Método	Accuracy (%)	Observação
VIANA, LÓPES, et al., 2016	MLP	95,80	
RIBEIRO, MARINS, et al., 2017	SBM + RFC	96,43	
MARINS, RIBEIRO, et al., 2018	SBM	98,50	
MARTINS, HEMERLY, et al., 2018	SBM + Random Forest	94,14	O objetivo do experimento foi verificar a exatidão para falhas de desbalanceamento
ROCHA, 2018	XGBoost + Transformada Wavelet de Haar	98,70	Apresentou as métricas accuracy e o produto entre Recall e sensibilidade.
SOUZA, 2019	MP-CNN	99,58	Modelo desenvolvido pelo autor

Fonte: O autor, 2020

O trabalho anterior com a melhor *accuracy* reportada é de 98,7%, apresentado por Rocha (2018) utilizando o classificador [XGBoost](#) e a transformada *Wavelet* de *Haar*. Para modelar o experimento, o autor empregou a base de dados (MaFaulDa) e classificou as falhas de desalinhamento, desbalanceamento e rolamento. O autor informou as *accuracies* individuais dos experimentos, conforme Tabela 15, e é possível constatar que o modelo desenvolvido apresentou menor *accuracy* na classificação das falhas de desalinhamento vertical e horizontal. Neste trabalho, realizou-se a classificação de falha de desalinhamento vertical e a severidade, o valor de *accuracy* foi o menor entre todas as falhas, demonstrando a dificuldade de classificação dessa falha.

Tabela 15– Accuracy do modelo XGBoost + Transformada *Wavelet de Haar*

Falha	Accuracy (%)
Normal	99,00%
Desalinhamento horizontal	98,76%
Desalinhamento Vertical	97,98%
Desbalanceamento	99,20%
Mancal Interno	99,63%
Mancal Externo	99,92%

Fonte: Adaptado Rocha, 2018

O segundo trabalho com a melhor *accuracy* reportada de 98,5% foi apresentado em Marins e colaboradores (2018) utilizou o método [Similarity-Based Modeling \(SBM\)](#) e o método [Random Forest](#), como classificador. A metodologia nesse trabalho consistiu em três etapas: pré-processamento dos dados, identificação das características através do método [SBM](#) e a classificação através do método [Random Forest](#). A etapa de pré-processamento converteu os dados brutos (valores baseados no tempo) em características estatísticas (curtose, média e desvio padrão) de cada sinal e transformou os dados do domínio do tempo para o domínio da frequência. Na segunda etapa, as características convertidas foram encaminhadas para o modelo [SBM](#) que comparou a semelhança entre o valor atual e o modelo existente. Na terceira etapa, os dados foram encaminhados para o método [Random Forest](#), que foi responsável pela classificação da falha. Os dados processados resultaram em 46 características, a saber: frequência de rotação (f_r) obtida do tacômetro, 21 características de frequência (f_r , $2 \times f_r$ e $3 \times f_r$) para cada sinal e 24 outras características estatísticas (média, curtose e entropia para cada um dos oito sinais). Para obter os resultados satisfatórios, trabalhos extras foram necessários para gerarem as características e serem utilizados no modelo.

A *accuracy* obtida no modelo MP-CNN foi maior do que o melhor resultado apresentado até agora em Rocha (2018) utilizando a mesma base de dados MaFaulDa. No modelo MP-CNN, não houve necessidade de trabalhos extras para o pré-processamento dos sinais; mostrou-se a possibilidade de utilizar apenas a conversão dos dados no domínio do tempo para o domínio da frequência para converter as características e, em seguida, realizar a classificação com uma *accuracy* mais alta. A aplicação do modelo MP-CNN mostrou-se capaz de superar as três etapas apresentadas em Marins e colaboradores (2018) e demonstrou maior *accuracy*. Outra grande vantagem do modelo MP-CNN foi a possibilidade de ter utilizado os sensores de vibração no mancal interno e, ainda assim, conseguir classificar com boa precisão as falhas do rolamento do mancal externo e provar ser capaz de atingir uma maior *accuracy*.

O resultado obtido neste trabalho demonstra que o modelo MP-CNN contribui para evolução do estado-da-arte em pesquisas na área de vibração mecânica e pode ser empregado no contexto de classificação de falha de ativos utilizando sinais de vibração.

6. CONCLUSÃO

O modelo MP-CNN atinge um novo patamar no estado-da-arte, em relação aos trabalhos analisados, pelos seguintes motivos:

- Obteve uma *accuracy* de 99,58%, portanto, superior aos demais trabalhos que utilizaram a base de dados MaFaulDa;
- Utilizou apenas um extrator de característica (FFT) para facilitar o processo de modelagem;
- Foi o primeiro a utilizar o modelo CNN, utilizando a base de dados MaFaulDa;
- Utilizou apenas os três acelerômetros unidirecionais instalados no mancal interno e mesmo assim obteve uma *accuracy* superior aos demais trabalhos;
- Utilizou a base de dados MaFaulDa para classificar diferentes tipos de falhas, contrapondo outros trabalhos que classificaram apenas falhas de rolamento;
- Outra vantagem desse modelo foi poder demonstrar a sua capacidade de adaptação, ao utilizar uma base de dados desbalanceada, ou seja, com quantidade de dados diferente para os diferentes tipos de falhas.
- O valor médio de *accuracy* (99,58%) foi superior quando as falhas foram agrupadas, o que significa que o resultado foi bom independente da severidade.

A partir deste estudo, a utilização do modelo MP-CNN para classificação de falha utilizando sinais de vibração demonstrou ser possível e viável. A classificação das falhas considerando as severidades apresentou uma *accuracy* de 97,26%, o que é um nível de precisão elevado, enquanto que, se forem consideradas as classes de falha agrupadas, independentemente dos níveis de severidade, obteve uma *accuracy* de 99,58%, o que também é um nível muito elevado. Assim, o modelo MP-CNN mostrou um grande potencial na classificação de falhas de ativos com diferentes níveis de severidade. O modelo MP-CNN demonstrou seu potencial para implementação em uma instalação industrial com o objetivo de otimizar os custos financeiros de manutenção, aumentar a confiabilidade do ativo, reduzir os custos de aquisição de sensores para identificação e classificação de falhas, além de facilitar o caminho para a transformação digital necessária para a quarta revolução industrial.

O modelo desenvolvido atingiu as premissas estabelecidas no objetivo geral ao prover uma solução utilizando aprendizagem profunda, através do modelo MP-CNN; ser capaz de classificar as falhas dos ativos, ter determinado a severidade das falhas. Como consequências da implementação desse modelo, será possível ter um aumento de confiabilidade do ativo, em função da possibilidade de obter respostas instantâneas ao diagnosticar uma falha.

6.1. SUGESTÕES

Para trabalhos futuros, sugere-se:

- Coletar sinais de vibração de ativos em instalação industrial que apresentem condição de operação sem falha e com falhas, testar o modelo com o objetivo de verificar sua *accuracy* e a capacidade de adaptação a sinais com ruído, caso tenha;
- Ampliar a aplicação para outros tipos de falhas, por exemplo: engrenagens, desalinhamento de correia.
- Desenvolver uma base de dados e torna-la pública, simulando falhas de engrenagens e desalinhamento de correia.
- Avaliar o resultado da aplicação do modelo para identificar falhas em sinais com múltiplos defeitos.

REFERÊNCIA

AHMED, H.; NANDI, A. K. Compressive Sampling and Feature Ranking Framework for Bearing Fault Classification With Vibration Signals. **IEEE Access**, 2018. 44731-44746.

BENGIO, Y. Learning Deep Architectures for AI. **Now - The essence of Knowledge**, Montreal, Canada, 15 nov. 2009. 133.

BROWNLEE, J. Dropout Regularization in Deep Learning Models With Keras. **Machine Learning Mastery**, 2016. Disponível em: <<https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/>>. Acesso em: 29 ago. 2019.

CHAPPELL, D. Introducing azure machine learning - A guide for technical professionals, San Francisco, 2018. 1-17.

CHATTOPADHYAY, P. et al. Deep Learning in Fault Diagnosis of Induction Motor Drives. **2018 Prognostics and System Health Management Conference**, Chongqing, 27 out. 2018. 1068-1073.

CHOLLET, F. **Deep learning with Python**. 1ª. ed. New York: Manning Publications Co, v. I, 2018.

COPPE - UFRJ. Machinery Fault Database - MaFaulDa, Rio de Janeiro, 18 out. 2018.

DAVIS, J.; GOADRICH, M. The relationship between Precision-Recall and ROC. **Proceedings of the 23rd international conference on Machine learning - ICML**, Madison, 2006. 233-240.

FILHO, L. F. S. **Manutenção por análise de vibrações: Uma valiosa ferramenta para gestão de ativos**. Universidade Federal do Rio de Janeiro - Escola Politécnica. Rio de Janeiro, p. 57. 2013.

GARCIA, M. S. **ANÁLISE DE DEFEITOS EM SISTEMAS MECÂNICOS ROTATIVOS A PARTIR DA MONITORAÇÃO DE VIBRAÇÕES**. UFRJ. Rio de Janeiro, p. 130. 2005.

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

GOOGLE DEVELOPERS. Machine Learning Crash Course, 2019. Disponível em: <<https://developers.google.com/machine-learning/crash-course>>. Acesso em: 13 nov. 2019.

GUO, X.; CHEN, L.; SHEN, C. Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. **Measurement**, 15 nov. 2016. 490-502.

HAYKIN, S. **Neural Networks and Learning Machines**. 3ª. ed. New Jersey: Pearson Education, Inc, v. I, 2009.

INCE, T. et al. Real-Time Motor Fault Detection by 1-D Convolutional Neural Networks. **IEEE Transactions on Industrial Electronics**, nov. 2016. 7067-7075.

ISO. International Organization for Standardization, 2015. Disponível em: <<https://www.iso.org/standard/60328.html>>. Acesso em: 21 junho 2018.

JANSSENS, O. et al. Convolutional Neural Network Based Fault Detection for Rotating Machinery. **Journal of Sound and Vibration**, 17 maio 2016. 331-345.

JERKIC, M. What Is Dropout? – Deep Learning. **MARKOJERKIC.COM**, 2017. Disponível em: <<https://markojerkic.com/what-is-dropout-deep-learning/>>. Acesso em: 29 ago. 2019.

JIA, F. et al. Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization. **Mechanical Systems and Signal Processing**, Shaanxi, 12 mar. 2018. 349-367.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. **Advances in Neural Information Processing Systems**, 06 jun. 2012. 1097--1105.

KUMAR, S. et al. Condition based maintenance of bearings and gears for fault detection – A review. **Materials Today - ScienceDirect**, India, 2018. 6128-6137.

LACEY, S. J. The Role of Vibration Monitoring in Predictive Maintenance. **Engineering Manager Schaeffler (UK) Limited**, 2017. Disponível em: <https://www.schaeffler.com/remotemedien/media/_shared_media/08_media_library/01_publications/schaeffler_2/technicalpaper_1/download_1/the_role_of_vibration_monitoring.pdf>. Acesso em: 08 Julho 2018.

LAVI, Y. The Rewards and Challenges of Predictive Maintenance. **InfoQ**, 2018. Disponível em: <<https://www.infoq.com/articles/predictive-maintenance-industrial-iot/>>. Acesso em: 20 jan. 2018.

LU, P. et al. Assessment of Data-Driven, Machine Learning Techniques for Machinery Prognostics of Offshore Assets. **Offshore Technology Conference**, Texas, 2 maio 2017.

MARINS, M. A. et al. Improved similarity-based modeling for the classification of rotating-machine failures. **Journal of the Franklin Institute**, mar. 2018. 1913-1930.

MÁRQUEZ, A. C. **The Maintenance Management Framework**. 1º. ed. [S.l.]: Springer-Verlag London, v. 1, 2007.

MARTINS, D. H. C. D. S. S. et al. Application of Machine Learning to Evaluate Unbalance Severity in Rotating Machines. **Proceedings of the 10th International Conference on Rotor Dynamics – IFToMM. IFToMM 2018**, 19 ago. 2018. 144-160.

MEOLA, T. **MONITORAMENTO EM TEMPO REAL DA QUALIDADE DE SINAIS DE VIBRAÇÕES, UTILIZANDO INTELIGÊNCIA ARTIFICIAL**. Universidade Federal de Uberlândia. Uberlândia, p. 130. 2005.

MICROSOFT AZURE. **INTRODUCING AZURE MACHINE LEARNING. A GUIDE FOR TECHNICAL PROFESSIONALS**, San Francisco, 2018. 1-17. Disponível em: <http://download.microsoft.com/download/3/b/9/3b9fba69-8aad-4707-830f-6c70a545c389/introducing_azure_machine_learning.pdf>. Acesso em: 16 Setembro 2018.

NASCIMENTO, E. G. S. **Um Algoritmo Baseado em Técnicas de Agrupamento para Detecção de Anomalias em Séries Temporais Utilizando a Distância de Mahalanobis**. Universidade Federal do Espírito Santo. Vitória, p. 63. 2012.

POLAMURI, S. Difference between softmax function and sigmoid function. **Dataaspirant**, 2017. Disponível em: <<http://dataaspirant.com/2017/03/07/difference-between-softmax-function-and-sigmoid-function/>>. Acesso em: 29 ago. 2019.

PRIDESYS IT LIMITED. **Machine Learning in Bangladesh**, 2018. Disponível em: <<http://pridesys.com/machine-learning-in-bangladesh/>>. Acesso em: 05 ago. 2019.

RIBEIRO, F. M. L. et al. Rotating machinery fault diagnosis using similarity-based models. **XXXV SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS**, São Pedro, 3-6 set. 2017. 5.

ROCHA, D. S. C. **APRENDIZADO DE MÁQUINA APLICADO AO RECONHECIMENTO AUTOMÁTICO DE FALHAS EM MÁQUINAS ROTATIVAS**. Belo Horizonte: Universidade Federal de Minas Gerais, 2018.

SOBIE, C.; FREITAS, C.; NICOLAI, M. Simulation-driven machine learning: Bearing fault classification. **Mechanical Systems and Signal Processing**, Leu ven, 2 jun. 2017. 403-419.

SRIVASTAVA, N. et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. **Journal of Machine Learning Research**, 2014. 1929-1958.

SWANSON, L. Linking maintenance strategies to performance. **International Journal of Production Economics**, 18 Abril 2001. 237-244.

TENSORFLOW. **Get Started with TensorFlow. TensorFlow**. Disponível em: <<https://www.tensorflow.org/tutorials>>. Acesso em: 22 ago. 2018.

THOMSON, W. ; DAHLEH, M. D. **Theory of vibration with applications**. 5^a. ed. [S.l.]: Pearson Education, v. I, 2005.

VIANA, D. P. et al. The Influence of Feature Vector on the Classification of Mechanical Faults using Neural Networks. **VII Latin American Symposium on Circuits and Systems (LASCAS) 2016**, 2016. 115-118.

VIEIRA, V.; RAZENTE, H. L.; BARIONI, M. C. N. **TÓPICOS EM GERENCIAMENTO DE DADOS E INFORMAÇÕES**. Uberlândia: Sociedade Brasileira de Computação - SBC, 2017.

ZHANG, W. et al. A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. **Mechanical Systems and Signal Processing**, Heilongjiang Province, 19 jun. 2017. 439-453.

ZHANG, W. et al. A New Deep Learning Model for Fault Diagnosis with Good Anti-Noise and Domain Adaptation Ability on Raw Vibration Signals. **Sensors**, Harbin, 22 fev. 2017. 12.

ZHAO, R. et al. Deep learning and its applications to machine health monitoring. **Mechanical Systems and Signal Processing**, 27 maio 2018. 213–237.

ZHU, X. et al. An SDP Characteristic Information Fusion-Based CNN Vibration Fault Diagnosis Method. **Shock and Vibration**, 31 mar. 2019. 14.

GLOSSÁRIO

A

Autoencoder (AE) – É um modelo utilizado em Rede Neural Artificial (RNA) não-supervisionada que aprende a comprimir e codificar dados de forma eficiente, em seguida, aprende a reconstruir os dados de volta a partir da representação codificada reduzida para uma representação que é o mais próximo possível da entrada original.

B

Bayesian Linear Regression - A regressão linear é formulada utilizando distribuições de probabilidade em vez de estimativas pontuais. O objetivo da Regressão Linear Bayesiana não é encontrar o "melhor" valor único dos parâmetros do modelo, mas sim determinar a distribuição posterior para os parâmetros do modelo.

Big data - São dados com maior variedade que chegam em volumes crescentes e com velocidade cada vez maior. Isso é conhecido como os três Vs, ou seja, é um conjunto de dados maior e mais complexo, especialmente de novas fontes de dados. No entanto, esses grandes volumes de dados podem ser usados para resolver problemas de negócios que você não conseguiria resolver antes.

Boosted Decision Tree Regression – É um dos vários modelos clássicos para criar modelos *Ensemble*, juntamente com *bagging*, [Random Forest](#) e assim por diante. *Boosted Decision Tree* usa uma implementação eficiente do algoritmo de *Gradient Boosting* do tipo MART. *Gradient Boosting* é um modelo de aprendizagem de máquina para problemas de regressão. Ele constrói cada árvore de regressão de forma gradual, usando uma função *loss* predefinida para medir o erro em cada etapa e corrigí-lo na próxima. Assim, o modelo de previsão é na verdade um conjunto de modelos de previsão mais fracos.

D

Data driven – É um método baseado em dados que podem ser usados para o diagnóstico de falhas em sistemas complexos e difíceis de modelar, identificando mudanças nos dados de medição que se correlacionam bem com mudanças na condição do sistema. São dados coletados de ativos através de sensores.

Denoising Autoencoder – É uma extensão do autoencoder básico e representa uma versão estocástica do mesmo. Os *Denoising Autoencoder* tentam abordar o risco da função identidade através da introdução de ruído que o autoencoder deve então reconstruir.

E

Epochs - O número de *epochs* é um hiperparâmetro que define o número de vezes que o algoritmo de aprendizagem trabalhará em todo o conjunto de dados de treinamento, ou seja um *epoch* é quando um conjunto de dados é passado para frente e para trás através da rede neural apenas uma única vez. O número de *epochs* é tradicionalmente grande, muitas vezes centenas ou milhares, permitindo que o algoritmo de aprendizagem seja executado até que o erro do modelo tenha sido suficientemente minimizado.

F

Features – É uma propriedade individual mensurável ou uma característica de um fenômeno que está sendo observado. Normalmente, são numéricas, mas podem ser uma característica estrutural como uma string ou vetor.

Feature learning – é a capacidade do modelo descobrir automaticamente as representações necessárias para a detecção ou classificação de características a partir de dados brutos. Isso substitui a necessidade de seleção manual das principais características e permite que uma máquina aprenda os recursos e os use para executar uma tarefa específica.

Feature map ou **Activation map** – é cada representação gerada por um filtro da camada convolucional que são empilhados, formando um tensor cuja profundidade é igual ao número de filtros e esse tensor será a entrada para a próxima camada.

G

Gaussian Process Regression - é uma técnica estatística não paramétrica (ou seja, não está limitada por uma forma funcional). Portanto, ao invés de calcular a distribuição de probabilidade dos parâmetros de uma função específica, o GPR calcula a distribuição de probabilidade para todas as funções admissíveis que se ajustam aos dados. GPR tem vários benefícios, trabalhando bem em pequenos conjuntos de dados e tem a capacidade de fornecer medições de incerteza sobre as previsões.

Global Average Pooling - é uma operação que calcula o valor médio de cada mapa de características na camada anterior. Esta operação reduz significativamente os dados e prepara o modelo para a camada totalmente conectada.

Gradiente Descendente Estocástico – É uma extensão do Gradiente Descendente que é uma técnica utilizada para reduzir uma função ao seu ponto mínimo através da derivada da função. Um problema recorrente na aprendizagem de máquinas é que grandes conjuntos de treinamento são necessários para uma boa generalização, mas grandes conjuntos de treinamento também requerem mais recursos computacionais. A ideia do gradiente descendente estocástico é que o gradiente é uma previsão. A previsão pode ser estimada aproximadamente usando um pequeno conjunto de amostras. Especificamente, em cada etapa do algoritmo, podemos amostrar um mini lote de exemplos $B = \{x, \dots, x^m\}$ retirados uniformemente do conjunto de treinamento. O tamanho do mini-lote m é tipicamente escolhido para ser um número relativamente pequeno de exemplos, variando entre 1 a algumas centenas. Crucialmente, o m é mantido fixo mesmo que o tamanho da base de dados de treino aumente. O algoritmo de otimização pode não ter a garantia de chegar a um mínimo local num período de tempo razoável, mas muitas vezes encontra um valor muito baixo da função *loss* suficientemente rápido.

Gradiente Vanishing - é uma dificuldade encontrada no treinamento de redes neurais artificiais com métodos de aprendizagem baseados em gradiente e retropropagação (*backpropagation*). Nesses métodos, cada um dos pesos das redes neurais recebe uma atualização proporcional à derivada parcial da função erro em relação ao peso atual em cada iteração de treinamento. O problema é que, em alguns casos, o gradiente será pequeno, impedindo efectivamente que o peso altere o seu valor. No pior dos casos, isto pode impedir completamente a rede neural de continuar o treinamento. A medida que mais camadas usando certas funções de ativação são adicionadas às redes neurais, os gradientes da função *loss* se aproximam de zero, tornando a rede difícil de ser treinada.

K

Kernels - Geralmente, refere-se a um operador aplicado à totalidade da imagem de tal forma que transforma a informação codificada nos pontos. Na prática, no entanto, um kernel é uma matriz de menor tamanho em comparação com as dimensões de entrada da imagem, que consiste em entradas com valor real.

K-nearest Neighbors (KNN) – KNN é um dos algoritmos de classificação mais simples. Seu objetivo é utilizar um banco de dados no qual os pontos de dados são separados em várias classes para prever a classificação de um novo ponto de amostragem. KNN não utiliza os pontos de dados de treino para fazer qualquer generalização. Em outras palavras, não há uma fase de treinamento explícita ou é muito mínima. Isto também significa que a fase de treinamento é muito rápida. KNN é baseado na semelhança de características (pode ser através da função de distância).

L

Logistic Regression – É uma técnica estatística para analisar um conjunto de dados no qual existem uma ou mais variáveis independentes que determinam um resultado. O resultado é medido com uma variável dicotômica (na qual existem apenas dois resultados possíveis). A variável dependente é binária ou dicotômica, ou seja, contém apenas dados codificados como 1 (VERDADEIRO, sucesso, gravidez, entre outros) ou 0 (FALSO, falha, não-grávida, entre outros). Em comparação com os modelos utilizados em aplicações de regressão, a regressão logística distingue-se essencialmente pelo fato de a variável resposta ser categórica. Enquanto modelo de predição para variáveis categóricas, a regressão logística é comparável com os modelos supervisionados utilizados em aprendizagem automática (árvores de decisão, redes neurais, entre outros).

Linear Regression - Em estatística, é uma abordagem linear para modelar a relação entre uma resposta escalar (ou variável dependente) e uma ou mais variáveis explicativas (ou variáveis independentes). Existem dois tipos de regressão linear: regressão linear simples e regressão linear múltipla. Na regressão linear simples, uma única variável independente é usada para prever o valor de uma variável dependente. Na regressão linear múltipla, duas ou mais variáveis independentes são usadas para prever o valor de uma variável dependente. A diferença entre as duas é o número de variáveis independentes. Em ambos os casos, há apenas uma única variável dependente.

M

Machinery Fault Simulator – é um simulador utilizado para estudar as assinaturas de falhas comuns de ativos, sem interferir em ativos instalados na área industrial. Através desses simuladores é possível provocar falhas nos componentes e coletar os dados através de sensores para realizar um estudo mais detalhado.

Multiclass Decision Jungle – É um módulo utilizado em aprendizagem de máquina baseado em algoritmo com aprendizagem supervisionada. Apresenta como vantagens: um menor consumo de memória e um melhor desempenho quando comparado com os módulos do tipo Decision Tree, e, como desvantagem apresenta a necessidade de um maior tempo para o treinamento do módulo.

Multilayer Perceptron (MLP) - É uma classe de Rede Neural Artificial Feedforward. Um MLP consiste de pelo menos três camadas de nós: uma camada de entrada, uma camada oculta e uma camada de saída. Exceto para os nós de entrada, cada nó é um neurônio que usa uma função de ativação não linear. MLP utiliza aprendizagem supervisionada chamada backpropagation para treinamento. Suas múltiplas camadas e ativação não-linear distinguem o MLP de um perceptron linear. Ele pode distinguir dados que não são linearmente separáveis. Tem a característica de camadas totalmente conectadas, em que cada perceptron é conectado com cada outro perceptron. A desvantagem é que o número de parâmetros totais crescem de acordo com a quantidade de camadas.

N

Neural Network Regression - Qualquer classe de modelos estatísticos pode ser chamada de rede neural, se ela usa pesos adaptativos e pode aproximar funções não-lineares de suas entradas. Assim, a regressão da rede neural é adequada para problemas em que um modelo de regressão mais tradicional não gera uma solução não linear. A regressão da rede neural é um modelo de aprendizagem supervisionada e, portanto, requer um conjunto de dados marcados, que inclui uma coluna de rótulos.

O

Overfitting - Refere-se a um modelo que modela os dados de treinamento muito bem. Acontece quando um modelo aprende os detalhes e o ruído nos dados de treinamento na medida em que ele impacta negativamente o desempenho do modelo em novos dados.

R

Random Forest - Em uma floresta aleatória (*random forest*), os engenheiros constroem conjuntos de árvores de decisão aleatórias (*random decision tree*) para isolar mais cuidadosamente o conhecimento da mineração de dados, com diferentes matrizes de variáveis aplicadas. Uma maneira de descrever a filosofia por trás da floresta aleatória é que, como as

árvores aleatórias têm alguma sobreposição, os engenheiros podem construir sistemas para estudar dados redundantemente com as várias árvores e procurar tendências e padrões que suportam um determinado resultado de dados. A floresta aleatória é um algoritmo de classificação que consiste em muitas árvores de decisão. Ele usa o ensacamento (*bagging*) e a aleatoriedade ao construir cada árvore individual para tentar criar uma floresta não correlacionada de árvores cuja predição por comitê é mais precisa do que a de qualquer árvore individual. É uma forma de calcular a média de múltiplas árvores de decisão profundas, treinadas em diferentes partes do mesmo conjunto de treinamento, com o objetivo de reduzir a variância.

S

Support Vector Machines (SVM) - É um algoritmo de aprendizagem de máquina supervisionado que pode ser usado para problemas de classificação ou regressão. Ele usa uma técnica chamada *kernel trick* para transformar seus dados e, com base nessas transformações, encontra um limite ótimo entre as saídas possíveis. O SVM padrão toma como entrada um conjunto de dados e prediz, para cada entrada dada, qual de duas possíveis classes a entrada faz parte, o que faz do SVM um classificador linear binário não probabilístico, encontrando uma linha de separação, mais comumente chamada de hiperplano entre dados de duas classes. Essa linha busca maximizar a distância entre os pontos mais próximos em relação a cada uma das classes.

Sigmoid - Uma função *sigmoid* é um tipo de função de ativação, e mais especificamente definida como uma função de amortecimento. As funções de amortecimento limitam a saída a um intervalo entre 0 e 1, tornando essas funções úteis na predição de probabilidades.

Uma função *sigmoid* é um tipo de função logística e refere-se puramente a qualquer função que mantém a forma "S", como $\tanh(x)$. Em que uma função *sigmoid* tradicional existe entre 0 e 1, $\tanh(x)$ segue uma forma similar, mas existe entre 1 e -1. Por si só, uma função *sigmoid* também é diferenciável, o que significa que podemos encontrar a inclinação da curva *sigmoid*, em dois pontos quaisquer.

As funções *sigmoids* são frequentemente utilizadas em Inteligência Artificial, especificamente no teste de redes neurais artificiais, como uma forma de entender a saída de um nó ou "neurônio". Por exemplo, uma rede neural pode tentar encontrar uma solução desejada dado um conjunto de entradas. A função *sigmoid* irá determinar a saída e que a saída será usada como a

entrada para o nó seguinte. Este processo irá repetir até que a solução para o problema original é encontrado.

Sparse Multiclass Cross-Entropy – É uma função utilizada para calcular o *loss*, deve ser usada quando as classes são mutuamente exclusivas e gera como resultado números inteiros. A vantagem de utilizar essa função é que consome menos memória e processamento.

Similarity-Based Modeling – É um modelo pelo qual o funcionamento normal de um sistema é modelado de modo a detectar falhas através da análise da sua semelhança com as condições normais do sistema.

T

Two-class boosted decision tree - Cria um modelo de aprendizagem de máquina baseado no algoritmo de árvores de decisão reforçadas que é um modelo de aprendizagem em conjunto no qual a segunda árvore corrige os erros da primeira árvore, a terceira árvore corrige os erros da primeira e da segunda árvores, e assim por diante. Previsões são baseadas em todo o conjunto de árvores em conjunto que faz a previsão.

W

White Gaussian Noise - É um modelo básico de ruído utilizado para imitar o efeito de muitos processos aleatórios que ocorrem na natureza.

X

XGBoost – Significa *Extreme Gradient Boosting*, é um algoritmo com base em árvore de decisão e utiliza uma estrutura de reforço de gradiente. Pode ser aplicado para resolver problemas de regressão e classificação. Por ser uma biblioteca de domínio público, diversas melhorias são implementadas e distribuídas.

APÊNDICE A – Publicação e submissão de artigos

O artigo “*Deep learning for diagnosis and classification of faults in industrial rotating machinery*” foi submetido a revista “*Computers & Industrial Engineering*”, cujo fator de impacto é 3 517 e qualis A2, está em fase de revisão, conforme recomendações dos revisores.

Foi publicado o artigo “Aprendizagem de máquina e aprendizagem profunda para classificação de falhas em equipamentos rotativos” no seminário “V Simpósio Internacional de Inovação e Tecnologia 2019 - SIINTEC”



APÊNDICE B – Depósito de patente

Em função da pesquisa realizada e o trabalho desenvolvido no mestrado, o SENAI/CIMATEC recomendou a defesa de patente da solução apresentada. Portanto, foi realizada, juntamente com uma empresa contratada pelo SENAI/CIMATEC, a patente da solução que está registrada sob nº BR 10 2020 013417 5, conforme imagem abaixo.

The screenshot shows the INPI website interface. At the top, there is a navigation bar with the Brazilian flag and the text 'BRASIL', followed by 'Acesso à informação', 'Participe', 'Serviços', 'Legislação', and 'Canais'. Below this is the INPI logo and the text 'Instituto Nacional da Propriedade Industrial' and 'Ministério da Economia'. The main heading is 'Consulta à Base de Dados do INPI' with a link '[Início | Ajuda?]'. The search results section is titled 'RESULTADO DA PESQUISA (02/07/2020 às 19:24:59)'. It shows 'Pesquisa por:' followed by 'Meus Pedidos' and a checkbox. The search criteria are 'Nº Pedido: BR 10 2020 013417 5 \'. An 'AVISO' states: 'Nº Pedido: BR 10 2020 013417 5 \ consta em nosso banco de dados. Por favor, acesse regularmente a Revista da Propriedade Industrial (RPI), a fim de acompanhar as publicações relativas ao pedido em questão.' Below the notice, there is a paragraph explaining the INPI's tracking tool and a list of four steps: 1º Retorne à página inicial da ferramenta de busca; 2º Entre com seu login e senha; 3º Busque novamente pelo processo; e 4º Clique em Meus pedidos. A note mentions that login and password are the same as for Union Collection Guides and that the email should be updated in the INPI registry. At the bottom, it says 'Dados atualizados até 30/06/2020 - Nº da Revista: 2582'. The footer includes the address 'Rua Mayrink Veiga, 9 - Centro - RJ - CEP: 20090-910' and the 'Fale conosco' logo.