



Universidade Federal da Paraíba  
Centro de Informática  
Programa de Pós-Graduação em Informática

Dissertação de Mestrado

UMA ABORDAGEM BASEADA EM PERCEPTRON  
MULTICAMADAS PARA DETECÇÃO DE FALTAS NO  
ESTATOR DE GERADORES EÓLICOS DO TIPO PMSG

Bruno Adônis de Sá

João Pessoa - PB  
31 de Janeiro de 2020

Universidade Federal da Paraíba  
Centro de Informática  
Programa de Pós-Graduação em Informática

UMA ABORDAGEM BASEADA EM PERCEPTRON  
MULTICAMADAS PARA DETECÇÃO DE FALTAS NO  
ESTATOR DE GERADORES EÓLICOS DO TIPO PMSG

Bruno Adônis de Sá

Dissertação submetida ao Programa  
de Pós-Graduação em Informática da  
Universidade Federal da Paraíba como parte  
dos requisitos necessários para obtenção do  
grau de Mestre em Informática.

Área de Concentração: Sistemas de Computação  
Linha de Pesquisa: Computação Distribuída

Orientador: Dr. Claurton de Albuquerque Siebra  
Coorientadora: Dra. Camila Mara Vital Barros

João Pessoa - PB  
31 de Janeiro de 2020

**Catálogo na publicação**  
**Seção de Catalogação e Classificação**

Silla Sá, Bruno Adônis de.

UMA ABORDAGEM BASEADA EM PERCEPTRON MULTICAMADAS PARA  
DETECÇÃO DE FALTAS NO ESTATOR DE GERADORES EÓLICOS DO  
TIPO PMSG / Bruno Adônis de Sá. - João Pessoa, 2020.  
78 f. : il.

Orientação: Claurton de A Siebra.

Coorientação: Camila M V Barros.

Dissertação (Mestrado) - UFPB/CI.

1. Aprendizagem de Máquina. 2. Detecção de Falhas no Estator. 3. Energia Eólica. 4. Gerador Síncrono de Ímã Permanente. 5. Perceptron Multicamadas. I. Siebra, Claurton de A. II. Barros, Camila M V. III. Título.

UFPB/BC



UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



Ata da Sessão Pública de Defesa de Dissertação de Mestrado de Bruno Adonis de Sá, candidato ao título de Mestre em Informática na Área de Sistemas de Computação, realizada em 31 de janeiro de 2020.

1 Aos trinta e um dias do mês de janeiro, do ano de dois mil e vinte, às dez horas, no Centro  
2 de Informática da Universidade Federal da Paraíba, em Mangabeira, reuniram-se os  
3 membros da Banca Examinadora constituída para julgar o Trabalho Final do Sr. Bruno  
4 Adonis de Sá, vinculado a esta Universidade sob a matrícula nº 20181000653, candidato ao  
5 grau de Mestre em Informática, na área de "Sistemas de Computação", na linha de pesquisa  
6 "Computação Distribuída", do Programa de Pós-Graduação em Informática, da Universidade  
7 Federal da Paraíba. A comissão examinadora foi composta pelos professores: Claurton de  
8 Albuquerque Siebra (PPGI-UFPB), Orientador e Presidente da Banca, Camila Mara Vital  
9 Barros (UFPB), Examinadora Externa ao Programa, Álvaro de Medeiros Maciel (IFPB),  
10 Examinador Externo à Instituição, Luciano Sales Barros (UFRN), Externo à Instituição.  
11 Dando início aos trabalhos, o Presidente da Banca cumprimentou os presentes, comunicou  
12 aos mesmos a finalidade da reunião e passou a palavra ao candidato para que o mesmo  
13 fizesse a exposição oral do trabalho de dissertação intitulado: "Uma Abordagem Baseada em  
14 Perceptron Multicamadas para Detecção de Falhas no Estator de Geradores Eólicos do Tipo  
15 PMSG". Concluída a exposição, o candidato foi arguido pela Banca Examinadora que emitiu  
16 o seguinte parecer: "aprovado". Do ocorrido, eu, Ruy Alberto Pisani Altafim, Coordenador do  
17 Programa de Pós-Graduação em Informática, lavrei a presente ata que vai assinada por mim  
18 e pelos membros da banca examinadora. João Pessoa, 31 de janeiro de 2020.

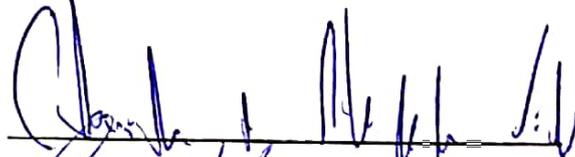
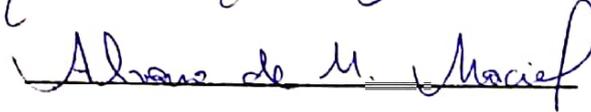
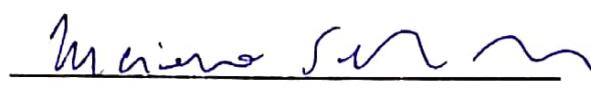
  
Prof. Dr. Ruy Alberto Pisani Altafim

Prof. Claurton de Albuquerque Siebra  
Orientador (PPGI-UFPB)

Profª. Camila Mara Vital Barros  
Examinadora Externa ao Programa (UFPB)

Prof. Álvaro de Medeiros Maciel  
Examinador Externo à Instituição (IFPB)

Prof. Luciano Sales Barros  
Examinador Externo à Instituição (UFRN)

# Dedicatória

*Dedico este trabalho primeiramente a Deus, por ser essencial em minha vida.*

*Ao meu pai, José de Sá (in memoriam) que apesar de não ter estudo formal, sempre me incentivou a estudar e conquistar patamares maiores que o dele. À minha mãe, Maria Domingas pelo suporte e dedicação, sempre me incentivando a ir atrás dos meus objetivos.*

*À minha avó paterna, Maria de Sá (in memoriam) e sua irmã Adelina de Sá (in memoriam) que faleceram durante este curso.*

*À minha noiva, Ana Karolyne pelo suporte, amor e carinho, que não mediu esforços para que eu chegasse até aqui.*

# Agradecimentos

Agradeço ao meu orientador e amigo Claurton pelo suporte e acompanhamento durante toda a minha carreira academia, projetos de iniciação científica, trabalho de conclusão de curso, as orientações extras-curriculares e este curso.

Aos meus orientadores Camila e Luciano pelo suporte e compartilhamento de conhecimento durante este curso. Agradeço também pela disponibilidade do dados, dados estes que viabilizaram esta pesquisa.

Também agradeço a Álvaro pela a participação e colaboração com esta pesquisa durante a defesa, abrindo novos horizontes.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

# Resumo

Os geradores eólicos apresentam interrupções de operação recorrentes devido à ocorrência de falhas internas. Falhas internas são de difícil detecção e que podem conduzir, silenciosamente, a danos na máquina e podem ocorrer entre as espiras, sendo denominadas espira-espira, ou entre espiras e a carcaça da máquina, sendo denominadas espira-terra. Assim, estas plantas devem ser constantemente monitoradas para que essas falhas sejam detectadas em seu estágio inicial. Essa detecção precoce possibilita a redução do custo de manutenção, ao mesmo tempo em que diminui o tempo de inatividade das turbinas eólicas. Este trabalho propõe uma estratégia para detectar falhas no estator em seu estágio inicial de forma não invasiva por meio de um módulo classificador que analisa os padrões da corrente do estator. Este classificador baseia-se em uma rede neural do tipo perceptron multicamadas (do inglês, *Multilayer Perceptron* - MLP), que foi treinada utilizando um conjunto de dados de instâncias geradas por meio de um modelo matemático de uma turbina eólica baseada no PMSG. Os resultados mostram que o classificador MLP é capaz de detectar o problema proposto com 97,62% de acurácia global. Além disso, a detecção foi alcançada em um estágio inicial, de 1% a 4% de espiras defeituosas com acurácia de 100%, contribuindo para uma detecção contínua e não invasiva de falhas internas ao estator da turbina eólica.

**Palavras-chave:** Aprendizagem de Máquina, Detecção de Falhas no Estator, Energia Eólica, Gerador Síncrono de Ímã Permanente, Perceptron Multicamadas.

# Abstract

Wind generators have recurring operating interruptions due to internal failures. Internal failures are difficult to detect and may silently lead to machine damages since they occur between turns, being named turn-to-turn, or between turn and machine housing, being named turn-to-ground. Thus, these plants must be constantly monitored so that these faults are detected in their initial stage. This early detection makes it possible to reduce maintenance costs while decreasing wind turbine downtime. This work proposes a strategy for noninvasive detecting stator failures in its initial stage through a classifier module that analyzes the stator current patterns. This classifier is based on a Multilayer Perceptron (MLP), that is a class of feedforward artificial neural network (ANN), which was trained using a dataset generated by a mathematical model of the PMSG-based wind turbine. The results show that the MLP classifier is able to detect the proposed problem with 97.62% global accuracy. In addition, detection was performed at a initial stage of 1% to 4% of faulty turns with 100% accuracy, contributing to continuous and noninvasive detection of internal wind turbine stator faults.

**Keywords:** Machine learning, Permanent Magnet Synchronous Generator, Stator Fault Detection, Wind Energy.

# Lista de Figuras

Figura 2.1 – Esquema simplificado de uma turbina eólica. . . . .	18
Figura 2.2 – Representação da turbina eólica com DFIG. . . . .	19
Figura 2.3 – Representação da turbina eólica com PMSG. . . . .	19
Figura 2.4 – Origem hierárquica das falhas em Turbinas Eólicas. . . . .	21
Figura 2.5 – Representatividade das falhas em Turbinas Eólicas. . . . .	21
Figura 2.6 – Fluxograma de diagnóstico de falhas. . . . .	23
Figura 3.1 – Diagrama de blocos do SNC. . . . .	25
Figura 3.2 – Abstração de um neurônio e suas respectivas partes. . . . .	25
Figura 3.3 – Modelo de um neurônio. . . . .	26
Figura 3.4 – Funções de ativação e sua derivada. . . . .	28
Figura 3.5 – Ilustração de uma MLP de 3 camadas. . . . .	29
Figura 3.6 – Entrada da MLP. . . . .	29
Figura 3.7 – Camada oculta da MLP. . . . .	30
Figura 3.8 – Neurônio da camada oculta da MLP. . . . .	30
Figura 3.9 – Diagrama em blocos de aprendizado supervisionado. . . . .	31
Figura 3.10–Ilustração do processo de validação cruzada <i>10-Fold</i> . . . . .	37
Figura 3.11–Ilustração do processo de validação cruzada <i>3-Fold</i> . . . . .	37
Figura 5.1 – Diagrama de blocos do plano de atividades. . . . .	44
Figura 5.2 – Ilustração do estador do PMSG para simulação das falhas: (a) espira-terra e (b) espira-espira. . . . .	46
Figura 5.4 – Esquema de alimentação do modelo. . . . .	48
Figura 5.3 – Diagrama esquemático do circuito para geração de falhas. . . . .	48
Figura 5.5 – Tipos de falhas internas . . . . .	48
Figura 5.6 – Exemplo de amostra normal e defeito 1 a 1% e 2%. . . . .	50
Figura 5.7 – Interpretação do <i>boxplot</i> genérico. . . . .	51
Figura 5.8 – Boxplot de amostras das classes normal (0% defeito) e 1% a 3% dos defeitos 1 e 2. . . . .	51
Figura 5.9 – Histograma de amostras das classes normal e 1% a 3% dos defeitos 1 e 2. . . . .	52
Figura 5.10–Métricas extraídas a partir da média das amostras por classe: média, desvio padrão, mínimo, máximo, primeiro quartil, mediana e terceiro quartil. . . . .	53
Figura 5.11–Correlação das métricas extraídas a partir da média das amostras por classe. . . . .	54
Figura 5.12–Correlação das métricas selecionadas por eliminação daquelas com forte correlação. . . . .	55
Figura 5.13–Métricas selecionadas em relação as classes analisadas. . . . .	55

Figura 5.14–Arquitetura da MLP encontrada. . . . .	57
Figura 6.1 – Representação dos casos desenvolvidos. . . . .	60
Figura 6.2 – Matriz de confusão para a frequência de chaveamento de 5 kHz. . . . .	61
Figura 6.3 – Curva de acurácia do treinamento e validação para a frequência de chaveamento de 5 kHz. . . . .	62
Figura 6.4 – Curva de erro para a frequência de chaveamento de 5 kHz. . . . .	62
Figura 6.5 – Matriz de confusão para a frequência de chaveamento de 2,5 kHz. . . . .	63
Figura 6.6 – Curva de acurácia do treinamento e validação para a frequência de chaveamento de 2,5 kHz. . . . .	63
Figura 6.7 – Curva de erro para a frequência de chaveamento de 2,5 kHz. . . . .	64
Figura 6.8 – Matriz de confusão para as frequências de chaveamento de 2,5 kHz e 5 kHz. . . . .	65
Figura 6.9 – Curva de acurácia do treinamento e validação para as frequências de chaveamento de 2,5 kHz e 5 kHz. . . . .	65
Figura 6.10–Curva de erro para as frequências de chaveamento de 2,5 kHz e 5 kHz. . . . .	65

# Lista de tabelas

Tabela 3.1 – Matriz de Confusão . . . . .	38
Tabela 3.2 – Exemplo da matriz de confusão da Classe B. . . . .	38
Tabela 5.1 – Desvio padrão médio de cada uma das classes e sua primeira diferença. . . . .	56
Tabela 6.1 – Resultados obtidos para a frequência de chaveamento de 5 kHz. . . . .	61
Tabela 6.2 – Resultados obtidos para a frequência de chaveamento de 2,5 kHz. . . . .	63
Tabela 6.3 – Resultados obtidos para as frequências de chaveamento de 2,5 kHz e 5 kHz. . . . .	64

# Lista de abreviaturas e siglas

ADAM	<i>A Method For Stochastic Optimization</i>
ANEEL	Agência Nacional de Energia Elétrica
CNN	<i>Convolutional Neural Network</i>
DFIG	<i>Double-Fed Induction Generator</i>
DPM	desvio padrão médio
EESG	<i>Electrically Excited Synchronous Generator</i>
GWEC	<i>Global Wind Energy Council</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
LBFSG	<i>Limited-memory Broyden–Fletcher–Goldfarb–Shanno</i>
MI	<i>Mutual Information Algorithm</i>
MCSA	<i>Current Signal Analysis</i>
MSCSA	<i>motor square current signature analysis</i>
MLP	<i>Multilayer Perceptrons</i>
PES	<i>Power and Energy Society</i>
PMSG	<i>Permanent Magnet Synchronous Generator</i>
RAW	Dados primários, não tratados
RNA	Redes Neurais Artificiais
SCSA	<i>Stator Current Signature Analysis</i>
SNC	Sistema Nervoso Central
SPAM	<i>Stupid Pointless Annoying Messages</i>
STD	<i>Standard Deviation</i>
SVM	<i>Support Vector Machine</i>

# Sumário

<b>1</b>	<b>Introdução</b>	<b>14</b>
1.1	Contextualização	14
1.2	Motivação	15
1.3	Objetivo Geral	16
1.4	Objetivos Específicos	16
1.5	Estrutura da Proposta de Dissertação	16
<b>2</b>	<b>Fundamentação Teórica: Turbinas Eólicas</b>	<b>17</b>
2.1	Introdução	17
2.2	Turbina Eólica	17
2.3	Gerador Síncrono com Ímã Permanente	20
2.4	Falhas em Turbinas Eólicas	20
2.5	Considerações do Capítulo	23
<b>3</b>	<b>Fundamentação Teórica: Redes Neurais Artificiais</b>	<b>24</b>
3.1	Introdução	24
3.1.1	O que é uma Rede Neural?	24
3.2	Fundamentos Biológicos	25
3.2.1	Cérebro Humano	25
3.2.2	Modelo de um Neurônio Artificial	26
3.3	<i>Perceptron</i> de Múltiplas Camadas	28
3.3.1	Algoritmo de Retropropagação de Erro	32
3.4	Treinamento	34
3.5	Critério de Parada	35
3.6	Validação do Módulo Classificador	35
3.7	Matriz de Confusão	37
3.8	Considerações do Capítulo	39
<b>4</b>	<b>Revisão da Literatura</b>	<b>41</b>
4.1	Artigos Relacionados	41
4.2	Considerações do Capítulo	43
<b>5</b>	<b>Metodologia</b>	<b>44</b>
5.1	Plano de Atividades Desenvolvido	44
5.2	Modelo para Geração de Falhas	45
5.3	Criação da Base de Dados	48
5.4	Análise Exploratória dos Dados	49
5.4.1	Segmentação e Rotulação	49
5.4.2	Sumarização dos Dados por Rótulo	52
5.4.3	Correlação entre as Métricas Extraídas	53

5.5	Criação do Módulo Classificador . . . . .	56
5.6	Validação do Módulo Classificador . . . . .	59
5.7	Considerações do Capítulo . . . . .	59
<b>6</b>	<b>Resultados . . . . .</b>	<b>60</b>
6.1	Frequência de Chaveamento de 5 kHz . . . . .	61
6.2	Frequência de Chaveamento de 2,5 kHz . . . . .	62
6.3	Frequências de Chaveamento de 2,5 kHz e 5 kHz . . . . .	64
6.4	Considerações do Capítulo . . . . .	66
<b>7</b>	<b>Conclusão . . . . .</b>	<b>67</b>
7.1	Contribuições . . . . .	67
7.2	Limitações . . . . .	67
7.3	Trabalhos Futuros . . . . .	69
	<b>Referências . . . . .</b>	<b>70</b>

# 1 Introdução

## 1.1 Contextualização

Nos últimos anos, a energia eólica tem desempenhado um papel significativo no campo das energias renováveis, em resposta ao aumento da demanda mundial por energia elétrica [1]. No Brasil, não foi diferente e em 2018 o país já ocupava a quinta posição no ranking mundial com a capacidade instalada de 14,7 GW, ficando atrás da China (211,39 GW), Estados Unidos (96,66 GW), Alemanha (59,56 GW) e Índia (35,12 GW) [2]. Se considerar a potência instalada por mil habitantes, temos Alemanha (718,21 kW), USA (295,46 kW), China (151,78 kW), Brasil (70,21 kW) e Índia (25,97 kW) [2, 3]. Em Janeiro de 2020 o Brasil já tem uma capacidade instalada superior a 15,3 GW distribuídos por mais de 629 parques eólicos, sendo que essa capacidade instalada corresponde à 9,04% da potência elétrica instalada no país, ficando atrás apenas de Usina Hidrelétrica e Termelétrica [4].

As turbinas eólicas têm um custo de manutenção relativamente alto [5] e estão localizadas em áreas remotas e expostas a ambientes hostis experimentando mudanças extremas de temperatura e umidade [6]. Um exemplo é o *Hornsea Project One* [7], que será o maior parque eólico do mundo no mar, localizado a uma distância de 120 km da costa de Yorkshire na Inglaterra, o qual alcançará 1,2 GW, ocupando uma área de 407 km<sup>2</sup> com 174 turbinas. Com isso, diversos equipamentos da turbina podem apresentar algum tipo de distúrbio.

Entre os componentes presentes em turbinas eólicas, cinco deles são responsáveis por 86% do tempo de inatividade das turbinas eólicas [8]: a rede elétrica, o gerador, a caixa de engrenagens, o sistema de controle e o sistema hidráulico. O custo de manutenção causado pela caixa de engrenagens, por exemplo, corresponde a 13% do custo total de uma turbina eólica [9]. Um diagnóstico precoce poderia minimizar essas falhas, reduzindo interrupções de produção não programadas e também os custos de manutenção [10]. Quanto ao gerador, uma das principais causas dos problemas no gerador das máquinas trifásicas são as falhas nas espiras do enrolamento do estator, denominadas faltas internas, que responde por 20% a 40% do número total de falhas [11, 12, 13, 14, 15].

As faltas internas no gerador em sua fase incipiente degradam o desempenho da máquina podendo levar a falhas mais severas, como o rompimento total do seu isolamento [16], o que resultaria em perdas financeiras significativas [17, 18, 19, 16]. Com relação ao monitoramento, os sensores de monitoramento das turbinas eólicas, tais como sensor de temperatura e vibração, só são capazes de detectar faltas internas quando estas já estão

em estados mais avançados de degradação, afetando diretamente o desempenho da turbina. Distúrbios internos ao gerador são difíceis de detectar, pois esta detecção só é possível a partir de uma avaliação minuciosa nas correntes do estador do gerador.

Diversas técnicas podem ser utilizadas para identificar falhas incipientes: o sinal de saída do sistema em análise pode ser comparado com um sinal de referência, como é apresentado em [20]; ou o teste de surto pode ser aplicado para a detecção estática de um curto entre as bobinas, como é apresentado em [21]. No entanto, os resultados gerados por essas técnicas também geram algumas dúvidas na tomada de decisão. A comparação com sinal de referência depende da qualidade da referência selecionada e do sinal analisado, sendo que deformidades de aquisição do sinal analisado podem resultar em falsos positivos. No caso do teste de surto, trata-se de um método bastante invasivo e, dependendo da gravidade do defeito, este teste pode agravar a situação ou até mesmo levar ao rompimento completo do isolamento [22]. O uso de uma técnica de monitoramento online, contínua e não invasiva pode tornar a detecção de falhas mais segura [23].

Um sistema de diagnóstico de falhas é definido a partir de uma perspectiva de reconhecimento de padrões que normalmente consiste em três etapas gerais: aquisição de dados, extração de características e classificação. Das três etapas, a extração de características é a mais crítica. Seu objetivo é obter informações que permitam caracterizar o estado da máquina servindo de entrada para o classificador. Porém este processo não é trivial para faltas incipientes internas ao gerador, devido à proximidade do estado de falta incipiente com o estado de funcionamento normal. Alguns autores como [6], já vêm utilizando o Aprendizado Profundo (do inglês, *Deep Learning*) com objetivo de modelar representações de dados de alto nível e classificar padrões através de múltiplas camadas empilhadas de unidades de processamento de informações não-lineares em uma arquitetura hierárquica, redes neurais artificiais [24]. Ademais, o sistema de aprendizado profundo com a arquitetura correta pode aprender automaticamente características discriminativas a partir do conjunto de dados, permitindo que o classificador tome a melhor decisão possível no processo de classificação.

## 1.2 Motivação

Melhorar a confiabilidade das turbinas eólicas é crucial para reduzir o custo da energia eólica [25, 26, 27, 28]. Devido à crescente expansão e ao alto custo de implementação dos parques eólicos, se fazem necessárias medidas de monitoramento e prevenção de falhas incipientes para manter o pleno funcionamento das turbinas. Assim, o diagnóstico precoce identifica essas falhas, reduzindo interrupções de produção não programadas e custos de manutenção [10].

Dentre os geradores eólicos, o Gerador Síncrono de Ímã Permanente (do inglês, *Permanent Magnet Synchronous Generator* - PMSG) teve um aumento relevante nos últimos anos e, especialmente no mercado asiático, devido à alta demanda energética e espera-se que essa tendência continue nos anos seguintes. O PMSG além de não requerer caixa de engrenagens, que reduz o número de componentes que podem apresentar problemas. Este também pode ser operado em toda a faixa de velocidade, provendo controle completo da potência ativa. Assim, os PMSGs são mais eficientes que os Geradores de Indução Duplamente Alimentados (do inglês, *Double Fed Induction Generation* - DFIG) e Gerador Síncrono Eletricamente Excitado (do inglês, *Electrically Excited Synchronous Generator* - EESG) tradicionais. Na Ásia de 2005 a 2013 o PMSG passou de 0% da capacidade total instalada para algo próximo de 40%, enquanto o DFIG se mantém com a parcela restante do mercado [29].

### 1.3 Objetivo Geral

Detectar faltas internas em máquinas elétricas de forma não invasiva. Pretende-se utilizar Aprendizagem de Máquina para detectar se há ou não um defeito e classificar em que estágio de evolução da severidade da falta.

### 1.4 Objetivos Específicos

Gerar artificialmente uma base de dados a partir de amostras de corrente advindas do PMSG em diferentes estados de funcionamento. Em posse do conjunto de dados, também será definida uma arquitetura de rede neural artificial com seus respectivos parâmetros que seja capaz de detectar e classificar em que estágio de evolução se encontra o defeito analisado, espira-espira e espira-terra.

### 1.5 Estrutura da Proposta de Dissertação

O restante da proposta está organizada da seguinte maneira: No Capítulo 2 é apresentada a fundamentação teórica de turbinas eólicas. No Capítulo 3 é apresentada a fundamentação teórica da RNA, detalhando a arquitetura do tipo MLP. No Capítulo 4 é apresentado o resumo de alguns trabalhos recentes relacionados à detecção e classificação de faltas em geradores eólicos. No Capítulo 5 são apresentados os métodos utilizados no desenvolvimento desta pesquisa. No Capítulo 6 são apresentados os resultados obtidos e as considerações finais. No Capítulo 7 é apresentada a conclusão da pesquisa, suas limitações e trabalhos futuros.

## 2 Fundamentação Teórica: Turbinas Eólicas

Neste capítulo são apresentados a fundamentação teórica sobre turbinas eólicas, componentes de uma turbina eólica e uma introdução às falhas que podem ocorrer em seus componentes.

### 2.1 Introdução

Em resposta ao aumento de demanda mundial de energia e também devido ao apelo político e ambiental em todo o mundo, tem-se investido cada vez mais em fontes alternativas de energia limpa e a turbina eólica é a que mais teve crescimento nos últimos anos [1]. A turbina eólica é responsável por captar e transformar energia cinética do vento em energia mecânica por meio do movimento de rotação das pás e, através de um gerador, transformar energia mecânica em energia elétrica que posteriormente é repassada à rede elétrica.

### 2.2 Turbina Eólica

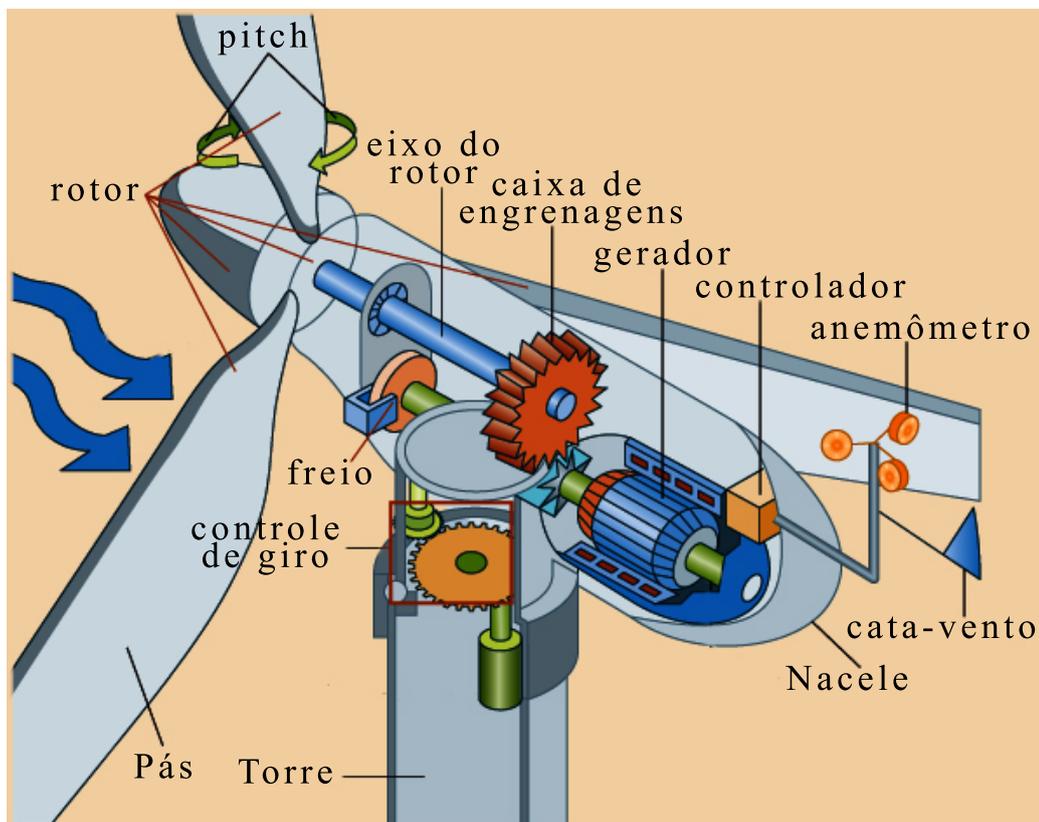
Na Figura 2.1 é ilustrado um esquema simplificado de uma turbina eólica. Os componentes principais de uma turbina eólica são:

- Pás: o conjunto de pás, na maioria dos casos igual a três, sendo denominado de rotor. As pás devem conciliar diversas características, dentre as quais se tem: leveza, rugosidade e um bom rendimento aerodinâmico. Pode possuir forma retangular ou trapezoidal, sendo esta última a mais utilizada por apresentar uma aerodinâmica mais eficiente [30]. Seu comprimento pode variar proporcionalmente à potência gerada;
- Eixo do rotor: é o componente responsável por transmitir o torque e as vibrações do rotor para a caixa de engrenagens ou para o gerador elétrico, no caso de modelos sem a caixa de engrenagens;
- Caixa de engrenagens: conjunto de engrenagens que realiza a interligação entre a baixa rotação das pás e a alta rotação do gerador elétrico, realizando uma transformação de velocidade. Em alguns casos, turbinas eólicas dispensam o uso de engrenagens através do uso de geradores elétricos com elevado número de pares de pólos;
- Gerador elétrico: elemento responsável pela geração de energia elétrica. Para unidades de média e alta potência são utilizadas máquinas de indução com rotor bobinado e,

para baixa potência, são empregadas máquinas síncronas com excitação via ímãs permanentes;

- Controlador: contém um computador que monitora continuamente a condição da turbina eólica e controla o mecanismo de guinada. Em caso de qualquer mau funcionamento (por exemplo, sobreaquecimento da caixa de velocidades ou do gerador), a turbina eólica pode parar automaticamente e enviar uma chamada para o operador da turbina através de algum link telefônico ou rede sem fio;
- Nacele: é a cabine responsável por abrigar e proteger a caixa de engrenagens, o gerador e parte das unidades de controle;
- Torre: constitui o elemento de sustentação da Nacele. Deve ter altura elevada para reduzir a probabilidade de turbulências, mais comuns próximas ao solo.

Figura 2.1 – Esquema simplificado de uma turbina eólica.



Fonte: Adaptada de [31].

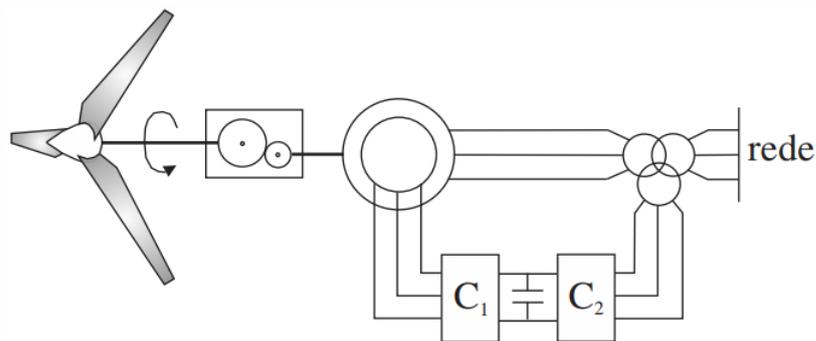
Dentre os tipos de aerogeradores utilizados em parques eólicos, tem-se os de velocidade fixa e os de velocidade variável. Aerogeradores de velocidade fixa apresentam estrutura mais simples e robusta, enquanto os de velocidade variável apresentam melhor desempenho com relação ao aproveitamento da energia do vento em diferentes velocidades, sendo esta forma o tipo mais comum nos parques eólicos atuais [32].

As vantagens da operação de turbinas com velocidades variável são [33]:

- A redução dos estresses mecânicos causados pelas variações de velocidade do vento, que podem ser absorvidas pela inércia mecânica da turbina eólica;
- O aumento da potência extraída do vento proporcionado pelo ajuste contínuo da velocidade rotacional da turbina eólica, conforme a velocidade do vento, de maneira que a potência extraída pela turbina seja maximizada;
- A redução de ruídos acústicos, tornando possível a operação em baixas velocidades com baixa potência gerada.

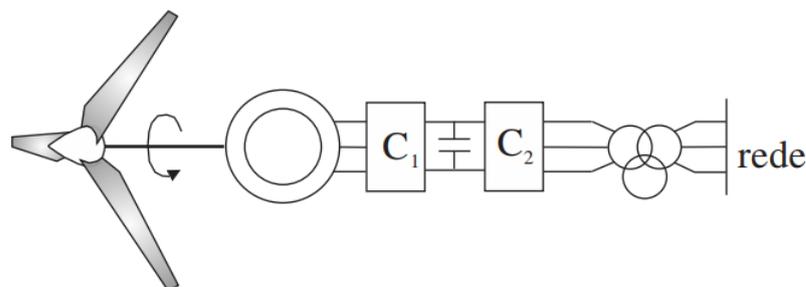
Os tipos de tecnologia em geradores amplamente utilizados para turbinas eólicas de velocidade variável são o DFIG [30], ilustrado na Figura 2.2 e o PMSG, apresentado na Figura 2.3. O DFIG utiliza o gerador de indução com rotor bobinado, onde o rotor da máquina é ligado à rede por meio de conversores e o estator é diretamente ligado à rede. Já no PMSG, o gerador pode ser de múltiplos pólos e dispensar o uso da caixa de engrenagens. Além disto, o estator da máquina é ligado à rede através de conversores [34].

Figura 2.2 – Representação da turbina eólica com DFIG.



Fonte: [30].

Figura 2.3 – Representação da turbina eólica com PMSG.



Fonte: [30].

## 2.3 Gerador Síncrono com Ímã Permanente

Os geradores síncronos utilizados na geração eólica são o PMSG e EESG. A topologia mais comum é o PMSG, que não possui necessidade de circuito de excitação para o campo magnético e apresenta uma vantagem na geração eólica por possibilitar injeção direta, onde o eixo da turbina é conectado diretamente ao eixo do rotor, dispensando a caixa de engrenagens. O PMSG utiliza um elevado número de polos para compensar a baixa velocidade com que opera, uma vez que está diretamente conectado ao rotor da turbina, cujas pás, tipicamente, giram a velocidade de 30 a 50 rev/min [30]. A não utilização da caixa de engrenagens apresenta vantagens como: uma melhor eficiência, redução do peso da turbina, de níveis de ruído e de custos associados à manutenção regular deste componente [35]. O rotor do gerador elétrico possui excitação com ímãs permanentes que operam em velocidade variável através do controle das pás, ilustrado na Figura 2.1, ajustando os ângulos das pás da turbina (do inglês, *pitch*).

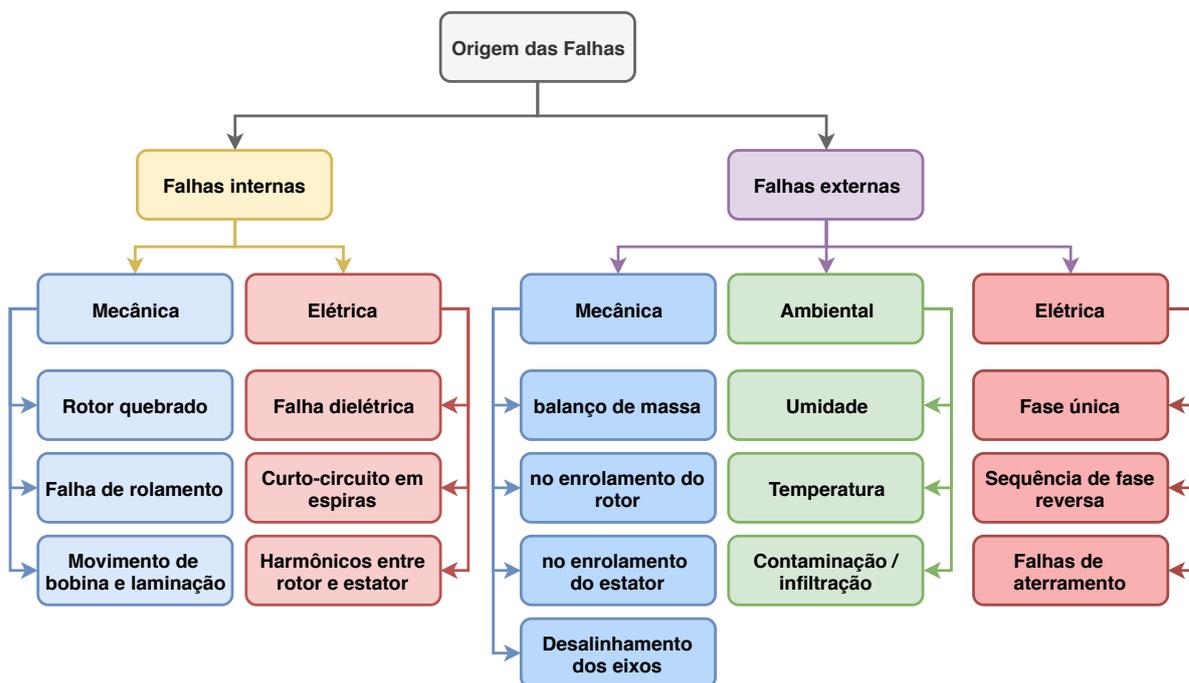
Especificamente, em relação aos sistemas de conversão de energia eólica com PMSG, suas vantagens podem ser sumarizadas por [36, 29]:

- Elevada eficiência e rendimento;
- Não é necessária fonte externa para excitação do campo magnético;
- Aprimoramento nas características térmicas do gerador a ímãs permanentes devido à ausência das perdas do campo;
- Elevada robustez em virtude da ausência de componentes mecânicos, por exemplo, anéis coletores e caixa de engrenagem;
- Controle da potência ativa e reativa;
- Controle total da amplitude e frequência da tensão;
- Leve, o que lhe garante uma elevada potência em relação ao seu peso.

## 2.4 Falhas em Turbinas Eólicas

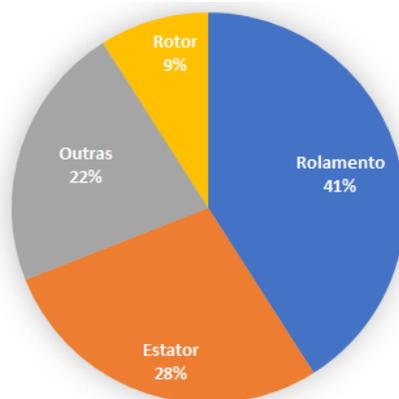
Em sua grande maioria, motores elétricos com o passar do tempo podem apresentar algum tipo de falha. Com os geradores de turbinas eólicas não é diferente. As falhas são usualmente classificadas em dois grandes grupos: falhas mecânicas e falhas elétricas. A Figura 2.4 elenca algumas das falhas mais recorrentes e sua hierarquia quanto à origem.

Figura 2.4 – Origem hierárquica das falhas em Turbinas Eólicas.



Fonte: Adaptada de [37].

Figura 2.5 – Representatividade das falhas em Turbinas Eólicas.



Fonte: Adaptada de [38].

A Figura 2.5 apresenta as falhas mais recorrentes e sua representatividade em relação ao total de falhas relatadas. Falhas no rolamento representam 41% do total de falhas, 28% para falhas no estator, 9% para falhas no rotor e 22% para as demais falhas. O curto-circuito em espiras, como apresentado em 2.4 é uma falha elétrica e interna.

As falhas no rolamento têm suas origens associadas a defeitos de fabricação, sobrecarga, lubrificação inadequada, desgaste de detritos [39, 40]. Enquanto as falhas no estator têm suas origens associadas à falha na instalação, sobretensão no fornecimento,

folga da bobina [41]. Já as falhas no rotor têm suas origens associadas ao desalinhamento interno do rotor, defeito de fabricação e movimento do anel [42].

Para o monitoramento de diferentes tipos de falhas, são utilizadas diferentes fontes de informação para obter características relevantes que podem ser aplicadas no diagnóstico e detecção de falhas. Especialmente análise dos sinais de corrente do motor (do inglês, *Current Signal Analysis* - MCSA), para detecção de falhas internas aos geradores.

Normalmente as metodologias que usam MCSA focam seus esforços na análise espectral dos sinais. Este método emprega os resultados da análise espectral da corrente instantânea do estator como característica para detecção e classificação das falhas [43]. Além de MCSA, são aplicados sensores de vibração como fonte de informação para detecção e diagnóstico de defeitos de origem mecânica [44, 45]. Para falhas internas ao estator, MCSA ainda é a técnica não invasiva mais útil para identificar esses problemas de forma contínua [46, 47, 48, 49]. Também têm sido utilizadas imagens térmicas do estator [41].

Em [43] são apresentados casos experimentais empregando a análise de assinatura de corrente quadrada do motor (do inglês, *motor square current signature analysis* - MSCSA). Este método é uma variante do MCSA. O autor utilizou análise espectral de frequências do MSCSA para detectar falhas de curto-circuito entre as espiras do estator do motor de indução.

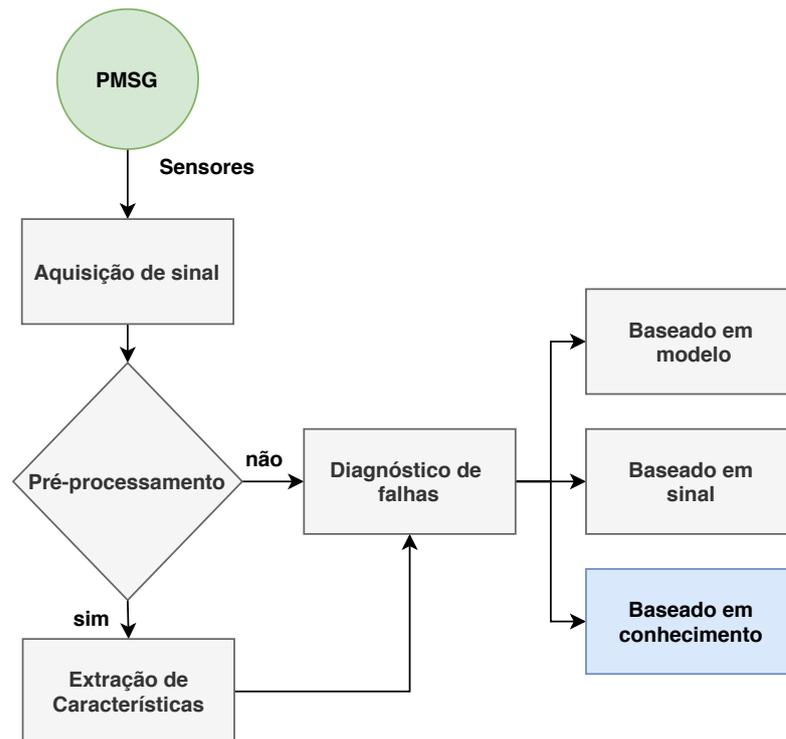
Já em [44] utilizou-se a análise espectral de frequências a partir de sinais acústicos para detecção de falhas no rolamento. Concluindo que diferentes condições físicas de funcionamento resultam em diferentes sinais acústicos com diferentes espectros de potência. Em [45] também foram utilizados sinais acústicos juntamente com sinais de vibração para monitoramento da condição dos rolamentos, o método de classificação aplicado foi a análise de forma de onda e espectro.

Na Figura 2.6 é exemplificado o fluxograma de diagnóstico de falhas. A partir da fonte de informação, neste caso o PMSG, os sensores realizam a aquisição dos dados de interesse. Em posse destes dados é realizado ou não um pré-processamento para extração de características. Por fim é realizado o diagnóstico, que podem ser divididos em três categorias: baseados em modelo [50], baseados em sinal [50] e baseados em conhecimento [51].

Os métodos baseados em modelo [50] são normalmente caracterizados por um modelo matemático do motor que emite um sinal de referência e, comparando os sinais de saída do motor com o sinal de referência, pode-se determinar o tipo de falha. Já os métodos baseados em sinais [50] são mais comumente usados para processar e extrair diferentes características dos sinais do motor, e essas informações geradas são utilizadas

para diagnósticos manuais usando informação prévia ou métodos baseado em conhecimento. Por fim, temos os métodos baseados em conhecimento [51] que podem determinar automaticamente o estado do motor e da falha com base no conhecimento especializado, que pode ter sido adquirido a partir dos dados brutos (*Deep Learning*) ou a partir das características extraídas (*Machine Learning*).

Figura 2.6 – Fluxograma de diagnóstico de falhas.



## 2.5 Considerações do Capítulo

Nesse Capítulo foram apresentados alguns dos componentes básicos da turbina eólica, as vantagens de turbinas com velocidade variável e um breve comparativo entre os dois modelos mais comuns, PMSG e DFIG. Em seguida foram apresentados mais detalhes do PMSG, as origens das falhas mais comuns e a representatividade das falhas mais recorrentes no rolamento, no estator e rotor de geradores. É possível notar também que os pesquisadores estão dando preferência às técnicas de detecção de falhas não-invasivas à máquina, isto é, aquelas que não aplicam nenhum tipo de excitação ao gerador com objetivo de detectar defeitos, como é o caso da técnica de surto [21].

No Capítulo seguinte será apresentada a fundamentação teórica da RNA, sendo esta a ferramenta utilizada para detectar as falhas internas do gerador.

# 3 Fundamentação Teórica: Redes Neurais Artificiais

Este Capítulo tem como objetivo fornecer embasamento para o entendimento das Redes Neurais Artificiais (RNAs). Dada a tamanha heterogeneidade e complexidade das RNAs, foram utilizados exemplos da própria literatura referenciada, objetivando facilitar o entendimento acerca do funcionamento das RNAs, entradas, saídas, processo de aprendizagem, entre outros termos chaves necessários para a compreensão.

## 3.1 Introdução

Em [52, Cap. 1] o estudo das redes neurais artificiais tem sido motivado, desde o começo, pelo reconhecimento de que o cérebro humano processa informações de uma forma inteiramente diferente do computador digital convencional. Entende-se por computador convencional, computadores desprovidos de algoritmos de inteligência artificial. O cérebro é um computador, cujo sistema de processamento de informação é altamente complexo, não-linear e paralelo que tem a capacidade de auto-organizar seus neurônios para realizar certos processamentos rotineiros, como por exemplo, o reconhecimento de padrões, percepção, reconhecimento de um rosto familiar inserido em um contexto não familiar, sendo essa reorganização de padrões efetuada em questão de milissegundos.

### 3.1.1 O que é uma Rede Neural?

As redes neurais são compostas de unidades de processamento chamadas de neurônios, os quais são mutuamente interligados. Como apresentado por [52, Cap 1], uma rede neural é um processador maciçamente e paralelamente distribuído e constituído de unidades de processamento simples que tem a propensão de armazenar conhecimento experimental e torná-lo disponível para o uso. Assim, o aprendizado de uma rede neural se dá pelo fortalecimento dos pesos sinápticos, conexões que interligam os neurônios, através de um algoritmo de aprendizagem, cuja função é modificar os pesos sinápticos da rede para alcançar um objetivo desejado. Durante o treinamento, graças à sua arquitetura e sua capacidade de generalizar, a rede extrai toda informação relevante para a resolução do problema e os armazena em seus pesos sinápticos. A generalização se refere ao fato da rede neural produzir saídas adequadas para entradas que não estavam presentes durante o treinamento, definido como processo de aprendizagem [52, Cap 2].

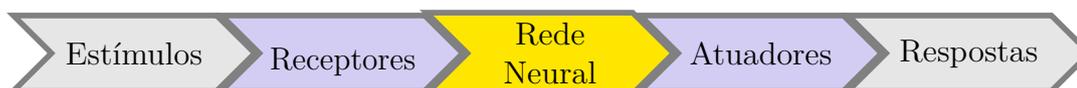
## 3.2 Fundamentos Biológicos

### 3.2.1 Cérebro Humano

O cérebro é basicamente a célula central do sistema nervoso central (SNC) que controla as mais variadas reações do nosso organismo. Partes do corpo que têm as funções controladas pelo sistema nervoso tem tempo de resposta relativamente rápido (100-200 ms) [53]. O SNC humano poder ser visto como um sistema de três estágios, como mostrado no diagrama em blocos da Figura 3.1. A partir de um estímulo externo captado pelos receptores, este estímulo é processado pela rede neural que gera uma atuação externa como resposta.

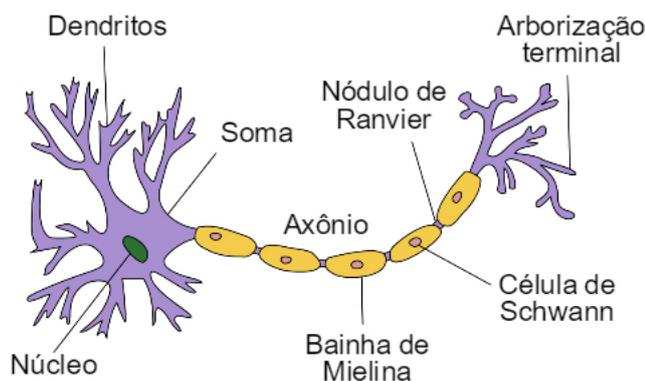
Os neurônios têm um corpo celular denominado soma e diversas ramificações. As ramificações são conhecidas como dendritos e conduzem sinais das extremidades para o corpo celular. Também temos o axônio, responsável por transmitir um sinal de um corpo celular para as suas extremidades, tais extremidades são conectadas com dendritos de outros neurônios através das sinapses. Geralmente um axônio é diretamente conectado com outros axônios ou com o corpo de outro neurônio. As sinapses possuem um papel fundamental na memorização da informação. Cada neurônio tem de 1 mil a 10 mil sinapses e pode receber informação de outros 1 mil neurônios aproximadamente [53]. A Figura 3.2 apresenta uma abstração de um neurônio e suas respectivas partes.

Figura 3.1 – Diagrama de blocos do SNC.



Fonte: Adaptada de [54, Cap 4].

Figura 3.2 – Abstração de um neurônio e suas respectivas partes.



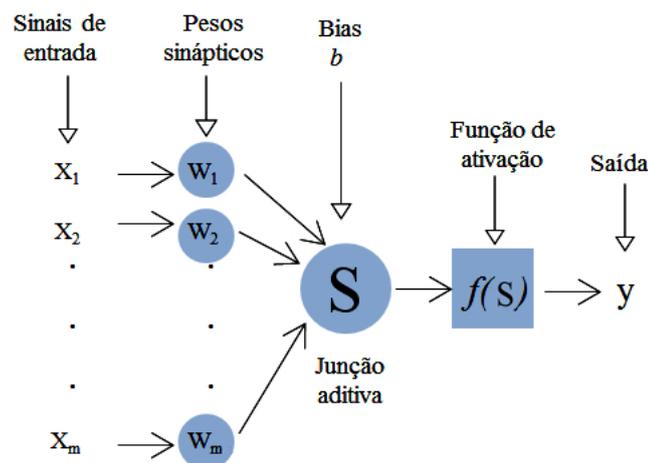
Fonte: Adaptada de [53].

Na biologia, o uso de um órgão favorece o seu desenvolvimento, conseqüentemente a não utilização o atrofia. Cada vez que uma sinapse é ativada e, por conseguinte, gera a ativação de outros neurônios, essa ligação é fortalecida, resultando em um aumento na conexão entre esses neurônios. Este processo chama-se facilitação. O mecanismo de facilitação inspirou a Teoria Hebbiana que descreve um mecanismo básico da plasticidade sináptica no qual um aumento na eficiência sináptica surge da estimulação repetida e persistente da célula pós-sinápticas. Essa teoria serviu de base para muitos algoritmos de aprendizagem de redes neurais [52, Cap 2].

### 3.2.2 Modelo de um Neurônio Artificial

Como falado anteriormente, o neurônio é uma unidade de processamento de informação que é fundamental para as operações de uma rede neural. A Figura 3.3 mostra o modelo de um neurônio. Este modelo serve de base para as redes neurais artificiais.

Figura 3.3 – Modelo de um neurônio.



Fonte: Adaptada de [54, Cap 4].

Os seguintes elementos básicos podem ser identificados no neurônio:

1. Conjunto de entradas caracterizado por sinais de entrada, são as informações que serão processadas pelo neurônio, são as informações que serão processadas pelo neurônio, denotadas pelo conjunto de valores  $\{X_1, X_2, \dots, X_m\}$ . Quando este conjunto de valores refere-se aos valores de entrada da rede neural é denotado vetor característica ou instância da base de dados. Os valores escalares que os sinais de entradas podem assumir pertencem ao conjunto dos números reais;
2. Conjunto de sinapses denominadas por pesos sinápticos, especificamente para cada sinal  $X_j$  de entrada  $j$  é multiplicado pelo seu respectivo peso sináptico  $W_j$ . Os valores escalares que os pesos sinápticos podem assumir pertencem ao conjunto dos números reais;

3. O Somador  $S$  denominado junção aditiva realiza a soma ponderada dos valores de entrada  $X_j$  com seus respectivos pesos sinápticos  $W_j$ , Eq. (3.1). O valor escalar resultante em  $S$  pertence ao conjunto dos números reais;
4. O *bias* denominado  $b$  é um valor escalar pertencente ao conjunto dos números reais independente dos sinais de entrada, que será somado ao resultado do somador  $S$  intensificando ou amenizando o resultado obtido do somador;
5. A função de ativação  $y = f(S)$  é a função de transição que restringe a amplitude da saída do neurônio para um intervalo definido por  $f$ . Os valores escalares que  $y$  pode assumir pertencem ao conjunto dos números reais. Na Figura 3.4 são apresentadas algumas das funções mais comuns e sua respectiva derivada. Além de outras, a *Softmax* é utilizada nos neurônios da camada de saída.

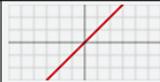
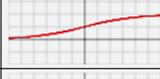
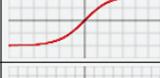
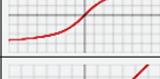
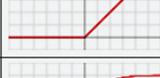
Em termos matemáticos, podemos escrever o neurônio com as seguintes equações, na Eq. (3.1) é apresentada a junção aditiva do neurônio e na Eq. (3.2) é apresentada a ativação do neurônio utilizando como exemplo a função *sigmoide*:

$$S = X_1W_1 + X_2W_2 + \dots + X_mW_m + b \quad (3.1)$$

$$y = f(S) = \frac{1}{1 + \exp^{-S}} \quad (3.2)$$

Em que  $m$  é o número de sinapses (entradas/pesos),  $f$  é a função de ativação e  $y$  é a saída do neurônio que pode assumir valores escalares pertencentes ao conjunto dos números reais restritos à função de ativação utilizada. A *Perceptron* de Camada única [55, Cap 3] é definida pela Eq. (3.2), sendo composta por apenas um neurônio. Sua convergência é garantida pelo Teorema da Convergência do *Perceptron*.

Figura 3.4 – Funções de ativação e sua derivada.

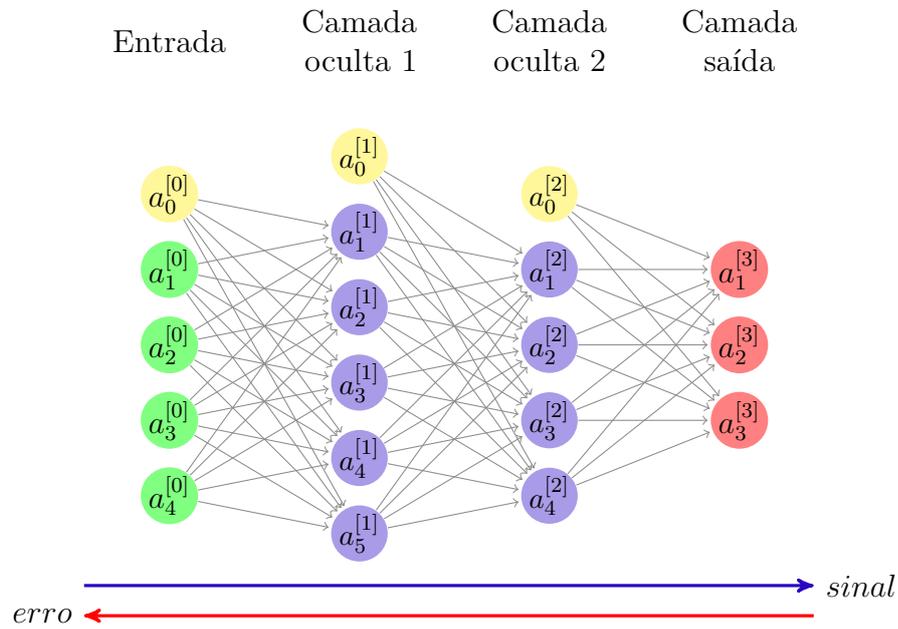
Nome	Gráfico	Equação	Derivada
Identidade		$f(x) = x$	$f'(x) = 1$
Degrau		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Sigmoide		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tangente Hiperbólica		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
Arco Tangente		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Softmax		$f(x) = \frac{\exp(x)}{\sum_k \exp(x)}$	$f'(x) = \begin{cases} f(x)_i(1 - f(x)_j) & \text{if } i = j \\ -f(x)_j f(x)_i & \text{if } i \neq j \end{cases}$

Fonte: Adaptada de [54, Cap 4].

### 3.3 Perceptron de Múltiplas Camadas

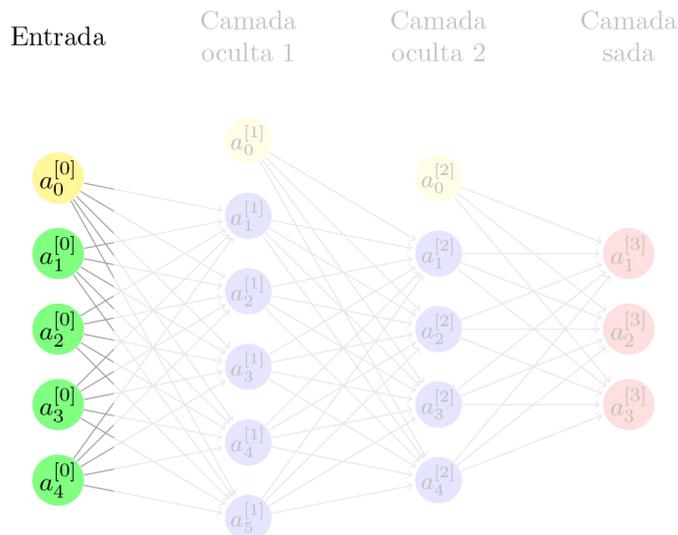
O tópico anterior tratou de uma *Perceptron* de camada única. Este tópico estende tal conceito para as *Perceptron* de múltiplas camadas (do inglês, *Multilayer Perceptron* - MLP). A rede MLP consiste de um conjunto de unidades sensoriais (neurônios) que constituem a camada de entrada, uma ou mais camadas ocultas e uma camada de saída que podem assumir diferentes estruturas e formas de conexões, inclusive pode-se utilizar algoritmos de busca para encontrar uma melhor estrutura para um problema específico como, por exemplo, algoritmos genéticos. A Figura 3.5 ilustra uma MLP de 3 camadas composta por uma entrada, duas camadas ocultas e uma saída. Os neurônios em amarelo representam os *bias*, verde o sinal de entrada, lilás neurônios ocultos e goiaba os neurônios de saída da rede.

Figura 3.5 – Ilustração de uma MLP de 3 camadas.



A *Perceptron* de Camada única foi abordada no tópico anterior. Neste tópico será exemplificada a arquitetura da MLP. A Figura 3.6 representa a entrada da MLP. Os neurônios de entradas  $a_1^{[0]}$  até  $a_k^{[0]}$  representam as variáveis de entrada  $X$  do modelo, nesse caso, leituras de corrente. Já o  $a_0^{[0]}$  representa o *bias* da entrada.

Figura 3.6 – Entrada da MLP.

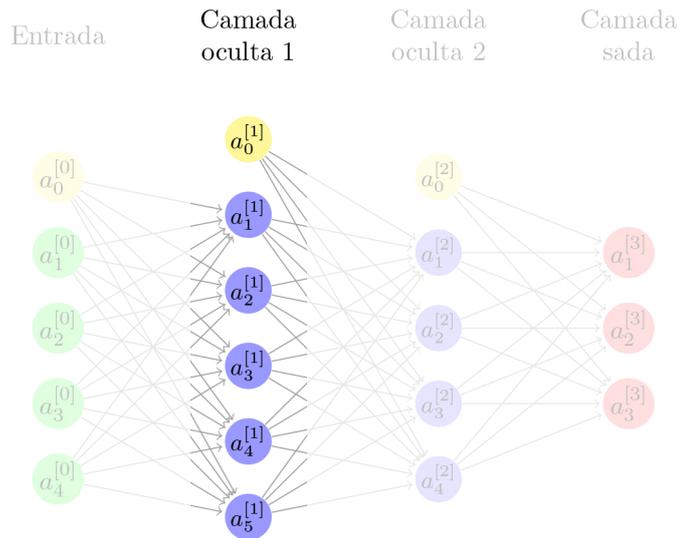


Na Figura 3.7 é possível visualizar a camada oculta  $l$  contendo um conjunto de neurônios. Cada neurônio presente nessa camada e em camadas adiante têm as suas

ativações definidas como  $a_k^{[l]}$  dada pela Eq. (3.3).

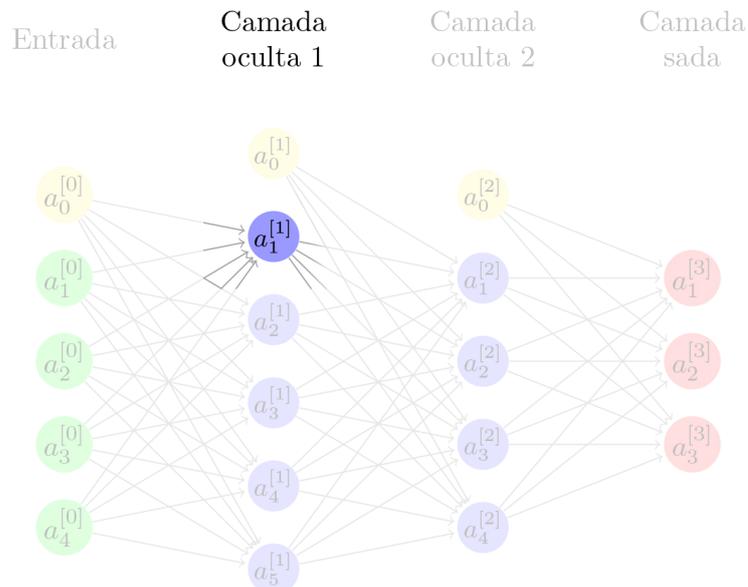
$$a_k^{[l]} = f(S_k^{[l]}) = \sum_{j=1}^m (W_{kj}^{[l]} a_j^{[l-1]}) + b_k^{[l]} \quad (3.3)$$

Figura 3.7 – Camada oculta da MLP.



Com a finalidade de exemplificar as denotações utilizadas, a ativação  $a_k^{[l-1]}$  representa a ativação do neurônio  $k$  da camada anterior  $l - 1$ . Ou seja, o neurônio apresentado na Figura 3.8 tem suas ativações representadas pela a camada de entrada  $a^{[0]}$  e pertence à camada  $l = 1$  que tem sua saída propagada à camada adiante.

Figura 3.8 – Neurônio da camada oculta da MLP.



Ativação resultante  $a^{[l]}$  de cada camada  $l$  é a soma de todas as ativações dos  $k$  neurônios da respectiva camada  $l$  dada pela Eq. (3.4), onde  $n^{[l]}$  representa o número de neurônios ocultos da camada  $l$ .

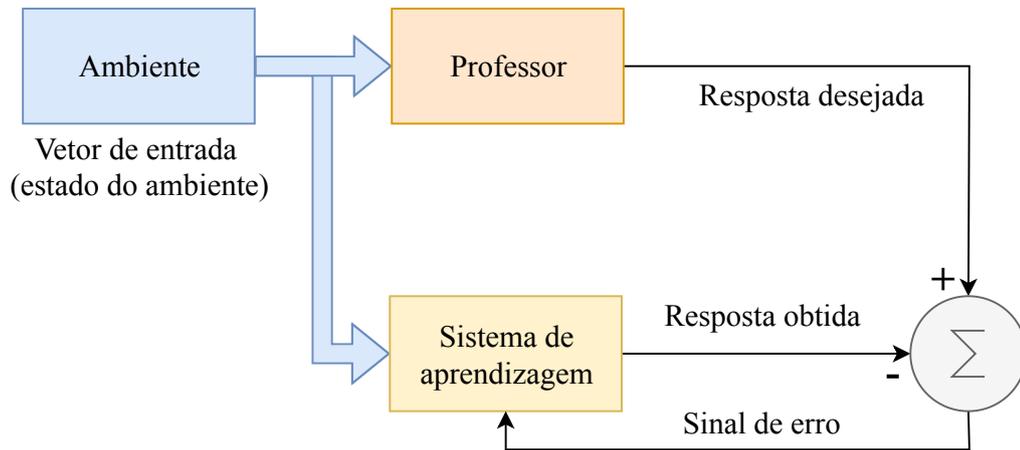
$$a^{[l]} = \sum_{k=1}^{n^{[l]}} a_k^{[l]} \quad (3.4)$$

Temos que  $l = \{1, 2, 3, \dots, L\}$  onde  $L$  é o número de camadas da MLP (profundidade da rede). Sendo assim, a Eq. (3.5) define a saída obtida  $\hat{y}$  igual à ativação  $a$  da camada  $L$ .

$$\hat{y} = a^{[L]} \quad (3.5)$$

A propagação do sinal camada por camada é definida na literatura como propagação adiante (do inglês, *forward propagation*) onde o objetivo é obter a saída do modelo, seja de modo não supervisionado ou supervisionado. Na Figura 3.9 é apresentado o diagrama em blocos de aprendizado supervisionado. O Professor são os rótulos de cada instância e o Sistema de aprendizagem neste exemplo é a rede neural artificial. Vale lembrar que o aprendizado supervisionado pode ser aplicado em diferentes algoritmos de aprendizagem de máquina.

Figura 3.9 – Diagrama em blocos de aprendizado supervisionado.



Fonte: Adaptada de [52, Cap 2].

O sinal de erro  $e^{(i)}$  é calculado para o  $i$ -ésimo elemento do vetor resposta subtraindo o vetor resposta obtida  $\hat{y}^{(i)}$  que foi definida na Eq. (3.5) da resposta desejada  $y^{(i)}$  como é apresentado na Eq. (3.6).

$$e^{(i)} = y^{(i)} - \hat{y}^{(i)} \quad (3.6)$$

Sobre este erro é definida uma função de custo  $J$  para as  $M$  instâncias de treinamento que serão utilizadas para calcular o custo total do modelo, tornando possível avaliar a aprendizagem do modelo a cada época, avaliando o custo. A Eq. (3.7) apresenta um exemplo de função de custo, erro quadrático médio.

$$J = \frac{1}{2M} \sum_{i=1}^M (e^{(i)})^2 \quad (3.7)$$

Outra função de custo utilizada para problemas multi-classes é a entropia cruzada (do inglês, cross-entropy). Na Eq. (3.8) pode-se visualizar a fórmula da entropia cruzada. Diversas funções de custo podem ser utilizadas, tudo dependerá do problema, podendo inclusive ser modelada de acordo com o problema a ser resolvido.

$$J = -\frac{1}{M} \sum_{i=1}^M [y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})] \quad (3.8)$$

Note que quando  $y^{(i)} = 1$  e  $\hat{y}^{(i)} = 1$ , temos que a primeira parcela da equação resulta em zero devido  $\log(1) = 0$  e a segunda parcela resulta em zero devido  $(1 - y^{(i)}) = 0$ . Sendo assim, quando a resposta desejada  $y^{(i)}$  coincide com a resposta obtida  $\hat{y}^{(i)}$  o custo acrescido a  $J$  é zero. Sabendo que  $\hat{y}^{(i)}$  é a saída obtida da rede, sendo esta definida por uma função de ativação.

Até então foram apresentadas a noção biológica do neurônio, a representação matemática do neurônio, *perceptron* de camada única e *perceptron* de múltiplas camadas. Como o sinal é propagado adiante e como é calculado o custo do modelo. A seguir será apresentado como é realizada a atualização dos pesos da rede neural.

### 3.3.1 Algoritmo de Retropropagação de Erro

As sinapses  $W$ , ou pesos da rede como são conhecidos, são responsáveis por reter o aprendizado da rede neural. Inicialmente os pesos são iniciados aleatoriamente, seguido alguma distribuição uniforme ou não. Após a inicialização dos pesos o sinal de entrada é propagado através das camadas seguintes até a saída, onde é calculado o erro como definido na Eq. (3.6). Durante este processo os pesos permanecem fixos. De modo reverso, o sinal de erro anteriormente calculado é propagado no sentido reverso como apresentando na Figura 3.5 e durante este processo os pesos são ajustados camada por camada de acordo com uma regra de correção de erro, como por exemplo o Algoritmo Retropropagação do erro (do inglês, *Backpropagation*).

De maneira resumida, o algoritmo de retropropagação segue os seguintes passos:

1. Inicialização dos pesos da rede de forma aleatória ou seguindo algum método específico.
2. Propagação adiante. Apresente uma ou mais informações do ambiente externo (instâncias), propagando o sinal de entrada camada por camada até que se obtenha a saída e então calcule o erro.
3. Propagação reversa. Em posse do erro o algoritmo de retropropagação calcula os novos pesos de cada neurônio de cada camada de rede, no sentido retroativo (isto é, da saída para entrada).
4. Teste de parada. Critério de parada adotado, caso o resultado já seja satisfatório. Por exemplo, definir um custo mínimo de  $J$  a ser alcançado pode ser um teste de parada.
5. Caso não seja satisfeito, retorne ao passo 2.

A atualização dos pesos é feita pelo cálculo dos gradientes locais (pesos de camada  $l$ ). O campo local como apresentado na Eq. (3.3) para o neurônio  $k$  na camada  $l$  calcula os gradientes locais  $\delta$  da rede definidos pela Eq. (3.9) adaptada de [54, Cap 4].

$$\delta_k^{[L]} = \begin{cases} e_k^{[L]} f'_k(V_k^{[L]}) & \text{para o neurônio } j \text{ da camada de saída } L, \\ f'_k(V_k^{[l]})'(\sum_i (\delta_i^{[l+1]} W_{ki}^{[l+1]})) & \text{para o neurônio } j \text{ na camada oculta } l \end{cases} \quad (3.9)$$

Em que o apóstrofe da função de ativação  $f$  em  $f'_k(\cdot)$  representa a diferenciação em relação ao argumento da função de ativação. Em posse dos  $\delta$  a etapa seguinte é atualizar os valores dos pesos sinápticos  $W$  da rede na camada  $l$  na iteração  $i$  de acordo com a regra delta generalizada, onde a iteração seguinte  $i + 1$  é dada por [54, Cap 4]:

$$W_{kj}^{[l]}(i + 1) = W_{kj}^{[l]}(i) + \alpha[W_{kj}^{[l]}(i - 1)] + \eta \delta_k^{[l]}(i) a_j^{[l-1]}(i) \quad (3.10)$$

Em que  $\eta$  é o parâmetro taxa de aprendizagem e  $\alpha$  é a constante do momento. Ambos são valores definidos normalmente dentro do intervalo  $(0, 1]$ . A taxa de aprendizagem como o próprio nome sugere, define a velocidade de convergência, se muito pequeno a rede neural precisará de um maior número de iterações, se muito grande a rede neural pode não vir a convergir. A constante de momento funciona como um penalizador, controlando a intensidade que os pesos da iteração anterior têm sobre o peso da iteração atual, podendo assumir valor zero.

## 3.4 Treinamento

Como visto nas seções anteriores, o aprendizado é resultado do processamento repetido de todas as amostras de treinamento da MLP. Essas amostras são processadas uma a uma pelo modelo neural, sendo que cada apresentação de todo o conjunto de treinamento durante o processo de aprendizagem é chamada de época. O processo de aprendizagem é repetido época após época, até que um critério de parada seja atingido. É uma boa prática fazer com que as amostras de treinamento sejam processadas de forma aleatória de uma época para a outra. Esta aleatoriedade tende a fazer com que a busca no espaço de pesos tenha um caráter estocástico ao longo do treinamento. A eficiência dos métodos de treinamento também dependem dos dados e do problema a ser tratado [54, Cap 4].

Para um conjunto de treinamento, o aprendizado por retropropagação pode ser feito de três maneiras básicas:

- **Local:** a atualização dos pesos é feita imediatamente após o processamento de cada amostra de treinamento. Considere um conjunto de treinamento com  $M$  amostras de treinamento  $\{(x_1, y_1), \dots, (x_M, y_M)\}$ , onde  $x_1$  é a primeira amostra de treinamento e  $y_1$  sua respectiva classe, é então processada pela rede, e são efetuadas atualizações dos pesos sinápticos através da retropropagação. Em seguida é a seguinte  $(x_2, y_2)$  e novamente os pesos sinápticos são atualizados. Esse processo é repetido até que a última amostra seja processada pela rede caracterizando uma época e até que atinja o critério de parada. Este método também é conhecido como treinamento *online* ou treinamento estocástico. Bastante utilizados nos algoritmos de SPAM (do inglês, *Stupid Pointless Annoying Messages*), nos sites e redes sociais, com a finalidade de capturar a atenção dos usuários recomendando ofertas de produtos que o usuário tenha pesquisado ou "curtido" recentemente.
- **Lote:** a atualização dos pesos só é realizada após o processamento das  $M$  amostras de treinamento pela MLP, o que de fato constitui uma época. O ajuste que deveria ser realizado no treinamento local é acumulado amostra por amostra e após o processamento da  $M$ -ésima amostra é então realizada a atualização dos pesos sinápticos utilizando o algoritmo de retropropagação. Este processo é também conhecido como treinamento *offline* ou *batch*.
- **Mini-Lote:** de maneira semelhante ao treinamento em lote. As  $M$  amostras de treinamento são subjetivamente subdivididas em  $Q$  subgrupos de  $M/Q$  amostras de treinamento, a cada subgrupo  $Q_i$  processado pela MLP é então realizada a atualização dos pesos utilizando o algoritmo de retropropagação. Na prática o algoritmo irá atualizar os pesos a cada  $s$  iterações, onde  $s$  é basicamente o tamanho do **mini-lote** definido anteriormente. E novamente esse processo é repetido inúmeras vezes até que

se atinja o critério de parada. Conhecido também por *mini-batch*.

## 3.5 Critério de Parada

O processo de minimização da função custo  $J$ , em geral, não tem convergência garantida e não possui um critério de parada bem definido, aquele que garanta o ótimo. Um critério de parada não recomendável, por não levar em conta o estado do processo iterativo de treinamento, é interromper o treinamento após um número fixo de iterações definido previamente. Outro critério de parada busca um valor mínimo de  $J$  tal que  $J \leq \epsilon$ , onde  $\epsilon$  é um valor fixo ou variável que é definido anteriormente ou dinamicamente ao longo dos treinamentos. A medida que o valor  $\epsilon$  é batido, um novo valor para  $\epsilon$  pode ser definido utilizando alguma regra desejada.

Um critério de parada bastante usual e eficaz é subdividir suas amostras em treinamento, validação e teste. O conjunto de treinamento será utilizado no processo de aprendizagem do modelo e o de teste para verificar após finalizado o treinamento se o modelo consegue generalizar para dados nunca apresentados à rede neural durante o treinamento. O conjunto de validação será utilizado para verificar o número de acertos que o modelo obteve com tal conjunto a cada época ou um número fixo de iterações passadas. Este método busca maximizar a acurácia do modelo sobre o conjunto de validação. Vale salientar que durante o teste sobre o conjunto de validação os pesos não são atualizados, ou seja, para o algoritmo de retropropagação essa verificação sobre o conjunto de validação não existiu. Isso garante que o conjunto de validação continue desconhecido do modelo garantindo que nas verificações seguintes, após outro ciclo de épocas ou iterações, o modelo verifique novamente sua acurácia sobre o conjunto de validação desconhecido. Utilizando esta técnica, a generalização do modelo dependerá diretamente da qualidade do conjunto de validação selecionado. Neste caso a diferença entre o conjunto de teste e o conjunto de validação pode ser apenas o volume de amostras ou até diferentes fontes de amostras, pois ambos os conjuntos se mantêm desconhecidos do modelo.

Diversos outros critérios de parada podem ser criados, modificados ou combinados, tudo dependerá do problema a ser tratado e das dificuldades encontradas durante do treinamento.

## 3.6 Validação do Módulo Classificador

A validação cruzada é uma técnica bastante aceita na estatística, servindo para avaliar a capacidade de generalização de um modelo, dado um conjunto de dados rotulados. Também é amplamente utilizada em problemas que tem como objetivo a modelagem e a predição. Essa técnica também ajuda a evitar o enviesamento modelo que pode ocorrer

devido a organização do conjunto de dados, isto é, a sequência ao qual as amostras são apresentadas ao modelo influencia diretamente o espaço de busca dos algoritmos de otimização.

O funcionamento das técnicas de validação cruzada se resume ao particionamento do conjunto de dados em subconjuntos mutuamente exclusivos, e posteriormente, é aferido estimativas a partir do processo de validação de cada um desses subconjuntos. O processo de divisão das amostras em subconjuntos é realizado de forma aleatória, garantindo que a organização dos dados não influencie o resultado do modelo.

Existem vários tipos de validação cruzada, porém são apenas variantes do método *K-Fold*. Nesta estratégia, a matriz de dados é subdividida aleatoriamente em  $K$  partições, levando em consideração o número de instâncias de cada classe. Um subconjunto é retirado do conjunto de dados para ser utilizado na fase de validação, enquanto as demais  $K - 1$  ( $K$  menos um) subconjuntos são utilizados como dados do treinamento. O processo de retirada de um subconjunto é então repetido  $K$  vezes, até que cada um dos subconjuntos seja utilizado uma vez como subconjunto de validação, sendo extraídas métricas de desempenho a cada subconjunto utilizado no processo de validação.

Na Figura 3.10 é apresentada uma ilustração do processo de validação cruzada *K-Fold* sobre o conjunto de dados, com  $K = 10$ , a cada *round* ( $K = 1$ ) uma partição é removida para ser utilizada na fase de validação daquele *round*, obtendo assim a acurácia do respectivo *round*. No *round* ( $K = 2$ ) seguinte, a partição utilizada anteriormente é reinsertada no conjunto de dados, mas removida do sorteio, restando assim apenas nove subconjuntos a serem testados. O modelo é reinicializado, os pesos da MLP são resetados, isto é, cada *round* funciona de forma independente ao anterior e posterior, garantindo que as estimativas sejam independentes umas das outras. Com essas  $K$  estimativas temos uma confiança maior sobre o comportamento do modelo. Ilustrando de outra forma, a Figura 3.11 apresenta o mesmo processo de validação cruzada *K-Fold*, com  $K = 3$ . Onde  $X$  são as instâncias e  $Y$  são os rótulos,  $T$  significa as partições utilizadas para treinamento e  $V$  a partição de validação no respectivo *round*.

Figura 3.10 – Ilustração do processo de validação cruzada *10-Fold*.

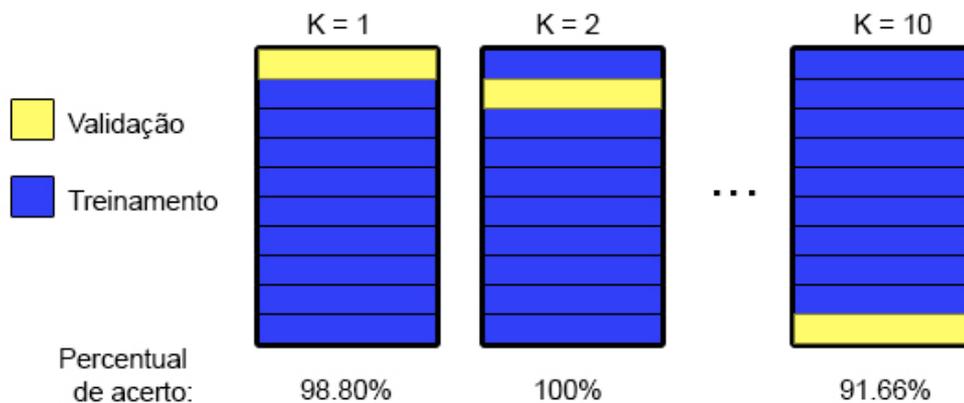
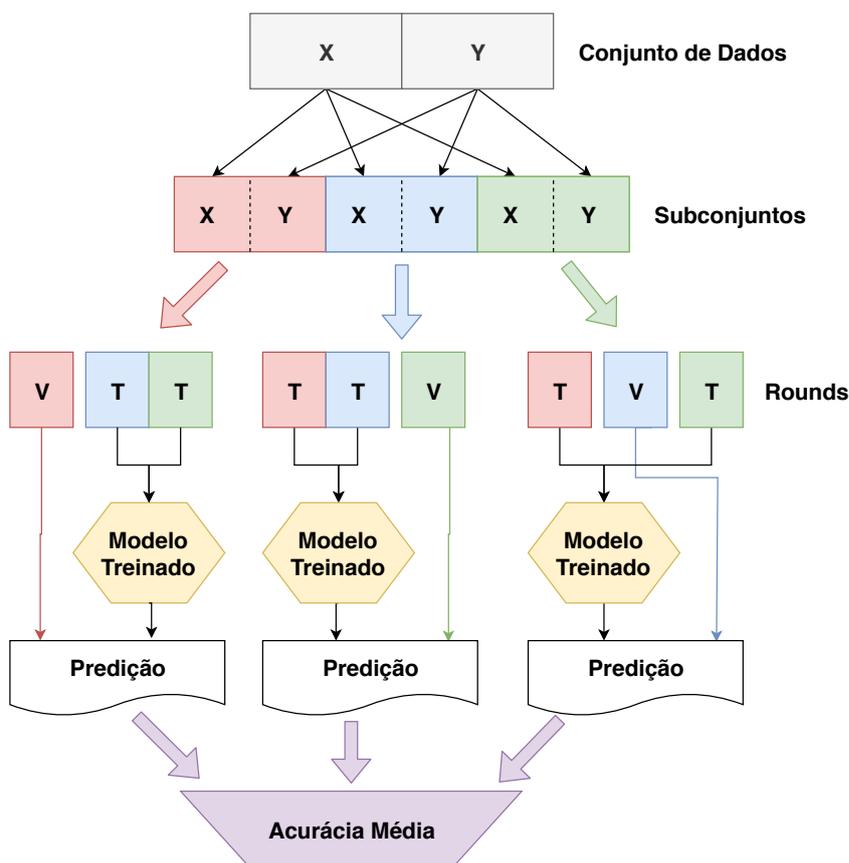


Figura 3.11 – Ilustração do processo de validação cruzada *3-Fold*.



### 3.7 Matriz de Confusão

Para exemplificar, a Tabela 3.1 ilustra um exemplo de matriz de confusão para três classes distintas, A, B e C. É um tabela que mostra as frequências de classificação para cada classe do modelo. As seguintes denominações são utilizadas:

- **Verdadeiro positivo** (do inglês, *true positive* - TP): ocorre quando a classe prevista

assume o valor esperado, rótulo para aquela instância. Exemplo, na Tabela 3.2 o TP tem valor 4, apenas quatro instâncias da classe B foram previstas como classe B;

- **Verdadeiro negativo** (do inglês, *true negative* — TN): são todas as classes que não pertencem à classe analisada e foram previstas corretamente. Exemplo, na Tabela 3.2 o TN tem valor 17, pois 6 instâncias da classe A foram previstas como classe A e 11 instâncias da classe C foram previstas como classe C, somando assim 17 instâncias previstas corretamente e que não pertence a classe analisada;
- **Falso positivo** (do inglês, *false positive* — FP): ocorre quando as amostras não pertencentes à classe analisada são classificadas como sendo da classe analisada. Exemplo, na Tabela 3.2, o FP tem valor 5, pois 4 instâncias da classe A e 1 instância da classe C foram classificadas que como sendo da classe B;
- **Falso negativo** (do inglês, *false negative* — FN): ocorre quando as amostras pertencentes à classe analisada são classificadas como sendo de outras classes. Exemplo, na Tabela 3.2, o FN tem valor 5, pois 3 instâncias foram classificadas como A e 2 instâncias como C, quando na verdade pertenciam à classe B.

		Previsto			FN
		A	B	C	
Rótulo	A	6	4	0	4
	B	3	4	2	5
	C	0	1	11	1
FP		3	5	2	10

Tabela 3.1 – Matriz de Confusão

B	
(TP) 4	(FN) 5
(FP) 5	(TN) 17

Tabela 3.2 – Exemplo da matriz de confusão da Classe B.

A precisão é o *TP* dividido pela soma de *TP* com *FP*. Para que a precisão seja de 100% o modelo precisa classificar todas as amostras que não pertencem à classe de interesse corretamente, ou seja, meu *FP* será zero. Traduzindo em formulação, temos:

$$Precisao = \frac{TP}{TP + FP} \quad (3.11)$$

O *recall* ou revocação é o *TP* dividido pela soma de *TP* com *FN*. Ou seja, representa o quão bom o modelo é para prever instâncias da classe analisada. Como visto

anterior,  $FN$  representa quantas amostras da classe de interesse foram classificadas como sendo de outras classes. Traduzindo para fórmula, temos:

$$Recall = \frac{TP}{TP + FN} \quad (3.12)$$

O  $F1$  Score ou  $F$ Score combina precisão com  $recall$  de modo a trazer um número único que indique a qualidade geral do modelo e trabalha bem até com conjuntos de dados que possuem classes desbalanceadas.

$$FScore = 2 \frac{Precisao * Recall}{Precisao + Recall} \quad (3.13)$$

A acurácia do modelo muitas vezes confundida com a precisão do modelo, diz quanto o modelo acertou das previsões possíveis. Esta métrica é bastante usual para problemas com classes balanceadas, mas em problemas multi-classe não é possível ter uma perspectiva individual de cada classe.

$$Acuracia = \frac{TP + TN}{TP + FP + TN + FN} \quad (3.14)$$

### 3.8 Considerações do Capítulo

As aplicabilidades das RNA's são ilimitadas, basicamente todo conhecimento passível de representação em máquinas de *Von Neumann* pode também ser adequadamente modelado e representado em RNA's, conhecimento este, ficando armazenado em seu conjunto de pesos. Isso significa que, com base na configuração resultante da rede, a intensidade das conexões é ajustada com o objetivo de armazenar informação relevante para solução de um problema específico. Como resultado natural desse processo de ajuste, o conhecimento armazenado fica distribuído pelas camadas da rede neural e disponível para uso.

Para predição de novas amostras, a amostra a ser classificada é multiplicada pela camada adiante, seu resultado é então ativado utilizando a função de ativação anteriormente definida e novamente propagado adiante até que se obtenha a ativação da camada de saída. Sobre a matriz resultante da ativação da saída é então interpretada a classificação da amostra como sendo aquele valor de maior intensidade e, a coluna ou a linha (a depender de como foi implementado) a qual se encontra este valor representará a classe. Ou seja, se o vetor de saída tem dimensão 1x11 (1 linha e 11 colunas) e o maior valor foi encontrado na coluna 5, implica que a amostra foi classificada como pertencente à classe representada pela coluna 5.

No Capítulo seguinte são apresentados os métodos desenvolvidos no decorrer deste trabalho. A descrição do modelo da turbina baseada em PMSG utilizado para gerar os dados, a análise exploratória dos dados e a descrição do modelo MLP proposto.

## 4 Revisão da Literatura

Neste capítulo são apresentados alguns dos trabalhos mais relevantes na detecção de falhas em turbinas eólicas, os dados e os classificadores utilizados para detecção das falhas.

### 4.1 Artigos Relacionados

O interesse em monitoramento e detecção de falhas em motores elétricos é crescente nas últimas décadas, sendo evidenciado pelos trabalhos disponíveis na área. Os pesquisadores estão focando seus esforços em métodos não invasivos e contínuos, os quais utilizam por exemplo: sensores de temperatura, sensores de vibração e leituras de corrente do motor como fonte de informação. Estes dados são coletados e analisados de forma contínua, diferentemente de outros métodos já validados, como o teste de surto [21] para detecção de falhas em espiras do estator do motor. Alguns dos trabalhos mais relevantes são apresentados a seguir em ordem cronológica.

Em [56] a detecção incipiente de falhas para DFIG é experimentalmente investigada. Os autores relatam que o percentual de falha por componentes em máquinas de indução é cerca de 40% e estas falhas estão relacionadas a rolamentos, 38% ao estator, 10% ao rotor e 12% a outros elementos. O artigo mostrou que a detecção dessas falhas também é possível pela análise de domínio de tempo e frequência, utilizando o espectro de frequência das correntes das linhas do estator e do rotor.

Em [57] é apresentado um estudo das principais causas de falhas na geração eólica, assim como métodos de detecção de falhas e considerações gerais sobre proteção e manutenção preditiva dos equipamentos e o quanto isso afeta os custos de manutenção com foco principal na caixa de engrenagens e rolamento, rotor e pás, gerador e conversores que compõem o sistema. O trabalho também fala que falhas em máquinas de indução podem produzir alguns dos seguintes fenômenos: desequilíbrios de corrente, harmônicos no fluxo de ar, correntes de fase, aumento da pulsação de torque, redução do torque médio, aumento de perdas e redução na eficiência e aquecimento excessivo no enrolamento.

Em [58] é apresentado um apanhado do monitoramento constante como forma de reduzir os custos operacionais e de manutenção, a fim de minimizar o tempo de inatividade e maximizar a produtividade. O trabalho descreve as diferentes estratégias de manutenção, técnicas e métodos de monitoramento, destacando em uma tabela as várias combinações que foram relatadas na literatura. Ao final, três problemas foram

destacados, os quais devem ser enfrentados pelos especialistas em monitoramento de condições de funcionamento de turbinas eólicas: o número e tipo de sensores, os métodos de processamento de sinal e a criação de um modelo eficaz (isto é, a combinação de sensores e métodos de processamento de sinal que proporcionem um desempenho otimizado). Este desempenho otimizado é caracterizado pela precisão do método e sua velocidade (latência), bem como custo operacional de processamento.

Em [59] é apresentado um método para detecção de falhas entre espiras em EESG e PMSG baseado em análise de assinatura de corrente (do inglês, *Stator Current Signature Analysis* - SCSA). O método utiliza análise do espectro de frequências na corrente estatórica para detectar uma ou mais falhas entre os enrolamentos das máquinas. O autor também relatou que o método SCSA com análise de espectro de frequências requer métodos mais sofisticados para geradores de velocidade variável, por exemplo, geradores PMSG. Os autores também destacam a dificuldade em realizar essa detecção utilizando o domínio do tempo.

Em [23] é apresentada uma proposta e a validação experimental de assinaturas de tensão e corrente para detectar falhas incipientes no PMSG. Dentre as falhas discutidas, uma delas é a falha nas espiras do estator. A metodologia proposta se baseia na análise da assinatura da terceira harmônica de tensão e seus múltiplos sendo capaz de detectar a fase defeituosa. Os autores também destacam a importância de realizar a detecção incipiente das falhas, evidenciando que o progresso da falha torna-se visível através do desequilíbrio, vibração ou aumento da corrente.

Em [10] é apresentado um método caracterizado por dois estágios: extração e classificação destas características utilizando árvore de decisão e MLP. Para isso, o autor utiliza o algoritmo de informação mútua (do inglês, *Mutual Information Algorithm* - MI) usando leituras de corrente das fases do estator como fonte de informação para cálculo do MI. Tais características são então normalizadas e utilizadas de entrada em dois classificadores distintos de reconhecimento de padrões, árvore de decisão e MLP. Os autores destacam que os melhores resultados foram obtidos com a MLP, tanto para avaliações offline quanto para avaliações online, com acurácia média de mais de 83%, independentemente das condições de operação e do classificador. Porém, considerando os níveis de curto-circuito iguais ou superiores a 3%, a precisão mínima foi de 95%.

Em [6] é apresentada uma abordagem baseada em uma rede neural de convolução (do inglês, *Convolutional Neural Network* - CNN) para detectar falhas na caixa de engrenagens de turbinas eólicas, a qual é capaz de extrair características e realizar classificação de forma contínua. Este método não requer uma etapa de processamento de sinal ou diagnósticos de especialistas na área, como definido na Aprendizagem Profunda. O método utiliza 3 CNN acopladas e cada uma recebe o sinal de sensores de vibração em

diferentes escalas de amostragem. O sinal de vibração vem de sensores presentes na caixa de engrenagens, esse sinal original é sub-amostrado a partir da média 2 a 2 elementos, gerando o sinal para a segunda CNN. Oito defeitos na caixa de engrenagem foram considerados, cada um a cinco níveis de rotação e dois carregamentos distintos. Por fim, o sinal original é sub-amostrado a partir da média 3 a 3 elementos para a terceira CNN. Destacando uma acurácia média de 98,53%.

Em [60] é apresentado um novo método que utiliza múltiplos sensores de sinais de vibração tridimensional advindos da caixa de engrenagem da turbina eólica. Este método converte janelas desse sinal tridimensional de vibração em imagens. Essas imagens são utilizadas como entrada na CNN que tem como objetivo classificar as falhas na caixa de engrenagem. O autor compara o método proposto com outras técnicas utilizadas na literatura, como a CNN para sinal de uma dimensão de um único sensor e de múltiplos sensores, além de comparar também com outros tipos de RNA e com a Máquina de Vetores de Suporte (do inglês, *Support Vector Machine* - SVM). Quatro tipos de falhas da caixa de engrenagens foram consideradas, sendo estas autoinfligidas em maquinário físico, duas no rolamento e duas no dente da engrenagem. Por fim, o trabalho apresenta acurácia de 99,47% relatando que o método proposto supera os demais métodos analisados, tanto em precisão, quanto em número de épocas necessárias para se obter resultados satisfatórios.

## 4.2 Considerações do Capítulo

Como apresentado nesse Capítulo, diversos trabalhos têm buscado melhorar a eficácia no monitoramento contínuo de falhas internas e de falhas externas ao gerador. Outros trabalhos [61, 62] mostraram que não existe um consenso sobre qual a melhor fonte de informação (e.g. corrente, tensão, vibração), como forma de caracterização das informações geradas por estas fontes, ou modelo para classificação das informações geradas. Portanto, dependendo da falha que se deseja classificar, diferentes fontes de informação podem ser utilizadas.

Foi notado que um dos principais problemas enfrentados em detecção e classificação de falhas internas ao gerador é a falta de dados de treinamento advindos de turbinas reais em diferentes estágios de evolução de falhas, o que dificulta a aplicação de técnicas de aprendizagem supervisionada, bem como a criação de modelos consistentes. Uma alternativa bastante utilizada pelos autores é a modelagem matemática, isto é, a geração de dados por meio de simulação com o Simulink do MatLab/MathWorks [63].

No Capítulo seguinte é apresentado o embasamento teórico das turbinas eólicas e as falhas relacionadas a estas turbinas.

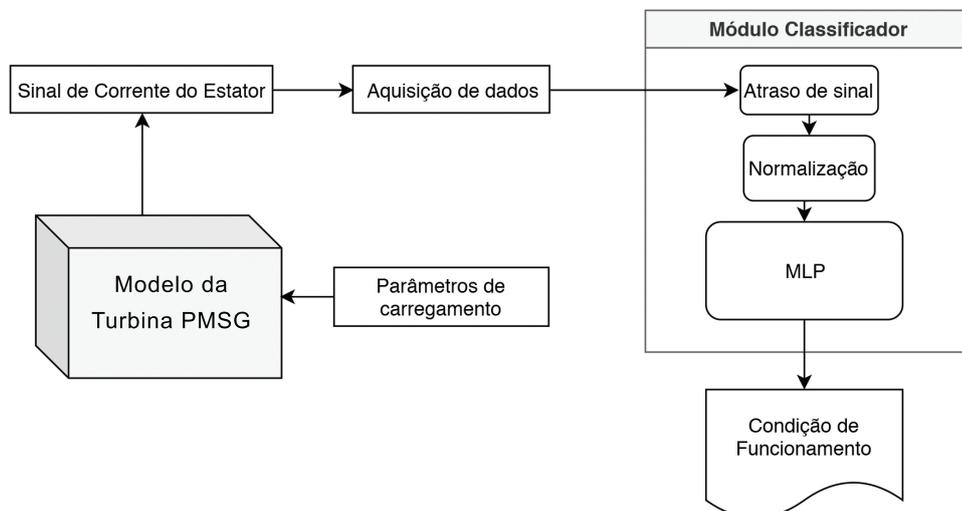
## 5 Metodologia

Neste capítulo são descritos os métodos utilizados desde a geração dos dados até a validação do modelo de aprendizagem de máquina baseado em aprendizagem profunda. Para isso, uma base de dados foi gerada através de um modelo matemático no Simulink/MatLab de um sistema eólico equipado com o PMSG [64], e foram armazenados dados de correntes elétricas advindas de uma das fases do estator do PMSG para os estados de defeito espira-terra e espira-espira com percentual de falha variando de 1% a 10% (incrementos de 1%) e em condição normal. Estas amostras foram utilizadas para a definição de uma arquitetura MLP com seus respectivos parâmetros. Assim, a partir das leituras de correntes foi possível detectar e classificar o estágio de evolução das falhas analisadas em estágios incipientes.

### 5.1 Plano de Atividades Desenvolvido

Na Figura 5.1 está ilustrado o plano de atividades utilizando diagrama de blocos. O modelo da Turbina eólica baseada em PMSG é responsável por gerar os dados a partir dos parâmetros de carregamento pré-definidos. O sinal de corrente do estator é coletado/exportado para ser utilizado no módulo classificador, o qual recebe um sinal segmentado de entrada pré-definida em janela de tempo como meio segundo (500 ms). A amplitude deste sinal é então normalizada para um intervalo conhecido (-1, 1) e em seguida é processada pela MLP. A saída da MLP é então interpretada para relatar a condição de funcionamento da turbina.

Figura 5.1 – Diagrama de blocos do plano de atividades.



A seguir são descritas as etapas que foram realizadas no desenvolvimento da pesquisa:

- Modelagem dos dados: Definição dos parâmetros de carregamento (percentual de faltas comprometidas, carregamento de carga, resistência da falha, etc.) para as diferentes condições de funcionamento;
- Aquisição de Dados: Coletar sinal de corrente do estator para as diferentes condições de funcionamento, conjunto de dados;
- Análise exploratória dos dados: Tem como finalidade o entendimento do comportamento dos dados coletados. Nesta etapa é realizada a rotulação dos dados de acordo com as informações extraídas e os objetivos propostos da pesquisa;
- Arquitetura da MLP: obter uma arquitetura da MLP (e.g. entradas, camadas, quantidade de neurônios e saídas) que satisfaça os objetivos da pesquisa;
- Validação do modelo: Treinamento e validação do modelo com as instâncias de estados que não participaram do treinamento.

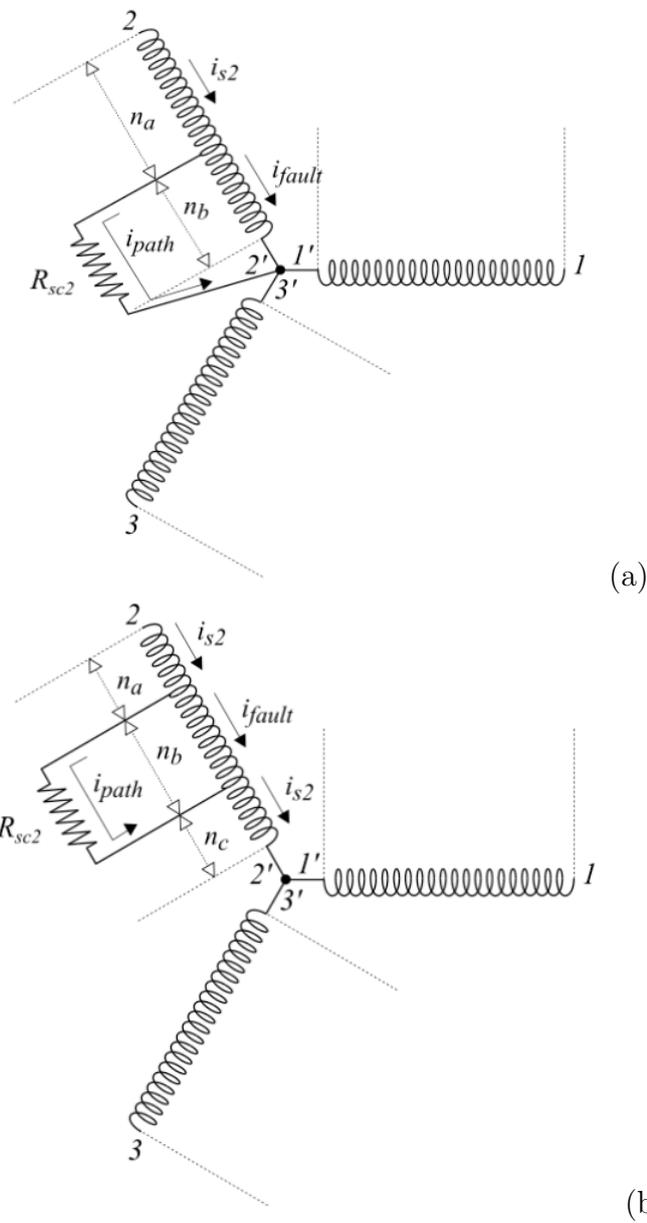
## 5.2 Modelo para Geração de Falhas

Um dos tipos mais comuns de faltas internas em geradores elétricos é a fuga de corrente entre espiras do estator através de falhas no isolamento [65]. Na Figura 5.2 estas ligações apresentam uma resistência elétrica  $R_{scj}$ , com  $j = 1, 2$  e  $3$ . O valor desta resistência depende do grau de falha no isolamento, sendo que  $R_{scj} = \infty$  quando não existem falhas e  $R_{scj} = 0$  quando existe uma eliminação total do isolamento. Nos caminhos, em condição normal a corrente  $i_{path}$  é acionada e, ao longo da seção defeituosa da bobina, a corrente  $i_{fault}$  é acionada. E  $i_{sj}$  são as correntes do estator. Além disso, diferentes posições ou frações das espiras da bobina podem sofrer curto-circuito devido às falhas [66].

Na Figura 5.2a é apresentado o esquemático da falha espira-terra. Supõe-se que o enrolamento do estator esteja em curto-circuito espira-terra em uma determinada posição na bobina da fase 2. A bobina é dividida em duas seções, espiras  $n_a$  que é adjacente ao terminal da máquina e outra com espiras  $n_b$  que é adjacente ao terra [66].

Na Figura 5.2b, apresenta a fase 2 sob a falha espira-espira. Nesse caso,  $n_a$  é o número de voltas da seção da bobina próximo ao terminal,  $n_b$  é a fração de voltas em curto-circuito e  $n_c$  é o número de espiras próximos ao terra [66].

Figura 5.2 – Ilustração do estador do PMSG para simulação das falhas: (a) espira-terra e (b) espira-espira.



Com o objetivo de modelar os efeitos dessas faltas, uma importante análise foi relacionada aos efeitos do curto-circuito sobre as resistências e indutâncias das espiras defeituosas. Deste modo, a resistência da espira pode ser dada por 5.1:

$$R_s = \rho \frac{l}{A_1} \quad (5.1)$$

Em que,

$\rho$  - Resistividade do material da espira;

$l$  - Comprimento do enrolamento da espira;

$A_1$  - Área da seção transversal do enrolamento.

O único parâmetro sensível a este tipo de falta é  $l$ . Desta forma, se uma fração  $n_j$  da espira estiver curto-circuitada, uma resistência igual a  $n_j R_s$  corresponderá à fração defeituosa da espira e outra resistência igual a  $(1 - n_j) R_s$  corresponderá à fração não defeituosa.

A autoindutância  $L_s$  de solenoides, tais como as espiras mostradas na Figura 5.2, podem ser computadas através da Equação 5.2, [30]:

$$L_s = \mu \frac{N^2}{l} A_2 \quad (5.2)$$

Em que,

$\mu$  - Permeabilidade magnética do núcleo;

$N$  - Número de espiras;

$A_2$  - Área da seção transversal envolta pelas espiras.

Os parâmetros sensíveis a este tipo de falta são  $N$  e  $l$ . De forma similar às resistências, uma indutância igual a  $n_j L_s$  corresponde à fração defeituosa da espira, enquanto outra indutância igual a  $(1 - n_j) L_s$  corresponde a fração não defeituosa. Desde que a indutância mútua entre enrolamentos do estator  $M_s$  é dada pela Equação 5.2, que desconsidera o fator de enrolamento, tal indutância também é sensível as falhas.

$$M_s = -\frac{L_s}{2} \quad (5.3)$$

Além disso, são consideradas forças eletromotrizes distribuídas ao longo dos enrolamentos. O circuito equivalente final é apresentado na Figura 5.3, onde as chaves  $Sa$ ,  $Sb$  e  $Sc$  são abertas/fechadas para simular as falhas nos isolamentos do sistema.

Figura 5.4 – Esquema de alimentação do modelo.

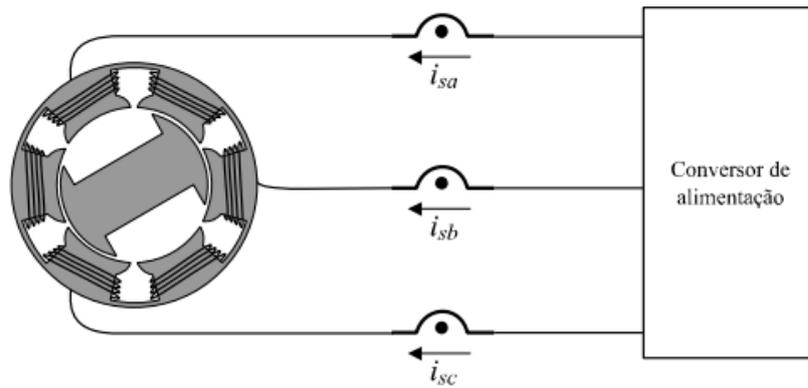
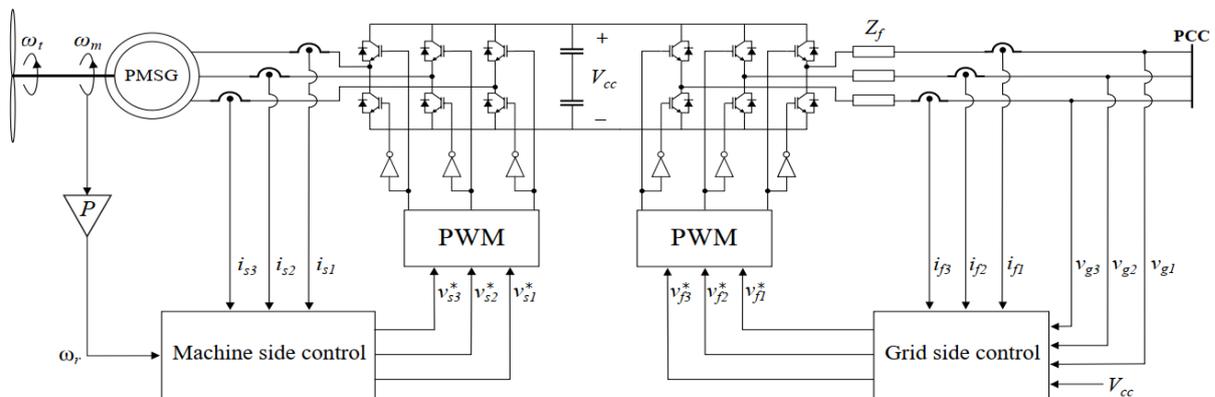


Figura 5.3 – Diagrama esquemático do circuito para geração de falhas.



Fonte: [66].

### 5.3 Criação da Base de Dados

Baseado na implementação do modelo para geração de falhas citado na seção anterior, foram realizadas as leituras de corrente da fase  $i_{sb}$ . O esquema ilustrado na Figura 5.4 apresenta a alimentação do modelo.

O modelo também permite a representação de dois tipos de defeitos. Na Figura 5.5 o esquema ilustrado apresenta uma abstração dos dois defeitos: o curto-circuito entre espiras (defeito 1) e o curto-circuito espira-terra (defeito 2). Ambos os defeitos podem ser simulados pelo modelo.

Figura 5.5 – Tipos de falhas internas



Existem diversos parâmetros que podem ser variados a partir deste modelo, sendo eles:

- Tipos de operação: normal e defeito 1 e 2;
- Carga do gerador: que depende diretamente da velocidade do vento, podendo variar de 0 a 100%;
- Frequência de chaveamento do conversor de alimentação: pode variar de 1 a 30 kHz;
- Percentual de espiras em curto: pode variar de 0 a 100%;
- Resistência do caminho (falha de isolamento): pode variar de 0 (rompimento total do isolamento) a infinito (operação normal).

Para experimentos iniciais, foram gerados dados a uma taxa de amostragem de 1 kHz com uma carga de 100%, com as frequências de chaveamento de 5 kHz e 2,5 kHz e uma resistência de falha 0,1 e 0,2 ohms para cada percentual de defeito, que variou neste trabalho de 1% a 10% para condições com defeito 1 e 2. Para condições normais foram utilizadas a mesma taxa de amostragem, carga, frequências de chaveamento e percentual definido em 0% de espiras em curto-circuito.

## 5.4 Análise Exploratória dos Dados

Neste tópico foi avaliado o comportamento das classes presentes na base de dados, com a finalidade de compará-las entre si, de modo que seja possível visualizar ou mensurar as peculiaridades de cada classe. Normalmente, utiliza-se de métodos visuais, explanados mais adiante no texto, para descobrir variáveis importantes para o problema, bem como detectar comportamentos anômalos nos dados. Esse processo auxilia na definição da arquitetura da MLP, auxiliando na identificação de possíveis métricas que podem servir de vetor características para entrada do modelo MLP.

Toda a análise exploratória foi realizada sobre amostras de 500 ms a 1 kHz de amostragem e 5 kHz de chaveamento das respectivas classes do problema, ou seja, cada instância/amostra contém meio segundo (0,5 s) de leitura de corrente  $i_{sb}$  para as condições de funcionamento consideradas. O tamanho da amostra e a frequência de amostragem foram definidos de forma empírica. Todos os gráficos a seguir foram desenvolvidos utilizando a linguagem Python [67] versão 3.x com auxílio das bibliotecas OpenSource: *pandas* [68], *matplotlib* [69], *IPython* [70], *NumPy* [71] e *Scikit-Learn* [72].

### 5.4.1 Segmentação e Rotulação

A segmentação e rotulação é caracterizada pela fase de entendimento do dado como um todo. É nessa fase onde são definidas as classes do problema que o modelo MLP irá trabalhar. Na Figura 5.6 é apresentada uma amostra normal e do defeito 1 a 1% e 2%. Do ponto de vista gráfico, não existe diferença significativa entre a classe normal e a 1%

de defeito, apenas uma leve irregularidade na amplitude das senoides do defeito a 1% em relação à condição normal. Esta irregularidade é facilmente visualizada no defeito a 2%.

O gráfico *boxplot* resume de forma gráfica as principais métricas das amostras analisadas, média, valores de máximo e mínimo, mediana, primeiro e terceiro quartil. A Figura 5.7 apresenta a interpretação generalista do *boxplot*.

Na Figura 5.8 podemos visualizar o *boxplot* de amostras das classes normal (0% defeito) e 1% a 3% dos defeitos 1 e 2. Note a semelhança entre os defeitos 1 e 2, seus limites superiores e inferiores, além dos quartis, são praticamente idênticos visualmente. Também é possível observar o quão próximos estão os dados normais dos dados com 1% de defeito. Como as senoides são sinais de corrente alternada, temos sua média e mediana tendendo a zero.

Já a Figura 5.9 é apresentado o histograma de amostras das classes normal e 1% a 3% dos defeitos 1 e 2. À medida que o percentual de defeito aumenta podemos notar a presença de valores de maior amplitude.

Figura 5.6 – Exemplo de amostra normal e defeito 1 a 1% e 2%.

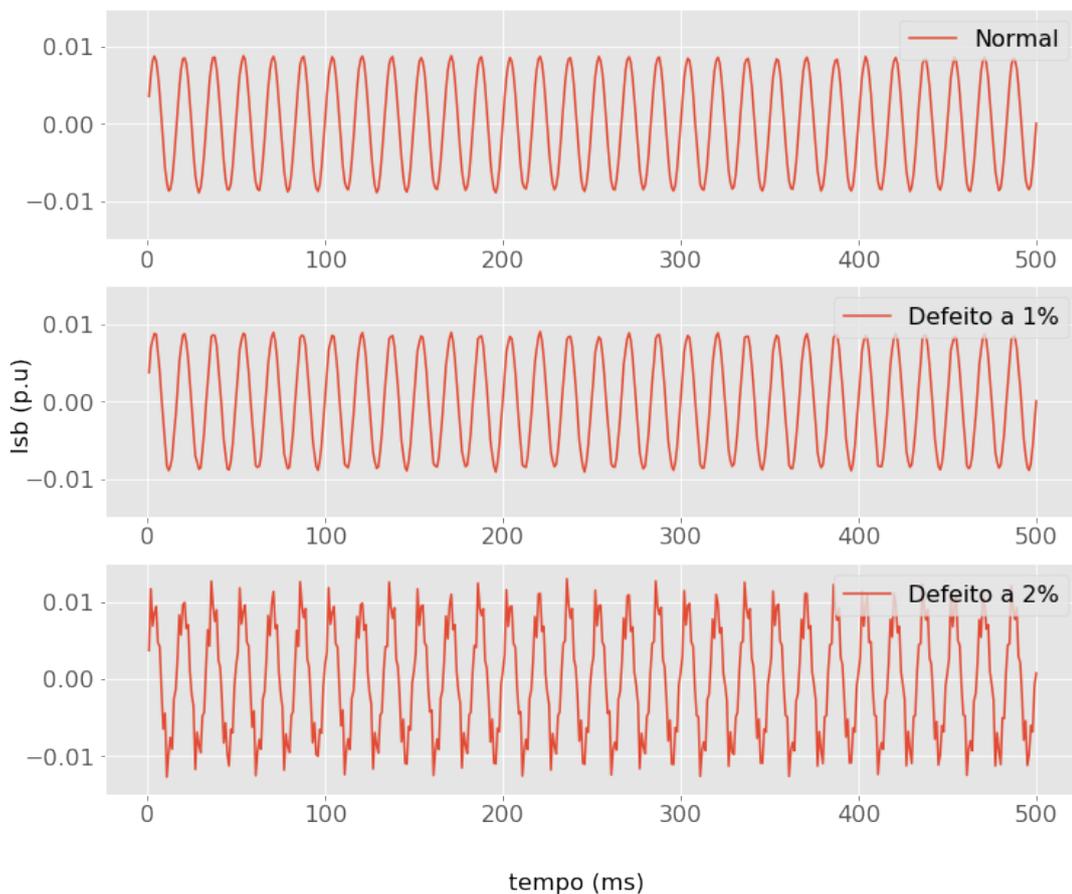


Figura 5.7 – Interpretação do *boxplot* genérico.

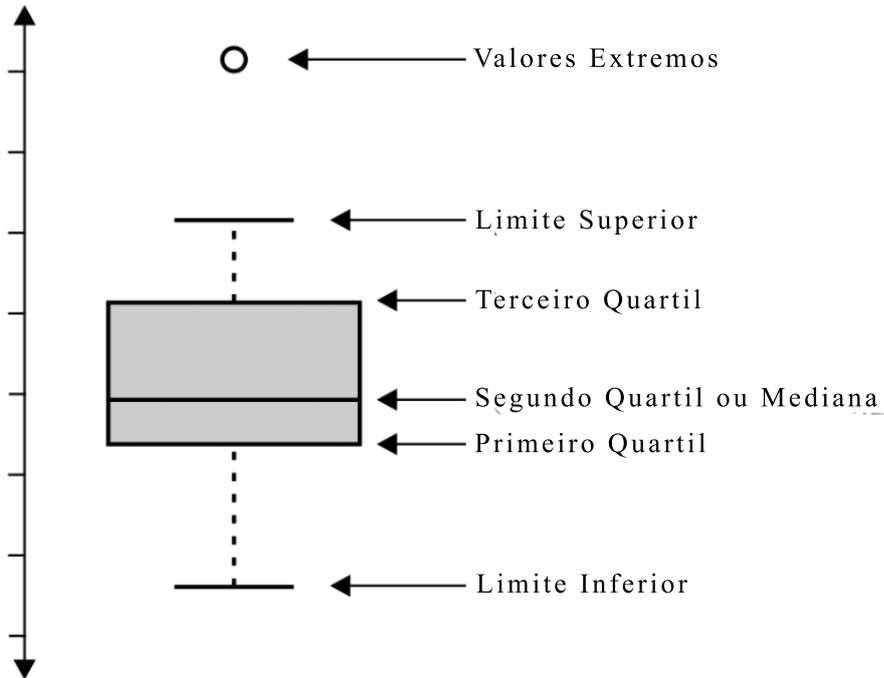


Figura 5.8 – Boxplot de amostras das classes normal (0% defeito) e 1% a 3% dos defeitos 1 e 2.

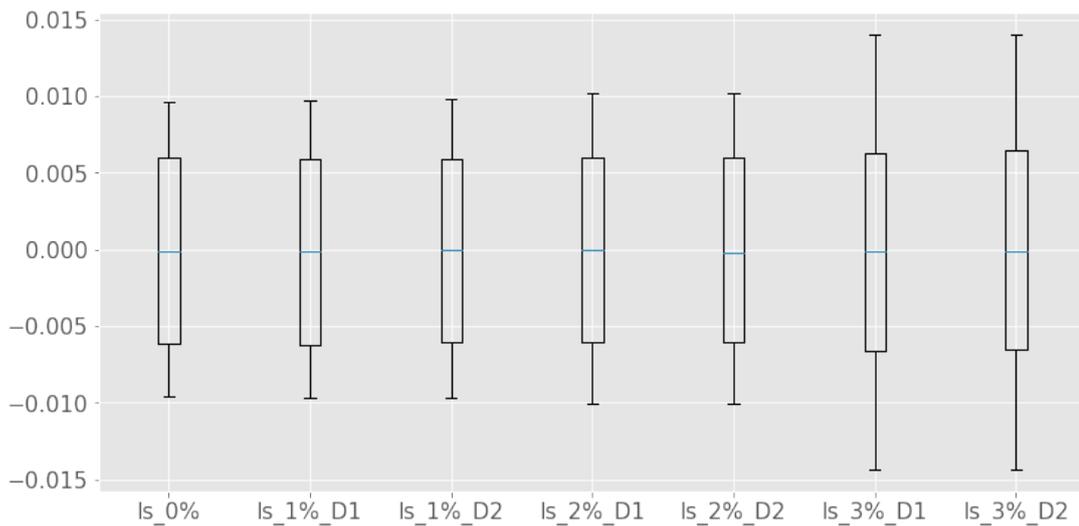
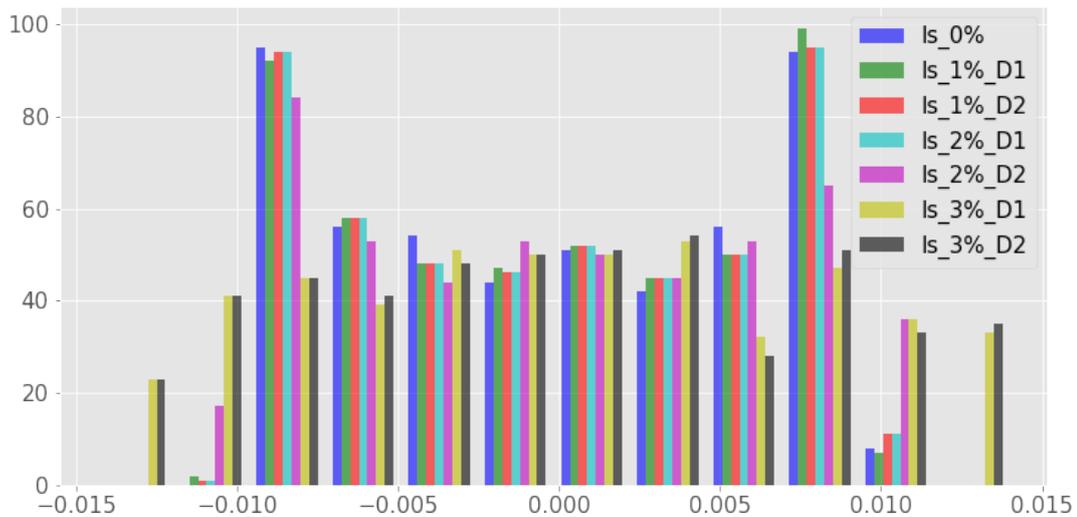


Figura 5.9 – Histograma de amostras das classes normal e 1% a 3% dos defeitos 1 e 2.



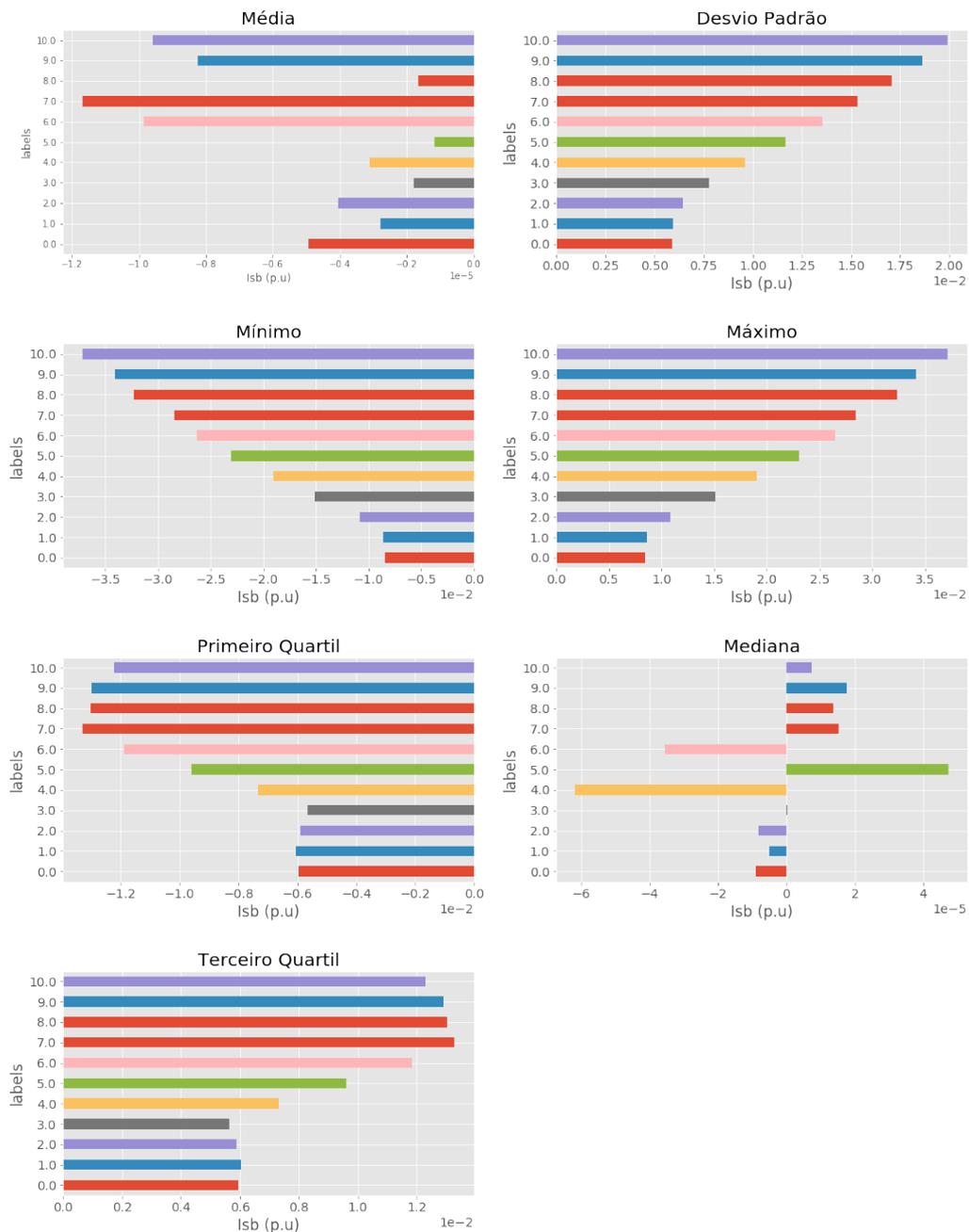
Devido à semelhança entre as classes de defeito 1 e 2, estas foram tratadas como um único defeito, uma vez que qualquer um dos defeitos quando detectado e classificado será necessário desligar a turbina e averiguar de forma mais profunda.

O problema foi então reduzido à condição do tipo normal e a condições do tipo defeituosa variando a intensidade de 1% a 10% de defeito. As classes foram então rotuladas de 0 a 10, onde 0 significa 0% de defeito (ou condição normal) e de 1 a 10 são os percentuais de defeito variando de 1% a 10%, respectivamente, que englobam variações do defeito 1 e 2, ou seja, a instância do defeito 1 a 1% de intensidade foi rotulada como sendo pertencente à classe 1, assim como à instância do defeito 2 a 1% de intensidade.

#### 5.4.2 Sumarização dos Dados por Rótulo

Nesta etapa, a dispersão das classes e como estão dispostas entre as métricas extraídas são apresentadas de forma visual. Na Figura 5.10 são apresentadas as métricas extraídas a partir da média das amostras por classe: média, desvio padrão, mínimo, máximo, primeiro quartil, mediana e terceiro quartil. Isso permite uma análise visual sobre as métricas extraídas em relação às respectivas classes do banco de dados, avaliando os limites numéricos de cada uma das classes.

Figura 5.10 – Métricas extraídas a partir da média das amostras por classe: média, desvio padrão, mínimo, máximo, primeiro quartil, mediana e terceiro quartil.



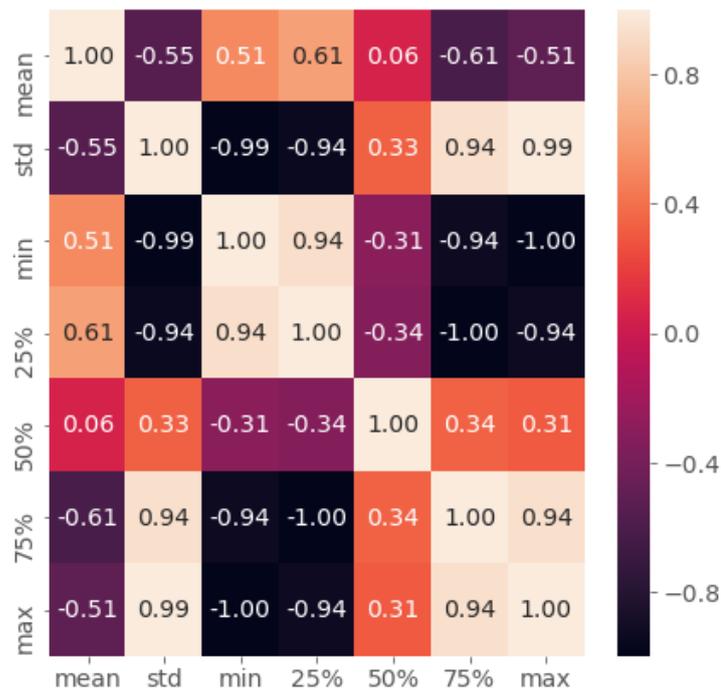
### 5.4.3 Correlação entre as Métricas Extraídas

A Figura 5.11 apresenta a correlação que cada métrica extraída tem com relação às demais. Quanto mais próximo à cor preta (-1) mais forte é a correlação inversa e quanto mais próximo à cor bege (+1) mais forte é a correlação direta entre os atributos.

Essa análise tem como objetivo principal verificar a relevância que cada atributo (métrica extraída) tem para o problema, reduzindo as redundâncias e conseqüentemente a dimensão do problema. Por exemplo, se é identificado que os atributos máximo e desvio

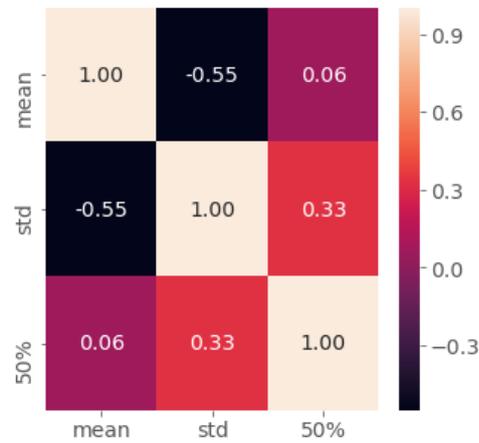
padrão estão fortemente correlacionados, utilizar ambos os atributos não traz informação adicional para a solução do problema. Assim como apresentado graficamente na Figura 5.10, note o comportamento de atributos fortemente correlacionados, o máximo e desvio padrão.

Figura 5.11 – Correlação das métricas extraídas a partir da média das amostras por classe.



A Figura 5.12 apresenta os atributos analisados que tiveram menor correlação direta/inversa, desvio padrão (std), média (mean) e mediana (50%). Porém, a média de um sinal de corrente alternada simétrico centralizado em zero tende a 0, assim a média acaba funcionando como um ruído de baixa significância para o problema. Na Figura 5.10 pode ser visualizado que a média mais alta diz respeito à classe 7 (percentual de defeito de 7%) com valor próximo de  $-0,12 \times 10^{-4}$ . De forma semelhante, a mediana também tende a zero. Dependendo de onde a janela de amostras seja cortada, a mediana tende a variar bastante, tornando-se também insignificante para o problema, uma vez que não existe um padrão definido da mediana em relação às classes analisadas. Restando assim, apenas o desvio padrão.

Figura 5.12 – Correlação das métricas selecionadas por eliminação daquelas com forte correlação.



Na Figura 5.13 são apresentados as métricas de baixa correlação. É visível a irregularidade da média e da mediana. Em relação ao desvio padrão, é nítida uma regularidade crescente do mesmo à medida que o percentual de falha é intensificado. Tal regularidade talvez seja justificável por se tratarem de dados decorrentes de simulação, modelo matemático, sem deformações de aquisição ou interferências externas e internas.

Figura 5.13 – Métricas selecionadas em relação as classes analisadas.

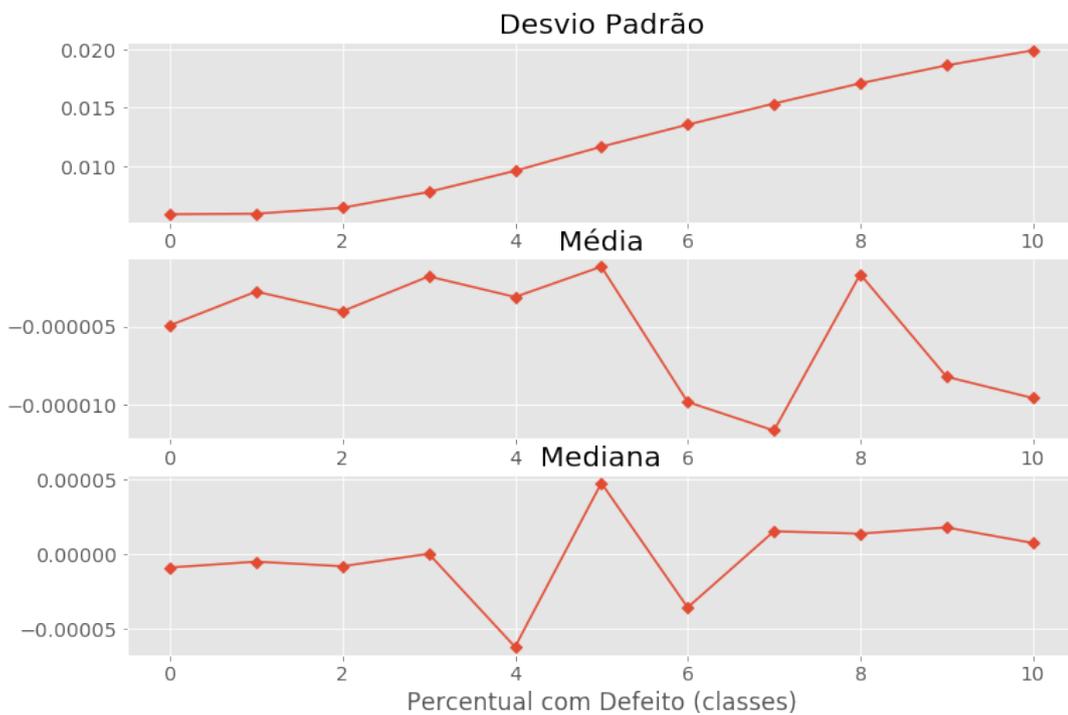


Tabela 5.1 – Desvio padrão médio de cada uma das classes e sua primeira diferença.

Classe	desvio padrão médio	Primeira diferença
0	0,005890	—
1	0,005936	0,000046
2	0,006438	0,000502
3	0,007798	0,001359
4	0,009602	0,001805
5	0,011666	0,002064
6	0,013552	0,001886
7	0,015352	0,001800
8	0,017088	0,001735
9	0,018622	0,001535
10	0,019909	0,001287

A Tabela 5.1 apresenta o desvio padrão médio (DPM) de cada uma das classes e a primeira diferença do DPM da classe atual com a classe anterior (elemento da linha anterior). Esta tabela serve para analisar o incremento do DPM ao longo das classes. Por exemplo, a classe 1 teve um incremento de 0,000046 com relação a classe 0, condição com 1% de defeito e condição normal, respectivamente.

Deve-se enfatizar que estes valores do DPM foram extraídos a partir da média de todas as amostras da respectiva classe. Isto é, apesar do DPM representar o valor médio das amostras, o mesmo não representa uma única amostra. Devido também à proximidade do DPM com sua classe vizinha, o atributo torna-se insuficiente para a classificação das amostras, sendo necessário utilizar algoritmos mais complexos para solução do problema.

## 5.5 Criação do Módulo Classificador

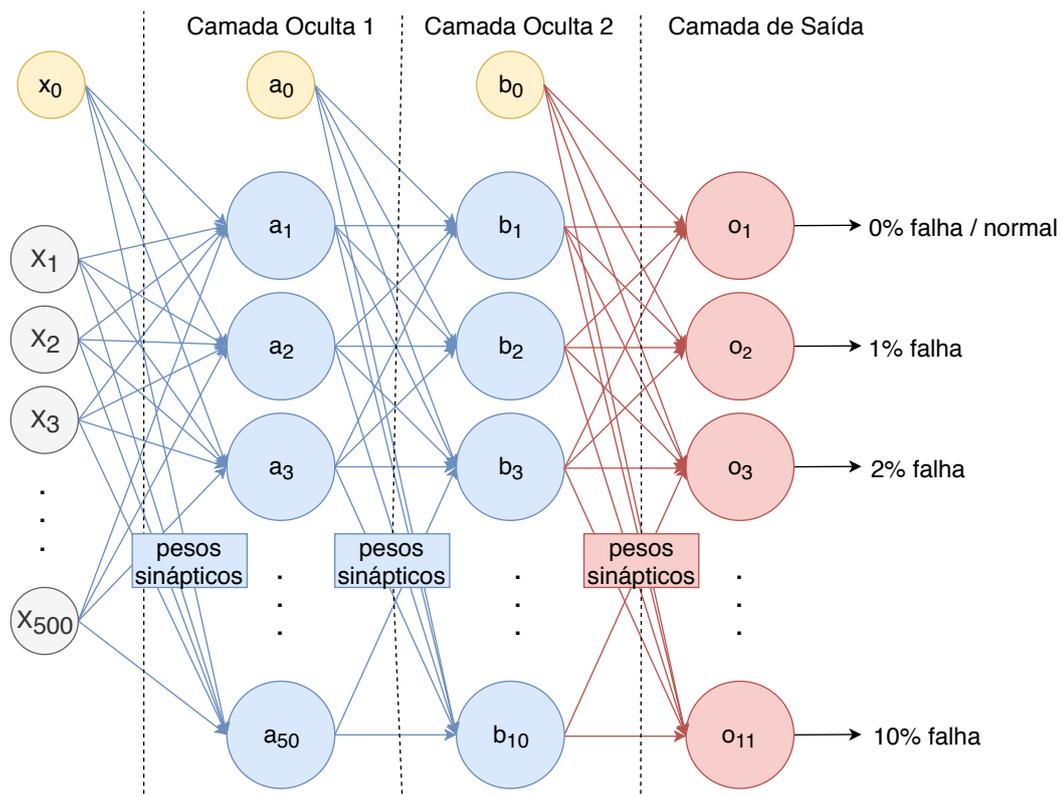
Com as informações obtidas a partir da análise anterior dos dados, foi desenvolvido um modelo que utilizava o desvio padrão médio como referência para classificação das novas amostras. O modelo MLP contendo apenas um neurônio de entrada, sendo utilizado apenas o desvio padrão extraído como atributo de entrada. Porém, os resultados obtidos não foram satisfatórios devido à proximidade interclasses do desvio padrão. Este processo é denotado na literatura como aprendizagem de máquina (do inglês, *Machine Learning*).

A partir dos resultados não satisfatórios obtidos utilizando a metodologia padrão de aprendizagem de máquina, onde o especialista é responsável por selecionar quais características são relevantes para o problema, foi então utilizada a metodologia de aprendizado profundo (do inglês, *Deep Learning*). Nessa técnica é passada à otimização do modelo a responsabilidade de aprender quais características são relevantes para a maximização dos resultados. Portanto, é passada a informação direto da fonte para o

modelo, neste caso leituras de corrente do estator PMSG, o modelo fica responsável por extrair a informação relevante para classificação das classes selecionadas. A dificuldade deste processo está em encontrar uma arquitetura: número de camadas, número de neurônios, funções de ativações, método de otimização, entre outros; que seja capaz de absorver informação dos dados, suficientes para a classificação correta de novas instâncias. Este processo é altamente dependente do conjunto de dados (fonte de informação) e do problema a ser resolvido (solução proposta).

Como apresentado na Figura 5.14, a arquitetura da MLP utilizada contém 500 pontos de entrada, os quais recebem os valores de corrente equivalente a meio segundo (0,5 s) de uma taxa de amostragem de 1 kHz. Em seguida, duas camadas ocultas, as quais possuem 50 e 10 neurônios respectivamente. Isso implica que suas matrizes de pesos têm as dimensões de (500, 50) e (50, 10), respectivamente, os quais implementam a função *tanh* (tangente hiperbólica), como função de ativação. Por fim, a camada de saída possui 11 neurônios, que implica em uma matriz de pesos na dimensão (10, 11), e implementa a função *Softmax* (OU-exclusivo) que classifica o sinal de entrada como um indicador de normalidade (ativação do neurônio 0), ou de uma porcentagem (1% a 10%) de espiras defeituosas (ativação do um neurônio de 1 ao 10). Em cinza são os valores de entrada ( $i_{sb}$ , 5.4), a instância. Em azul são as camadas ocultas e seus respectivos pesos sinápticos. Em vermelho é a camada de saída e seus respectivos pesos sinápticos. Em amarelo temos o *bias* de cada camada da rede neural.

Figura 5.14 – Arquitetura da MLP encontrada.



Uma das dificuldades na definição do modelo é a seleção dos parâmetros da arquitetura da rede MLP e os hiperparâmetros de treinamento, existe um número infinito de possibilidades que podem ser utilizados. Para a definição e otimização destes parâmetros foi utilizada uma função denominada *RandomizedSearchCV*, a qual está especificada na *Scikit-learn* [72] em sua versão *0.20.0*, é uma biblioteca de aprendizagem de máquina de código aberto para a linguagem de programação *Python* [67]. A função recebe como entrada uma lista de parâmetros e testa todas as possibilidades possíveis dado a lista, com objetivo de maximizar a quantidade de acertos do classificador MLP. Este processo é feito utilizando a validação cruzada *K-Fold* para extrair métricas de desempenho de cada modelo. A lista de parâmetros que é passada à função *RandomizedSearchCV* é refinada manualmente com objetivo de limitar o espaço de busca e maximizar a quantidade de acertos do modelo (*MLPClassifier*).

E para criação do modelo MLP foi utilizado a função *MLPClassifier* [73] que também é especificada na *Scikit-learn* [72]. Esta função cria uma MLP com os parâmetros passados. Após encontrados os parâmetros da MLP, foi criado um novo modelo com os parâmetros selecionados e foi realizado o treinamento do modelo utilizando o processo de treinamento em modo mini-lote (*min-batch*) com tamanho de 132 amostras e 5 mil épocas, para chaveamento de 2,5 kHz e 5 kHz. O modelo que utiliza ambos os chaveamentos foi utilizado apenas 3 mil épocas, devido ter mais dados para treinamento. O tamanho do *batch* e a quantidade de épocas também foi encontrado de forma empírica.

Os parâmetros instanciados durante o treinamento foram:

- *batch*: define a quantidade de amostras utilizadas a cada atualização de pesos da MLP. O valor utilizado foi 132 amostras;
- *alpha*: define o termo de regularização [74], que controla a intensidade da regularização, o qual tem como objetivo evitar problemas de *overfitting* através da limitação dos valores dos pesos de cada ligação entre neurônios. O valor final selecionado foi 0,0447161753369818 a partir de uma lista de valores gerados aleatoriamente dentro do intervalo (0,1] pela função *srand*;
- *learning\_rate*: taxa de aprendizagem que é responsável por controlar a velocidade (intensidade) de atualização dos pesos da MLP foi definida como um valor inicial igual a 0,002. E em modo *adaptive* que mantém a taxa constante enquanto o erro for decrescente, caso contrário a taxa de aprendizagem anterior é dividida por 5;
- *solver*: foi utilizado um método para otimização estocástica (do inglês, *A Method For Stochastic Optimization* - ADAM) [75], como otimizador dos pesos da MLP durante a fase de treinamento;
- *activation*: Foi utilizada a função *tanh* (tangente hiperbólica) como função de ativação das camadas escondidas. Outras funções foram avaliadas, *Identidade*, *Sigmoide* e a

Linear Retificada (*relu*), mas não tiveram resultado satisfatório;

- *random\_state*: Este parâmetro seta uma semente de aleatoriedade, permitindo que o processo de treinamento possa ser repetido em diferentes máquinas e sistemas operacionais. O valor utilizado foi 1001;
- *hidden\_layer\_sizes*: Como foram utilizadas apenas duas camadas escondidas, o valor final selecionado foi 50 e 10 neurônios, respectivamente. Diversos outros valores foram avaliados.

## 5.6 Validação do Módulo Classificador

O método de validação cruzada *K-Fold* foi utilizado durante o processo de seleção dos parâmetros da MLP. Em seguida, o conjunto de dados foi particionado, 80% para treinamento do modelo e 20% para teste. Essa proporção de dados de treinamento e teste foi definida de acordo com a quantidade de amostras disponíveis, de modo que o conjunto de teste contivesse amostras de todas as classes. Os resultados são apresentados no capítulo seguinte.

## 5.7 Considerações do Capítulo

Neste capítulo foi realizada uma análise exploratória sobre o conjunto de dados de amostras a 5 kHz de chaveamento a uma taxa de amostragem de 1kHz com a finalidade de obter informações para auxiliar na modelagem de uma solução para o problema.

O primeiro ponto analisado foi a viabilidade e a necessidade de se distinguir entre os defeitos espira-espira e espira-terra, defeito 1 e 2 respectivamente. Como não houve discrepância significativa entre os defeitos 1 e 2, optou-se por trata-los como um único defeito, reduzindo o problema a amostras sem defeito e amostras com defeito variando de 1% a 10% de intensidade, as quais foram rotuladas de 0 a 10 respectivamente. Essa redução de dimensionalidade do problema reduz fortemente a complexidade do módulo classificador. O segundo ponto foi verificar se haveria a necessidade de utilizar uma rede neural artificial para solucionar o problema. Soluções mais simples, tais como limiares aplicados ao desvio padrão e redes menos complexas foram utilizadas, porém os resultados foram insignificantes.

No Capítulo seguinte são discutidos os resultados obtidos até o momento e sua respectiva matriz de confusão e gráficos de treinamento, que permite uma análise visual dos resultados, principalmente quando se trata de um modelo de múltiplas classes, sendo possível verificar os falsos positivos e falsos negativos de cada classe.

## 6 Resultados

Como descrito no t3pico anterior, para cada caso descrito a seguir foi criada uma MLP com a mesma arquitetura. Em cada caso foram aferidas as m3tricas de desempenho. Segue abaixo a descri33o dos tr3s casos desenvolvidos:

- Caso 1: foram utilizadas apenas amostras com frequ3ncia de chaveamento de 5 kHz, onde para o conjunto de treinamento foram inclu3das apenas amostras com resist3ncia da falha de 0,2 ohms e para conjunto de teste 0,1 ohms;
- Caso 2: foram utilizadas apenas amostras com frequ3ncia de chaveamento de 2,5 kHz, onde para o conjunto de treinamento foram inclu3das apenas amostras com resist3ncia da falha de 0,1 ohms e para conjunto de teste 0,2 ohms;
- Caso 3: foram utilizadas ambas as frequ3ncias de chaveamento, o conjunto de 2,5 kHz e 5 kHz de chaveamento, onde para o conjunto de treinamento foram inclu3das amostras com resist3ncia da falha de 0,2 ohms para amostras de 2,5 kHz e 0,1 ohms para 5 kHz e para o conjunto de teste 0,1 ohms para 2,5 kHz e 0,2 ohms para 5 kHz.

A figura 6.1 apresenta uma interpreta33o visual do particionamento descrito acima. O conjunto de treinamento (80% do conjunto de dados) 3 representado em azul e o conjunto de teste (20% do conjunto de dados) em verde. Quanto 3s siglas, *RF* para resist3ncia da falha e *Chav* para a frequ3ncia de chaveamento.

Figura 6.1 – Representa33o dos casos desenvolvidos.

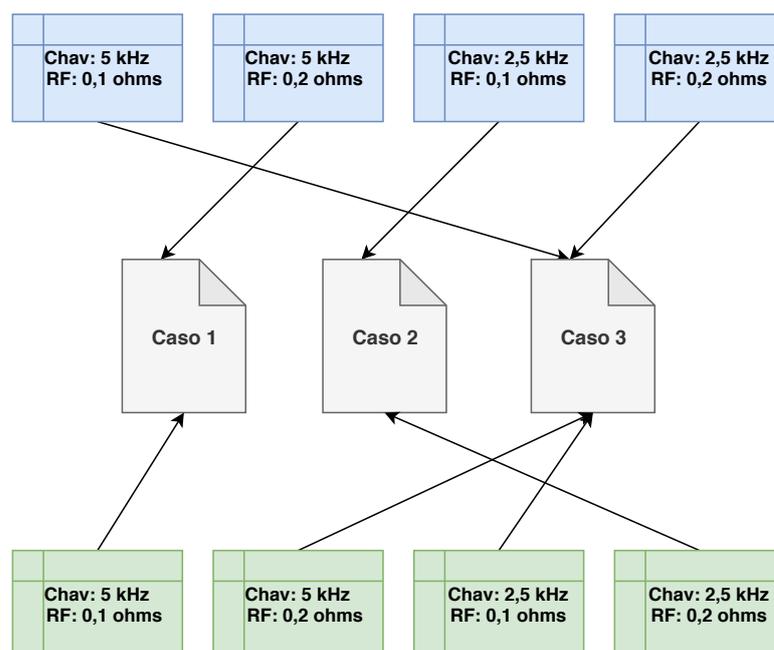




Figura 6.3 – Curva de acurácia do treinamento e validação para a frequência de chaveamento de 5 kHz.

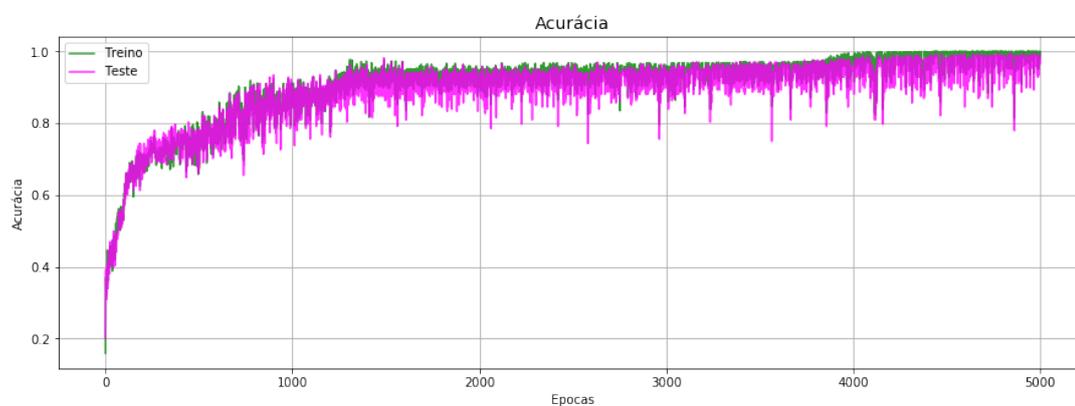
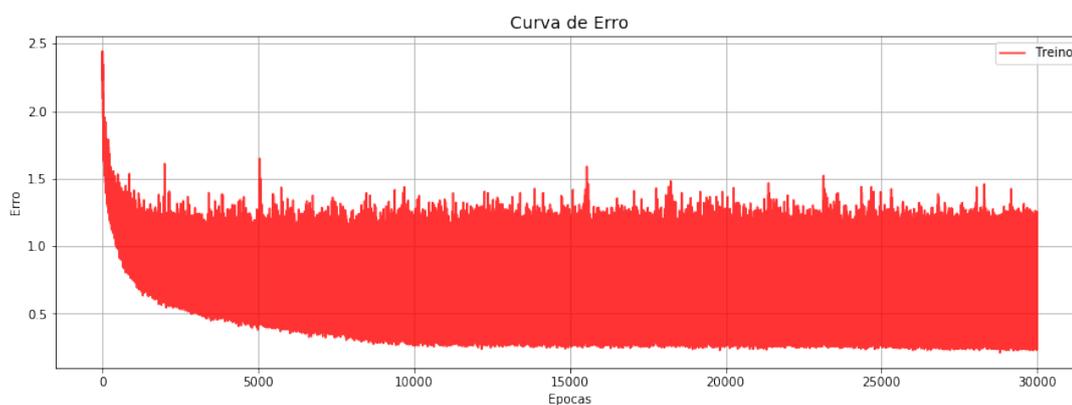


Figura 6.4 – Curva de erro para a frequência de chaveamento de 5 kHz.



## 6.2 Frequência de Chaveamento de 2,5 kHz

A Tabela 6.2 e a Figura 6.5 resumam os resultados obtidos com a base de dados de 2,5 kHz de chaveamento. Na Figura 6.6 é apresentada a acurácia do treinamento e validação ao longo das épocas, com acurácia global também de 97,62%. E na Figura 6.7 é apresentada a curva de decaimento do erro ao longo das épocas de treinamento.

Tabela 6.2 – Resultados obtidos para a frequência de chaveamento de 2,5 kHz.

Classe	Precisão	Recall	FScore
0	100%	100%	100%
1	100%	100%	100%
2	100%	100%	100%
3	100%	100%	100%
4	100%	100%	100%
5	100%	100%	100%
6	100%	100%	100%
7	100%	87,50%	93,33%
8	88,89%	100%	94,12%
9	93,75%	93,75%	93,75%
10	93,75%	93,75%	93,75%
média	97,85%	97,72%	97,72%

Figura 6.5 – Matriz de confusão para a frequência de chaveamento de 2,5 kHz.

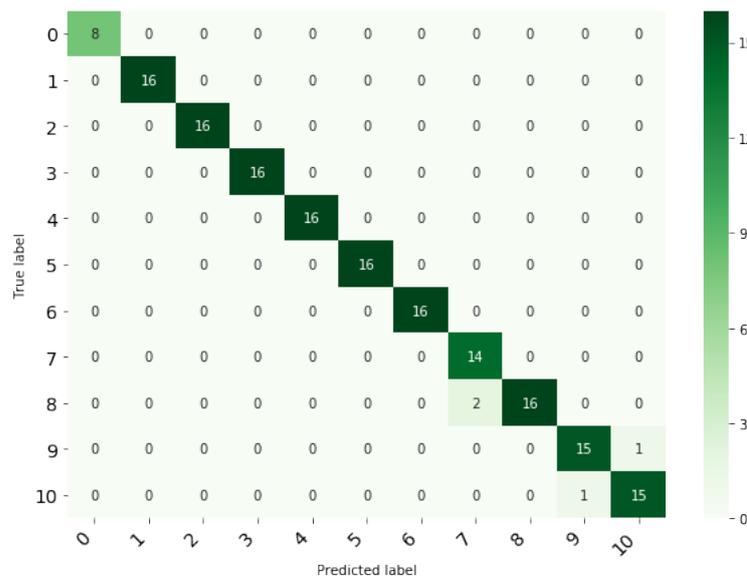


Figura 6.6 – Curva de acurácia do treinamento e validação para a frequência de chaveamento de 2,5 kHz.

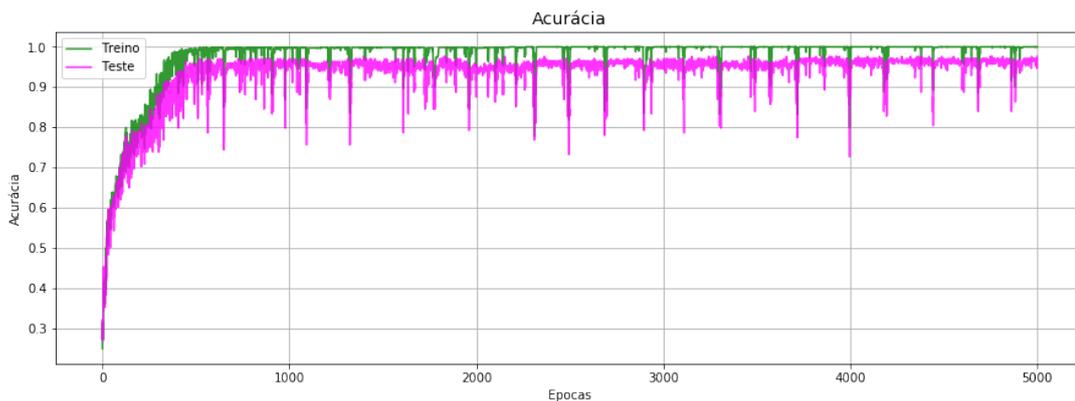
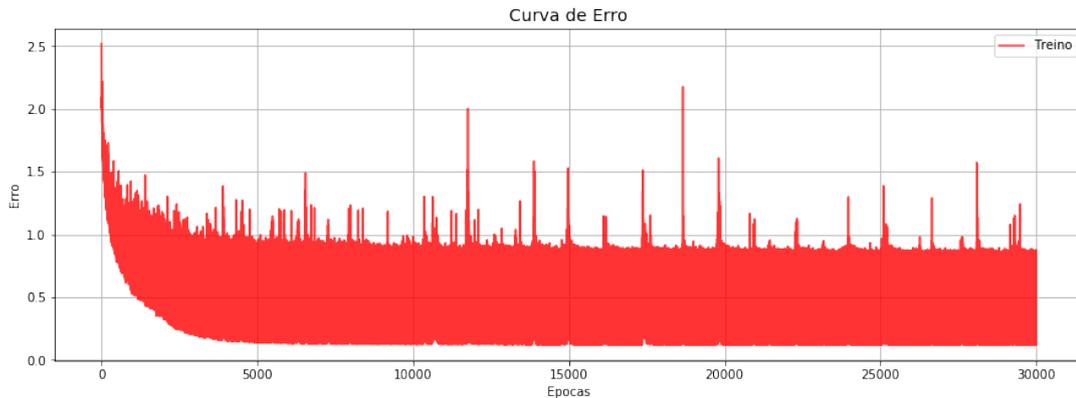


Figura 6.7 – Curva de erro para a frequência de chaveamento de 2,5 kHz.



### 6.3 Frequências de Chaveamento de 2,5 kHz e 5 kHz

A Tabela 6.3 e a Figura 6.8 resumam os resultados obtidos com as duas bases de dados, 2,5 kHz e 5 kHz de chaveamento. Na Figura 6.9 é apresentado a acurácia do treinamento e validação ao longo das épocas, com acurácia global de 92,86%. E na Figura 6.10 é apresentada a curva de decaimento do erro ao longo das épocas de treinamento.

Tabela 6.3 – Resultados obtidos para as frequências de chaveamento de 2,5 kHz e 5 kHz.

Classe	Precisão	<i>Recall</i>	<i>FScore</i>
0	88,89%	100%	94,12%
1	100%	100%	100%
2	100%	100%	100%
3	100%	100%	100%
4	100%	100%	100%
5	96,55%	87,50%	91,80%
6	90,00%	84,38%	87,10%
7	83,33%	93,75%	88,24%
8	80,00%	87,50%	83,58%
9	90,00%	84,38%	87,10%
10	93,33%	87,50%	90,32%
média	92,91%	93,18%	92,93%

Figura 6.8 – Matriz de confusão para as frequências de chaveamento de 2,5 kHz e 5 kHz.

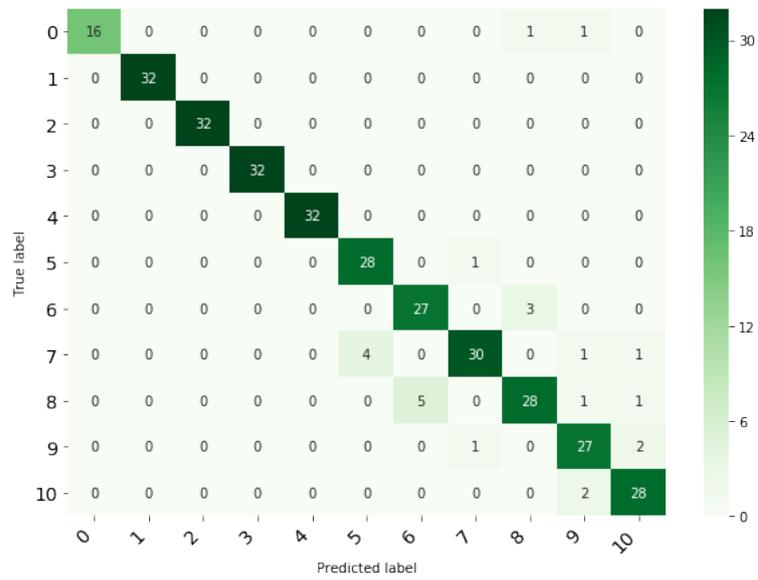


Figura 6.9 – Curva de acurácia do treinamento e validação para as frequências de chaveamento de 2,5 kHz e 5 kHz.

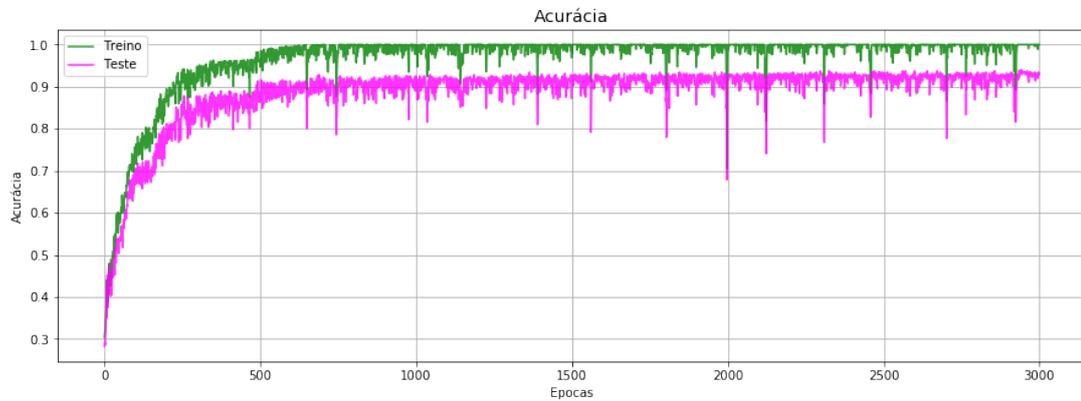
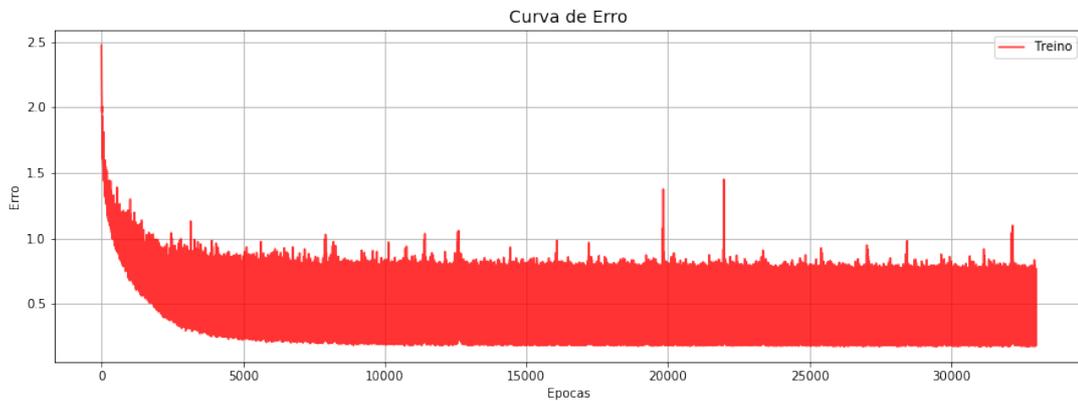


Figura 6.10 – Curva de erro para as frequências de chaveamento de 2,5 kHz e 5 kHz.



## 6.4 Considerações do Capítulo

Os resultados apresentados neste capítulo validam a proposta para o diagnóstico de curto-circuito entre as espiras do estator considerando os três casos desenvolvidos, e que a rede MLP desenvolvida para o modelo matemático do PMSG [64] é capaz de realizar a detecção e classificação do percentual de espiras com defeito e em condições normais sem a necessidade de uma etapa de pré-processamento de sinais para a extração de características, tornando o sistema mais rápido, possibilitando a predição em tempo real. Também foi possível avaliar que o mesmo modelo conseguiu realizar a predição com diferentes frequências de chaveamento, 2,5 kHz e 5 kHz.

De fato, os resultados retornam uma acurácia média similar à acurácia de outros trabalhos da literatura atual. Em [10], a acurácia variou entre 83% e 97% utilizando redes MLP e árvore de decisão. Já em [76] variou de 93% a 98,3% para detecção de falhas no estator a partir de leituras de corrente do motor e uma rede MLP. Neste, as falhas foram autoinfligidas em motores físicos e depois realizada a coleta dos dados.

Anteriormente, o artigo<sup>1</sup> contendo os resultados parciais desta pesquisa foi apresentado dia 16 de setembro de 2019 em Gramado, Brasil, no congresso internacional da IEEE *PES Innovative Smart Grid Technologies Latin America 2019*, organizado pela IEEE *Power and Energy Society* - (PES). Ele também foi disponibilizado na livraria digital do IEEE (*IEEE Explorer*) dia 11 de novembro de 2019.

No Capítulo seguinte é apresentada a conclusão do trabalho como um todo e perspectivas futuras da pesquisa.

---

<sup>1</sup> [77] B. A. Sa, C. M. V. Barros, C. A. Siebra, and L. S. Barros, “A Multilayer Perceptron-Based Approach for Stator Fault Detection in Permanent Magnet Wind Generators,” in *2019 IEEE PES Innovative Smart Grid Technologies Conference-Latin America - (ISGT Latin America)*. IEEE, Sep. 2019.

# 7 Conclusão

## 7.1 Contribuições

Durante o decorrer deste trabalho foi apresentada uma metodologia alternativa para a detecção de curto-circuito nas espiras do estator, espira-espira e espira-terra, a qual utiliza assinaturas de corrente de uma das fases do estator como fonte de informação para classificação das falhas a partir de uma rede MLP.

A metodologia apresentada é comumente utilizada em diversos outros problemas de aprendizagem de máquina, tais como classificação de imagem, classificação de áudio, etc. A contribuição deste trabalho neste ponto apenas reforça o uso desta metodologia em problemas de reconhecimento de padrões que envolvem classificação de falhas em motores elétricos. Como apresentado no capítulo 4, com o passar dos anos os autores têm buscado técnicas cada vez mais rápidas e precisas para detecção de falhas em motores elétricos. Isso deve-se ao fato de que o hardware normalmente destinado à controlar estas máquinas é limitado em memória e processamento.

Apesar das redes neurais artificiais terem um processo complexo e que exige alto consumo de processamento durante a fase de treinamento do modelo, uma vez treinada, o conhecimento fica armazenado nos pesos de cada camada. A partir dessas matrizes de pesos é realizada a predição de novas instâncias, ou seja, a predição de uma nova instância são multiplicações de matrizes e aplicação da função de ativação utilizada em cada camada.

Verificou-se que o mesmo modelo consegue realizar a predição com diferentes frequências de chaveamento, 2,5 kHz e 5 kHz, reforçando mais uma vez a flexibilidade desta metodologia.

Devido à limitações encontradas no decorrer desta pesquisa que serão descritas no tópico seguinte, foi disponibilizada a base de dados utilizada em repositório público no Kaggle em [78]. Assim, autores poderão citar o trabalho e utilizar a base de dados como base de comparação entre os métodos utilizados.

## 7.2 Limitações

Foi notado que dos trabalhos referenciados nesta pesquisa, aqueles autores que não desenvolveram sua própria base de dados, fizeram parceria com empresa privada. E em sua grande maioria, fizeram uso de modelos matemáticos em hardware e/ou software,

com suas próprias metodologias de desenvolvimento. Também foi realizada uma busca em repositórios públicos e enviado e-mail para alguns autores de artigos publicados com o mesmo objetivo de detecção de falhas em espiras do estator de máquinas elétricas. O objetivo era obter a base de dados e utilizá-la para realizar um comparativo direto entre o método utilizado nesta pesquisa e o método utilizado pelo outro trabalho sobre a mesma base de dados. Este tipo de teste é comum em problemas de aprendizagem de máquina e reconhecimento de padrões. Existem inclusive sites [79] onde as empresas privadas disponibilizam a base de dados e realizam um concurso daquelas metodologias que tiveram o melhor desempenho.

Apesar dos dados serem oriundos de uma única fonte de informação (simulador da turbina eólica baseada em PMSG) [64], foram simulados duas frequências distintas, com frequências de chaveamento de 2,5 kHz e 5 kHz e, com duas variações de resistência da falha de 0,1 ohms e 0,2 ohms. Além das variações de estados de condição normal e condição de defeito com evolução de 1% a 10% do percentual de espiras comprometidas. Os demais parâmetros foram mantidos fixos. Porém, mesmo com a variação de parâmetros, o simulador cria uma situação ideal para o classificador MLP, onde os sinais de falhas não apresentam deformidades além das esperadas, que podem vir a ocorrer devido a ruídos externos ou interferências no momento da coleta dos dados em um ambiente real. Essa ausência de interferências externas pode inclusive afetar a questão do *overfitting* porque, apesar das instâncias de treinamento e de testes não pertencerem ao mesmo conjunto, são geradas pela mesma fonte (modelo matemático da turbina eólica) e implicitamente possuem as mesmas características. Então para cada caso o conjunto de treinamento foi incluído apenas instâncias com resistência da falha diferente daquelas utilizada no conjunto de teste. Assim, cada modelo foi treinado a uma determinada resistência à falha e avaliada com outra.

Segundo [80], o aumento da frequência de chaveamento diminui o rendimento do inversor e aumenta o rendimento do motor. Geralmente, na maioria das aplicações é utilizada a frequência de 5 kHz e, quanto maior for a frequência de chaveamento (pulsção) do inversor, mais rápida será a degradação do sistema isolante. É sabido que em frequências de chaveamento menores ou iguais a 5 kHz a probabilidade de falha do isolamento é diretamente proporcional à frequência de chaveamento, ao passo que para frequências de chaveamento maiores que 5 kHz a probabilidade de falha do isolamento é diretamente proporcional ao quadrado da frequência de chaveamento [81]. Enquanto a resistência da falha, também denotada resistência do caminho, pode variar de 0 (rompimento total do isolamento) a infinito (operação normal).

Além dos pontos levantados até então, com base nos trabalhos avaliados ao longo desta pesquisa, técnicas que utilizam algum tipo de extração de características tendem a

ter um custo mais elevado de esforço computacional. Como reforçado por [60], técnicas de aprendizagem profunda são mais robustas a ruídos e podem aprender automaticamente as informações necessárias para a detecção e discriminação dos defeitos analisados, criando assim um sistema de monitoramento ponta-a-ponta (do inglês, *end-to-end*), notação utilizada para sistemas de aprendizagem de máquina que são responsáveis por todo o caminho da informação até sua classificação.

Em geral, os resultados obtidos foram satisfatórios dentro dos limites de tempo e recursos destinados à esta pesquisa. Com uma acurácia média de 92,86% para o caso 3 e 97,62% para os demais casos.

### 7.3 Trabalhos Futuros

Os trabalhos futuros teriam como base a melhoria da robustez do modelo, incrementando o tamanho da base de dados com mais amostras e abrangendo mais estados de funcionamento, tais como variações de carga, a partir do modelo utilizado [64]. Outra alternativa seria o acréscimo a partir de outras fontes de informação provenientes de motores físicos, tais como bancadas de teste em hardware ou plantas físicas de turbinas eólicas.

# Referências

- [1] F. Blaabjerg and Ke Ma, “Future on Power Electronics for Wind Turbine Systems,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 1, no. 3, pp. 139–152, sep 2013.
- [2] GWEC, *Global Wind Statistics*, 2019. [Online]. Available: <https://gwec.net/wp-content/uploads/2019/04/GWEC-Global-Wind-Report-2018.pdf>
- [3] T. W. Bank, *Population, total*, 2018. [Online]. Available: <https://data.worldbank.org/indicator/SP.POP.TOTL?end=2018>
- [4] ANEEL, *Banco de Informações da Geração*, 2020. [Online]. Available: <http://www2.aneel.gov.br/aplicacoes/capacidadebrasil/capacidadebrasil.cfm>
- [5] L. Chen, G. Xu, L. Liang, Q. Zhang, and S. Zhang, “Learning Deep Representation for Blades Icing Fault Detection of Wind Turbines,” in *2018 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, jun 2018, pp. 1–8.
- [6] G. Jiang, H. He, J. Yan, and P. Xie, “Multiscale Convolutional Neural Networks for Fault Diagnosis of Wind Turbine Gearbox,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 3196–3207, apr 2019.
- [7] Ørsted, *Hornsea Project One*, 2019. [Online]. Available: <http://hornseaprojectone.co.uk/en>
- [8] B. Kerres, “A comparison of wind turbine life-cycle cost and profits resulting from different maintenance strategies,” Master’s thesis, Chalmers University of Technology, 2010.
- [9] Y. Feng, Y. Qiu, C. J. Crabtree, H. Long, and P. J. Tavner, “Monitoring wind turbine gearboxes,” *Wind Energy*, vol. 16, no. 5, pp. 728–740, jul 2013.
- [10] G. H. Bazan, P. R. Scalassara, W. Endo, A. Goedtel, R. H. C. Palacios, and W. F. Godoy, “Stator Short-Circuit Diagnosis in Induction Motors Using Mutual Information and Intelligent Systems,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 3237–3246, apr 2019.
- [11] M. Barzegaran, A. Mazloomzadeh, and O. A. Mohammed, “Fault Diagnosis of the Asynchronous Machines Through Magnetic Signature Analysis Using Finite-Element Method and Neural Networks,” *IEEE Transactions on Energy Conversion*, vol. 28, no. 4, pp. 1064–1071, dec 2013.

- 
- [12] M. Eftekhari, M. Moallem, S. Sadri, and M.-F. Hsieh, “Online Detection of Induction Motor’s Stator Winding Short-Circuit Faults,” *IEEE Systems Journal*, vol. 8, no. 4, pp. 1272–1282, dec 2014.
- [13] D. C. Patel and M. C. Chandorkar, “Modeling and Analysis of Stator Interturn Fault Location Effects on Induction Machines,” *IEEE Transactions on Industrial Electronics*, vol. 61, no. 9, pp. 4552–4564, sep 2014.
- [14] M. Wolakiewicz, G. Tarchala, T. Orłowska-Kowalska, and C. T. Kowalski, “Online Stator Interturn Short Circuits Monitoring in the DFOC Induction-Motor Drive,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2517–2528, apr 2016.
- [15] L. Frosini, S. Zanazzo, and A. Albini, “A wavelet-based technique to detect stator faults in inverter-fed induction motors,” in *2016 XXII International Conference on Electrical Machines (ICEM)*. IEEE, sep 2016, pp. 2917–2923.
- [16] T. Ghanbari, “Autocorrelation function-based technique for stator turn-fault detection of induction motor,” *IET Science, Measurement & Technology*, vol. 10, no. 2, pp. 100–110, mar 2016.
- [17] A. Bellini, F. Filippetti, C. Tassoni, and G.-A. Capolino, “Advances in Diagnostic Techniques for Induction Machines,” *IEEE Transactions on Industrial Electronics*, vol. 55, no. 12, pp. 4109–4126, dec 2008.
- [18] M. B. K. Bouzid and G. Champenois, “New Expressions of Symmetrical Components of the Induction Motor Under Stator Faults,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 4093–4102, sep 2013.
- [19] N. Rama Devi, D. V. Siva Sarma, and P. V. Ramana Rao, “Detection of stator incipient faults and identification of faulty phase in three-phase induction motor – simulation and experimental verification,” *IET Electric Power Applications*, vol. 9, no. 8, pp. 540–548, sep 2015.
- [20] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*, ser. Electrical Engineering and Electronics. Taylor & Francis, 1998.
- [21] G. C. Stone, I. Culbert, E. A. Boulter, and H. Dhirani, *Electrical Insulation for Rotating Machines: Design, Evaluation, Aging, Testing, and Repair*, ser. IEEE Press Series on Power Engineering. Wiley, 2014.
- [22] S. Grubic, T. Habetler, B. Lu, and J. Aller, “Investigation on surge testing for winding insulation fault detection in an online environment,” in *2009 IEEE Energy Conversion Congress and Exposition*. IEEE, Sep. 2009.

- [23] W. E. Kuchenbecker, W. M. Rosa, and J. C. Teixeira, “PMSG fault identification applied to wind power,” *Electric Power Systems Research*, vol. 165, no. July, pp. 102–109, dec 2018.
- [24] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, may 2015.
- [25] C. A. Walford, “Wind turbine reliability :understanding and minimizing wind turbine operation and maintenance costs.” Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA (United States), Tech. Rep. March, mar 2006.
- [26] K. W. Klontz, T. J. E. Miller, M. I. McGilp, H. Karmaker, and P. Zhong, “Short-Circuit Analysis of Permanent-Magnet Generators,” *IEEE Transactions on Industry Applications*, vol. 47, no. 4, pp. 1670–1680, jul 2011.
- [27] Y. Lin, L. Tu, H. Liu, and W. Li, “Fault analysis of wind turbines in China,” *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 482–490, mar 2016.
- [28] Y. Qiu, H. Jiang, Y. Feng, M. Cao, Y. Zhao, and D. Li, “A New Fault Diagnosis Algorithm for PMSG Wind Turbine Power Converters under Variable Wind Speed Conditions,” *Energies*, vol. 9, no. 7, p. 548, jul 2016.
- [29] J. Serrano-González and R. Lacal-Aránategui, “Technological evolution of onshore wind turbines—a market-based analysis,” *Wind Energy*, vol. 19, no. 12, pp. 2171–2187, 2016.
- [30] C. M. V. Barros, “Estratégia de Controle para Melhoramento do Comportamento Dinâmico de Turbinas Eólicas Baseadas em PMSG ligadas à Rede Elétrica,” Doctoral thesis, Universidade Federal de Campina Grande, 2015.
- [31] *Electrical Academia*, 2019. [Online]. Available: <http://electricalacademia.com/electric-power/power-plants-work/>
- [32] M. L. Anders, “Análise transitória de faltas em geradores síncronos no contexto da energia eólica.” Ph.D. dissertation, Universidade de São Paulo, São Paulo, apr 2018.
- [33] L. Hansen, L. Helle, F. Blaabjerg, E. Ritchie, S. Munk-Nielsen, H. Bindner, P. Sørensen, and B. Bak-Jensen, *Conceptual survey of generators and power electronics for wind turbines*. DTU Library, 2002.
- [34] J. Slootweg, S. De Haan, H. Polinder, and W. Kling, “Voltage control methods with grid connected wind turbines: a tutorial review,” *Wind Engineering*, vol. 25, no. 6, pp. 353–365, 2001.

- [35] M. Chinchilla, S. Arnaltes, and J. Burgos, "Control of permanent-magnet generators applied to variable-speed wind-energy systems connected to the grid," *IEEE Transactions on Energy Conversion*, vol. 21, no. 1, pp. 130–135, Mar. 2006.
- [36] H. Li and Z. Chen, "Overview of different wind generator systems and their comparisons," *IET Renewable Power Generation*, vol. 2, no. 2, pp. 123–138, Jun. 2008.
- [37] A. Choudhary, D. Goyal, S. L. Shimi, and A. Akula, "Condition Monitoring and Fault Diagnosis of Induction Motors: A Review," *Archives of Computational Methods in Engineering*, vol. 26, no. 4, pp. 1221–1238, sep 2019.
- [38] C. Heising *et al.*, "Ieee recommended practice for the design of reliable industrial and commercial power systems," *IEEE Inc., New York*, 2007.
- [39] A. Glowacz and Z. Glowacz, "Diagnosis of stator faults of the single-phase induction motor using acoustic signals," *Applied Acoustics*, vol. 117, pp. 20–27, 2017.
- [40] J. Nunez, L. Velazquez, L. Hernandez, R. Troncoso, and R. Osornio-Rios, "Low-cost thermographic analysis for bearing fault detection on induction motors," 2016.
- [41] M. Eftekhari, M. Moallem, S. Sadri, and M.-F. Hsieh, "A novel indicator of stator winding inter-turn fault in induction motor using infrared thermal imaging," *Infrared Physics & Technology*, vol. 61, pp. 330–336, 2013.
- [42] A. K. Verma, S. Sarangi, and M. Kolekar, "Shaft misalignment detection using stator current monitoring," *International Journal of Advanced Computer Research*, vol. 3, no. 1, p. 305, 2013.
- [43] V. F. Pires, D. Foito, J. Martins, and A. Pires, "Detection of stator winding fault in induction motors using a motor square current signature analysis (mscsa)," in *2015 IEEE 5th international conference on power engineering, energy and electrical drives (POWERENG)*. IEEE, 2015, pp. 507–512.
- [44] S. Kumar, D. Goyal, and S. S. Dhama, "Statistical and frequency analysis of acoustic signals for condition monitoring of ball bearing," *Materials Today: Proceedings*, vol. 5, no. 2, pp. 5186–5194, 2018.
- [45] M. S. Othman, M. Z. Nuawi, and R. Mohamed, "Experimental comparison of vibration and acoustic emission signal analysis using kurtosis-based methods for induction motor bearing condition monitoring," *Przegląd Elektrotechniczny*, vol. 92, no. 11, pp. 208–212, 2016.
- [46] W. T. Thomson and M. Fenger, "Current signature analysis to detect induction motor faults," *IEEE Industry Applications Magazine*, vol. 7, no. 4, pp. 26–34, 2001.

- [47] N. Mehala and R. Dahiya, “Motor current signature analysis and its applications in induction motor fault diagnosis,” *International journal of systems applications, engineering & development*, vol. 2, no. 1, pp. 29–35, 2007.
- [48] S. Guedidi, S. Zouzou, W. Laala, M. Sahraoui, and K. Yahia, “Broken bar fault diagnosis of induction motors using mcsa and neural network,” in *8th IEEE Symposium on Diagnostics for Electrical Machines, Power Electronics & Drives*. IEEE, 2011, pp. 632–637.
- [49] S. Ahamed, M. Mitra, S. Sengupta, and A. Sarkar, “Identification of mass-unbalance in rotor of an induction motor through envelope analysis of motor starting current at no load.” *Journal of Engineering Science & Technology Review*, vol. 5, no. 1, 2012.
- [50] Z. Gao, C. Cecati, and S. X. Ding, “A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, Jun. 2015. [Online]. Available: <https://doi.org/10.1109/tie.2015.2417501>
- [51] Z. Gao, C. Cecati, and S. Ding, “A survey of fault diagnosis and fault-tolerant techniques part II: Fault diagnosis with knowledge-based and hybrid/active approaches,” *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2015. [Online]. Available: <https://doi.org/10.1109/tie.2015.2419013>
- [52] S. HAYKIN, *Introdução*, 2nd ed. Porto Alegre: Bookman, 2001, ch. 1-2, pp. 27–136.
- [53] D. Silverthorn, *Fisiologia Humana: Uma Abordagem Integrada*. Artmed Editora, 2017.
- [54] S. HAYKIN, *Perceptrons de Múltiplas Camadas*, 2nd ed. Porto Alegre: Bookman, 2001, ch. 4, pp. 183–281.
- [55] ———, *Teorema de Convergência do Perceptron*, 2nd ed. Porto Alegre: Bookman, 2001, ch. 3, pp. 163–168.
- [56] L. M. Popa, B.-B. Jensen, E. Ritchie, and I. Boldea, “Condition monitoring of wind generators,” in *38th IAS Annual Meeting on Conference Record of the Industry Applications Conference, 2003.*, vol. 3. IEEE, 2003, pp. 1839–1846.
- [57] B. Lu, Y. Li, X. Wu, and Z. Yang, “A review of recent advances in wind turbine condition monitoring and fault diagnosis,” in *2009 IEEE Power Electronics and Machines in Wind Applications*. IEEE, jun 2009, pp. 1–7.
- [58] F. P. García Márquez, A. M. Tobias, J. M. Pinar Pérez, and M. Papaelias, “Condition monitoring of wind turbines: Techniques and methods,” *Renewable Energy*, vol. 46, pp. 169–178, oct 2012.

- [59] R. Ibrahim and S. Watson, “Stator winding fault diagnosis in synchronous generators for wind turbine applications,” in *5th IET International Conference on Renewable Power Generation (RPG) 2016*. Institution of Engineering and Technology, 2016, pp. 29 (6 .)–29 (6 .).
- [60] H. Wang, S. Li, L. Song, and L. Cui, “A novel convolutional neural network based fault recognition method via image fusion of multi-vibration-signals,” *Computers in Industry*, vol. 105, pp. 182–190, feb 2019.
- [61] L. Song, H. Wang, and P. Chen, “Vibration-Based Intelligent Fault Diagnosis for Roller Bearings in Low-Speed Rotating Machinery,” *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 8, pp. 1887–1899, 2018.
- [62] C. Sun, M. Ma, Z. Zhao, and X. Chen, “Sparse Deep Stacking Network for Fault Diagnosis of Motor,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3261–3270, jul 2018.
- [63] *MathWorks*, 2019. [Online]. Available: <https://www.mathworks.com/products/simulink.htm>
- [64] C. M. V. Barros and L. S. Barros, “Modeling and analysis of stator interturn faults in permanent magnet synchronous machine,” in *2017 IEEE Power & Energy Society General Meeting*. IEEE, 2017, pp. 1–5.
- [65] S. Li and J. Li, “Output predictor-based active disturbance rejection control for a wind energy conversion system with pmsg,” *IEEE Access*, vol. 5, pp. 5205–5214, 2017.
- [66] C. M. Barros, G. T. Patrício, S. A. de Carvalho, and L. S. Barros, “Analysis of turn-to-ground and turn-to-turn stator faults in permanent magnet synchronous machine,” in *2019 IEEE PES Innovative Smart Grid Technologies Conference-Latin America (ISGT Latin America)*. IEEE, 2019, pp. 1–6.
- [67] P. S. Foundation, *Python Language Reference*, version 3.x. [Online]. Available: <http://www.python.org>
- [68] W. McKinney, *AQR Capital Management, pandas: a python data analysis library*. [Online]. Available: <http://pandas.sourceforge.net>
- [69] e. a. J. Hunter, *matplotlib: Python plotting*. [Online]. Available: <http://matplotlib.sourceforge.net/>
- [70] e. a. F. Perez, *IPython: an interactive computing environment*. [Online]. Available: <http://ipython.scipy.org>
- [71] T. Oliphant. [Online]. Available: <https://numpy.org/>

- [72] e. a. D. Cournapeau, *Scikits.learn: machine learning in Python*. [Online]. Available: <http://scikit-learn.sourceforge.net>
- [73] *SkLearn Neural Network*, version 0.20.0. [Online]. Available: [https://scikit-learn.org/stable/modules/generated/sklearn.neural\\_network.MLPClassifier.html](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html)
- [74] C. Cortes, M. Mohri, and A. Rostamizadeh, “L2 regularization for learning kernels,” 2012.
- [75] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014.
- [76] R. H. C. Palácios, I. N. da Silva, A. Goedel, and W. F. Godoy, “A novel multi-agent approach to identify faults in line connected three-phase induction motors,” *Applied Soft Computing*, vol. 45, pp. 1–10, 2016.
- [77] B. A. Sa, C. M. V. Barros, C. A. Siebra, and L. S. Barros, “A multilayer perceptron-based approach for stator fault detection in permanent magnet wind generators,” in *2019 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America)*. IEEE, Sep. 2019. [Online]. Available: <https://doi.org/10.1109/isgt-la.2019.8895013>
- [78] B. A. Sá, *Dataset: Wind turbine PMSG - Short-Circuit Fault*, 2020. [Online]. Available: <https://www.kaggle.com/brunoadonis/wind-turbine-pmsg-short-circuit-fault-mcsa>
- [79] kaggle, *Kaggle Competitions*, 2020. [Online]. Available: <https://www.kaggle.com/competitions>
- [80] G. WEG, *Motores de indução alimentados por inversores de frequência PWM*, 2019. [Online]. Available: <https://static.weg.net/medias/downloadcenter/h35/h10/WEG-motores-de-inducao-alimentados-por-inversores-de-frequencia-pwm-50029351-brochure-portuguese-web.pdf>
- [81] G. Ramírez, M. A. Valenzuela, M. D. Weaver, and R. D. Lorenz, “The impact of switching frequency on pwm ac drive efficiency,” in *2016 IEEE Pulp, Paper & Forest Industries Conference (PPFIC)*. IEEE, 2016, pp. 153–163.

