

**UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO**

**Alan Cechin Mello**

**DETECÇÃO DE DESBALANCEAMENTO DE MASSA NO ROTOR DE  
TURBINAS EÓLICAS UTILIZANDO ALGORITMOS DE  
APRENDIZADO PROFUNDO**

Santa Maria, RS

2022

**Alan Cechin Mello**

**DETECÇÃO DE DESBALANCEAMENTO DE MASSA NO ROTOR DE TURBINAS  
EÓLICAS UTILIZANDO ALGORITMOS DE APRENDIZADO PROFUNDO**

Monografia apresentada ao Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Engenharia de Controle e Automação**.

Orientador: Prof. Dr. Daniel Fernando Tello Gamarra

Santa Maria, RS  
2022

**Alan Cechin Mello**

**DETECÇÃO DE DESBALANCEAMENTO DE MASSA NO ROTOR DE TURBINAS  
EÓLICAS UTILIZANDO ALGORITMOS DE APRENDIZADO PROFUNDO**

Monografia apresentada ao Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Engenharia de Controle e Automação**.

**Aprovado em 19 de agosto de 2022:**

---

**Daniel Fernando Tello Gamarra, Dr. (UFSM)**  
(Presidente/Orientador)

---

**Frederico Menine Schaf, Dr. (UFSM)**

---

**Leonardo Ramos Emmendorfer, Dr. (UFSM)**

Santa Maria, RS  
2022

## RESUMO

# DETECÇÃO DE DESBALANCEAMENTO DE MASSA NO ROTOR DE TURBINAS EÓLICAS UTILIZANDO ALGORITMOS DE APRENDIZADO PROFUNDO

AUTOR: Alan Cechin Mello

ORIENTADOR: Daniel Fernando Tello Gamarra

De acordo com dados de 18 de março de 2022 da Associação Brasileira de Energia Eólica (ABEEólica) a média de geração de energia eólica anual abastece cerca de 29 milhões de residências por mês no Brasil, o que corresponde à 86,4 milhões de habitantes. O país possui 795 parques eólicos, 9.176 aerogeradores em operação em 12 estados. As turbinas eólicas são expostas a condições climáticas adversas e altamente variáveis. Devido a essas variações externas, as turbinas eólicas sofrem cargas que mudam constantemente, o que resulta em condições operacionais que levam a tensões mecânicas intensas. Com isso, a análise do aspecto de custo relativo à operação e manutenção dos parques existentes é muito importante para que os parques continuem funcionando de maneira efetiva na matriz energética brasileira. Dessa forma, a manutenção preditiva se mostra como uma das melhores alternativas para a equipe de manutenção poder se programar para a intervenção, evitando paradas prolongadas da linha de produção. Caracterizada pela medição e análise de condições da máquina é que se pode prever possíveis falhas em turbinas eólicas como a de desbalanceamento de massa entre as pás a qual é uma das falhas mais comuns nos aerogeradores. Sendo assim, o objetivo geral do trabalho consiste em desenvolver análise preditiva da falha de desbalanceamento de massa em rotores de turbinas eólicas mediante aplicação das redes neurais convolucionais *Fully Convolutional Network* (FCN) e *Residual Network* (Resnet) em um contexto de aprendizado supervisionado a partir da classificação de séries temporais univariadas de velocidade estimada de rotação de aerogerador no domínio da frequência gerado por meio de uma estrutura usando TurbSim/FAST/Simulink de uma turbina eólica de 1,5 MW para diferentes cenários de vento e parâmetros de desbalanceamento de pás. Sendo assim, verificou-se significativa diferença nos resultados de desempenho do algoritmo FCN ao utilizar equilíbrio de classes nos conjuntos em comparação com o algoritmo Resnet. Além disso, pode-se verificar que o algoritmo FCN obteve resultados de desempenho por classe e final com alta variabilidade, enquanto Resnet conseguiu obter mais consistência. Dessa forma, foi possível concluir que os experimentos utilizando conjunto de dados de classes equilibradas com o algoritmo de aprendizado profundo Resnet conseguiu obter os melhores desempenhos médios para os testes realizados.

**Palavras chaves:** aprendizado profundo, classificação de series temporais, aerogerador, desbalanceamento de massa.

## ABSTRACT

### DETECTION OF MASS UNBALANCE IN WIND TURBINE ROTOR USING DEEP LEARNING ALGORITHMS

AUTHOR: Alan Cechin Mello

ADVISOR: Daniel Fernando Tello Gamarra

According to March 18, 2022 data from the Brazilian Wind Energy Association (ABEEólica) the average annual wind power generation supplies about 29 million households per month in Brazil, which corresponds to 86.4 million inhabitants. The country has 795 wind farms, 9,176 wind turbines in operation in 12 states. Wind turbines are exposed to adverse and highly variable weather conditions. Due to these external variations, wind turbines experience constantly changing loads, resulting in operating conditions that lead to intense mechanical stresses. The analysis of the costs related to the operation and maintenance of the existing parks is very important for the parks to continue functioning effectively in the Brazilian energy matrix. Thus, predictive maintenance is one of the best alternatives for the maintenance team to be able to schedule the intervention, avoiding prolonged stops on the production line. Characterized by the measurement and analysis of machine conditions, it is possible to predict possible failures in wind turbines such as the unbalanced mass between the blades, which is one of the most common failures in wind turbines. Thus, the general objective of the work is to develop predictive analysis of mass imbalance failure in wind turbine rotors by applying the Fully Convolutional Network (FCN) and Residual Network (Resnet) in a supervised learning context from the classification of univariate time series of estimated wind turbine rotation speed in the frequency domain generated through a framework using TurbSim/FAST/Simulink of a 1.5 MW wind turbine for different wind scenarios and blade unbalance parameters. Therefore, there was a significant difference in the performance results of the FCN algorithm when using class balancing on the datasets compared to the Resnet algorithm. Furthermore, it can be seen that the FCN algorithm obtained per-class and final performance results with high variability, while Resnet was able to obtain more consistency. Thus, it was possible to conclude that the experiments using balanced class datasets with the Resnet deep learning algorithm was able to obtain the best average performances for the tests performed.

**Keywords:** deep learning, time series classification, wind turbine failure, mass unbalance.

## LISTA DE FIGURAS

Figura 1 – Matriz elétrica brasileira .....	9
Figura 2 - Operação manutenção corretiva.....	11
Figura 3 - Operação manutenção preventiva.....	11
Figura 4 - Operação manutenção preditiva.....	12
Figura 5 – Componentes turbina eólica .....	16
Figura 6 - Regiões de operação de turbinas eólicas.....	17
Figura 7 – Modelo de desequilíbrio de massa .....	18
Figura 8 - Geração de amostras sintéticas .....	21
Figura 9 - Efeito equilíbrio de classes em amostras .....	21
Figura 10 - Resultado de convolução discriminativa sobre o conjunto de dados GunPoint ....	26
Figura 11 – Rede FCN.....	28
Figura 12 – Rede Resnet.....	29
Figura 13 - Etapas para geração dos dados.....	33
Figura 14 - Observações de condição sem falha gerada no domínio do tempo .....	34
Figura 15 - Representação da condição sem falha no domínio da frequência .....	35
Figura 16 - Distribuição de classe do conjunto total .....	37
Figura 17 - Amostras por classe conjunto de teste .....	38
Figura 18 - Amostras por classe conjunto de treino/validação.....	38
Figura 19 - Amostras por classe conjunto de treino/validação balanceado.....	39
Figura 20 - Funcionamento validação <i>StratifiedKfold</i> .....	42
Figura 21 - Comparação entre experimentos utilizando a métrica acurácia.....	66
Figura 22 - Comparação entre experimentos utilizando a métrica F1- score.....	66

## LISTA DE TABELAS

Tabela 1 – Parâmetros configuração PSD .....	24
Tabela 2 - Definição de classes pelo agrupamento de cenários .....	36
Tabela 3 - Amostras em experimentos com quantidade de classes menores que 6.....	40
Tabela 4 - Classes em experimentos com quantidade de classes menores que 6.....	41
Tabela 5 – Parâmetros das redes neurais treinamento.....	43
Tabela 6 – Validação por classe FCN treinado sob 6 classes desbalanceadas .....	47
Tabela 7 - Teste por classe FCN treinado sob 6 classes desbalanceadas .....	48
Tabela 8 - Teste geral FCN treinado sob 6 classes desbalanceadas .....	49
Tabela 9 - Validação por classe Resnet treinado sob 6 classes desbalanceadas.....	50
Tabela 10 - Teste por classe Resnet treinado sob 6 classes desbalanceadas .....	51
Tabela 11 - Teste geral Resnet treinado sob 6 classes desbalanceadas .....	52
Tabela 12 - Validação por classe FCN treinado sob 6 classes balanceadas .....	53
Tabela 13 - Teste por classe FCN treinado sob 6 classes balanceadas.....	54
Tabela 14 - Teste geral FCN treinado sob 6 classes balanceadas .....	54
Tabela 15 - Validação por classe Resnet treinado sob 6 classes balanceadas.....	55
Tabela 16 - Teste por classe Resnet treinado sob 6 classes balanceadas .....	56
Tabela 17 - Teste geral Resnet treinado sob 6 classes balanceadas .....	57
Tabela 18 - FCN sob conjuntos desbalanceados com quantidade de classes inferior a 6 .....	58
Tabela 19 - Resnet sob conjuntos desbalanceados com quantidade de classes inferior a 6 .....	58
Tabela 20 - Desempenho por classe sob conjunto desbalanceado com 4 classes .....	59
Tabela 21 - Desempenho por classe sob conjunto desbalanceado com 3 classes .....	60
Tabela 22 - Pior desempenho por classe sob conjunto desbalanceado com 2 classes.....	60
Tabela 23 - Melhor desempenho por classe sob conjunto desbalanceado com 2 classes .....	61
Tabela 24 - FCN sob conjuntos balanceados com quantidade de classes inferior a 6 .....	61
Tabela 25 - Resnet sob conjuntos balanceados com quantidade de classes inferior a 6 .....	62
Tabela 26 - Desempenho por classe sob conjunto balanceado com 4 classes.....	63
Tabela 27 - Desempenho por classe sob conjunto balanceado com 3 classes.....	63
Tabela 28 - Pior desempenho por classe sob conjunto balanceado com 2 classes.....	64
Tabela 29 - Melhor desempenho por classe sob conjunto desbalanceado com 2 classes .....	64

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>9</b>
1.1	CONTEXTUALIZAÇÃO.....	9
1.2	DESCRIÇÃO DO PROBLEMA.....	10
1.3	OBJETIVOS.....	12
<b>1.3.1</b>	<b>Objetivos específicos .....</b>	<b>13</b>
1.4	ORGANIZAÇÃO DO TRABALHO .....	13
<b>2.</b>	<b>REFERENCIAL TEÓRICO.....</b>	<b>15</b>
2.1	ENERGIAS RENOVÁVEIS.....	15
2.2	ENERGIA EÓLICA.....	15
2.3	REGIÕES DE OPERAÇÃO DE AEROGERADORES.....	17
2.4	FALHA DE DESEQUILÍBRIO EM ROTORES DE AEROGERADORES .....	18
2.5	SMOTE .....	19
2.6	CLASSIFICAÇÃO DE SÉRIES TEMPORAIS .....	22
2.7	POWER SPECTRAL DENSITY .....	23
2.8	APRENDIZADO DE MÁQUINA E APRENDIZAGEM PROFUNDA .....	24
2.9	REDES NEURAIAS CONVOLUCIONAIS .....	25
2.10	FULLY CONVOLUTIONAL NETWORK.....	27
2.11	RESIDUAL NETWORK .....	29
<b>3.</b>	<b>METODOLOGIA .....</b>	<b>31</b>
3.1	AMBIENTE DE DESENVOLVIMENTO .....	31
3.2	ORIGEM DO BANCO DE DADOS .....	32
3.3	MODELAGEM DO EXPERIMENTO COM 6 CLASSES DEFINIDAS.....	35
<b>3.3.1</b>	<b>Preparação .....</b>	<b>35</b>
<b>3.3.2</b>	<b>Pré-processamento .....</b>	<b>37</b>
3.4	MODELAGEM COM OUTROS EXPERIMENTOS .....	40
3.5	MÉTODO DE AVALIAÇÃO E AJUSTE DE MODELOS .....	41
<b>4.</b>	<b>APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS.....</b>	<b>46</b>
4.1	CONJUNTO DE DADOS COM SEIS CLASSES .....	46
<b>4.1.1</b>	<b>Conjunto sem equilíbrio de classes .....</b>	<b>47</b>
<b>4.1.2</b>	<b>Conjunto com equilíbrio de classes.....</b>	<b>52</b>
4.2	CONJUNTOS COM QUANTIDADE DE CLASSES INFERIOR A SEIS .....	57
<b>4.2.1</b>	<b>Conjuntos sem equilíbrio de classes .....</b>	<b>57</b>

<b>4.2.2 Conjuntos com equilíbrio de classes.....</b>	<b>61</b>
<b>4.2.3 Comparação entre experimentos .....</b>	<b>65</b>
<b>5. CONCLUSÃO .....</b>	<b>67</b>
<b>REFERÊNCIAS .....</b>	<b>69</b>
<b>APÊNDICE A – FUNÇÃO DE CUSTO/PERDA PARA SEIS CLASSES DESBALANCEADAS .....</b>	<b>74</b>
<b>APÊNDICE B – MATRIZ DE CONFUSÃO TESTE PARA SEIS CLASSES DESBALANCEADAS .....</b>	<b>75</b>
<b>APÊNDICE C – FUNÇÃO DE CUSTO/PERDA PARA SEIS CLASSES BALANCEADAS .....</b>	<b>76</b>
<b>APÊNDICE D – MATRIZ DE CONFUSÃO TESTE PARA SEIS CLASSES BALANCEADAS .....</b>	<b>77</b>
<b>APÊNDICE E – COMPARAÇÃO GERAL EXPERIMENTOS TREINADOS COM DADOS DESBALANCEADOS .....</b>	<b>78</b>
<b>APÊNDICE F – COMPARAÇÃO GERAL EXPERIMENTOS TREINADOS COM DADOS BALANCEADOS .....</b>	<b>79</b>

# 1 INTRODUÇÃO

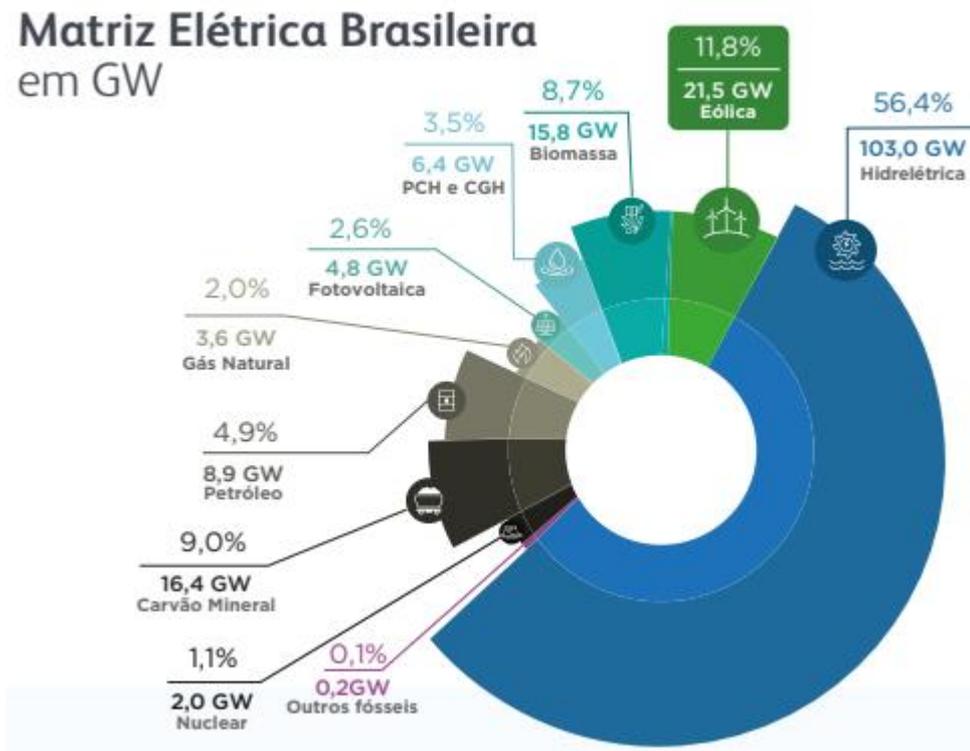
## 1.1 CONTEXTUALIZAÇÃO

De acordo com dados de 18 de março de 2022 da Associação Brasileira de Energia Eólica (ABEEólica) a média de geração de energia eólica anual abastece cerca de 29 milhões de residências por mês no Brasil, o que corresponde à 86,4 milhões de habitantes.

O país possui 795 parques eólicos contendo 9.176 aerogeradores em operação em 12 estados, o que totaliza 21,56 GW de potência eólica instalada, vide Figura 1.

Esses dados representam um aumento de 12,8% de potência instalada em relação a junho de 2021, quando a capacidade instalada era de 19,10 GW (ABEEólica, 2021).

Figura 1 – Matriz elétrica brasileira



Fonte: (ABEEólica, 2022).

De acordo com o relatório, divulgado em 04/04/2022, da Global Wind Energy Council (GWEC), o Brasil ocupa o 6º lugar entre os países com a maior capacidade total instalada de energia eólica *Onshore* em 2021. Em relação à capacidade instalada, o país foi superado pela Espanha (28,3 GW), Índia (40 GW), Alemanha (56,8 GW), EUA (134,3 GW) e China (310,6 GW). Em 2012, o Brasil ocupava o 15º lugar.

No ranking que contabiliza especificamente a nova capacidade instalada no ano de 2021, o Brasil aparece em terceiro lugar, tendo instalado 3,8 GW de nova capacidade, ficando atrás apenas de China e EUA.

O relatório também coloca o Brasil como um dos cinco principais mercados do mundo para novas instalações, junto com a China, Estados Unidos, Vietnã e Reino Unido. Juntos, esses cinco países representam 75,1% das instalações (GWEC, 2022).

Para gerar energia através do vento são necessárias grandes torres as quais possuem em seu topo turbinas que fazem a conversão da energia mecânica para energia elétrica. A cada uma das torres denominamos em geral como turbinas eólicas que, por sua vez, compõe as usinas/parques de energia.

## 1.2 DESCRIÇÃO DO PROBLEMA

As turbinas eólicas são expostas a condições climáticas adversas e altamente variáveis, incluindo calma a ventos fortes, calor tropical, relâmpagos, frio ártico, granizo e neve. Devido a essas variações externas, as turbinas eólicas sofrem cargas que mudam constantemente, o que resulta em condições operacionais altamente variáveis que levam a tensões mecânicas intensas.

Conseqüentemente, segundo Tchakoua et al. (2014) a indisponibilidade operacional de turbinas eólicas chega a 3% de sua vida útil. Além disso, os custos de Operação e Manutenção (O&M) podem representar de 10% a 20% do *Levelized Cost of Energy* (LCOE) ou traduzindo para o português “Custo Nivelado de Energia” que representa o custo total de energia para um projeto eólico, e esse percentual pode chegar a 35% para uma turbina no final da vida.

Em de Oliveira (2013), relata que as avarias em componentes como pás, caixa de engrenagens e o gerador elétrico originam elevados tempos de indisponibilidade e necessitam de equipamentos de difícil deslocamento e com pouca disponibilidade. Além disso, em Ribrant e Bertling (2007); Sinha e Steel (2015); Hahn (2007); Pérez (2013) é determinado que a falha

que tem maior impacto no quesito disponibilidade, ou seja, mantém as turbinas desligadas por maior tempo, está relacionado as pás, cuja substituição apresenta grande dificuldade.

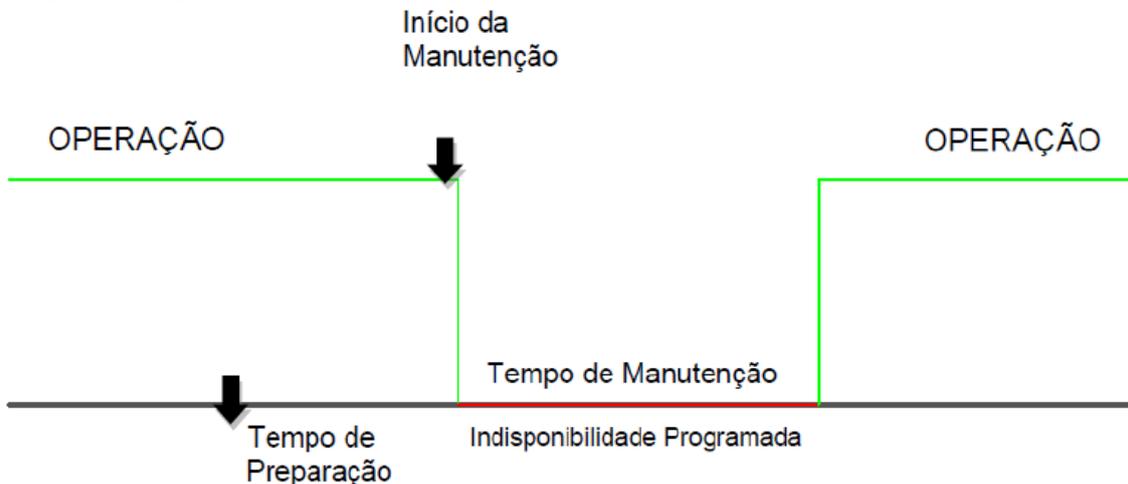
Uma estratégia de manutenção centrada na predição pode evitar o desligamento da máquina ou então reduzir o tempo despendido em manutenção e o consequente tempo de indisponibilidade operacional e assim, poder reduzir consideravelmente os custos atrelados a produção dessa fonte de energia. Em da Silva (2017), é ilustrado de forma comparativa as diferentes estratégias de manutenção e a capacidade de redução da preditiva, conforme Figura 2, 3 e 4.

Figura 2 - Operação manutenção corretiva



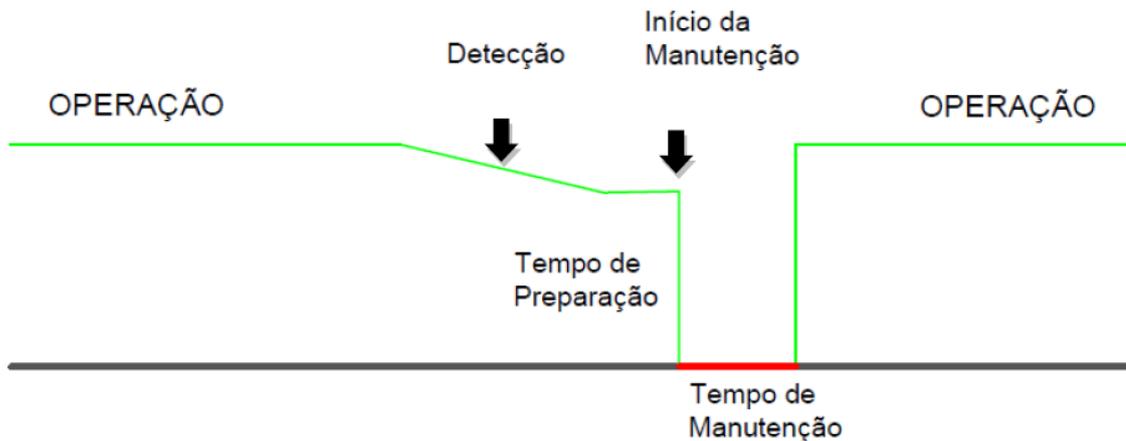
Fonte: (da SILVA, 2017). Adaptado de Operação de Parques Eólicos (PEREIRA, 2015).

Figura 3 - Operação manutenção preventiva



Fonte: (da SILVA, 2017). Adaptado de Operação de Parques Eólicos (PEREIRA, 2015).

Figura 4 - Operação manutenção preditiva



Fonte: (da SILVA, 2017). Adaptado de Operação de Parques Eólicos (PEREIRA, 2015).

Conforme Tchakoua et al. (2014), a tendência atual na indústria de energia eólica é o uso de turbinas eólicas cada vez maiores e em locais remotos, por exemplo situados no mar para ótimas condições de vento. Os fatores de tamanho e localização levaram a desafios de manutenção que são únicos em comparação com os sistemas tradicionais de geração de energia. Para lidar com essa realidade, os fabricantes de turbinas eólicas e *Condition Monitoring Systems* (CMS) devem desenvolver técnicas mais adequadas.

O objetivo futuro do CMS é continuar a minimizar os esforços exigidos dos operadores por meio do uso de algoritmos de *software* inteligentes e análise automatizada. Tais sistemas poderiam ser operados por pessoal treinado, mas não exigiria especialistas para a interpretação dos resultados porque um sistema de monitoramento inteligente será capaz de realizar operações de classificação e previsão. Portanto, o número de turbinas que um técnico é capaz de supervisionar pode dobrar.

### 1.3 OBJETIVOS

O objetivo geral do trabalho consiste em desenvolver a classificação de falha de desbalanceamento de massa em rotores de turbinas eólicas mediante aplicação dos algoritmos de redes neurais convolucionais 1D *Fully Convolutional Networks* (FCN) e *Residual Networks* (Resnet) de aprendizado profundo de máquina, conhecidos na literatura como *Deep Learning*,

para treinamento sob dados sem e com equilíbrio de classes utilizando a técnica *Synthetic Minority Over-Sampling Technique* (SMOTE).

A problemática a ser solucionada dada a forma como os dados foram concebidos é no contexto de classificação supervisionada a partir de um conjunto de amostras de velocidade estimada de rotação de aerogerador no domínio da frequência gerado por meio de uma estrutura usando TurbSim/FAST/Simulink para simular sinais elétricos gerados a partir de uma turbina eólica de 1,5 MW para diferentes cenários de vento e parâmetros de desbalanceamento de pás.

### 1.3.1 Objetivos específicos

Para alcançar o objetivo geral, será necessário:

- Compreender o contexto de funcionamento da problemática adotada;
- Estudar a natureza e característica dos conjuntos de amostras compostas de dados sequenciais no domínio da frequência;
- Estudar a técnica de balanceamento de amostras por classe SMOTE;
- Estudar abordagens existentes para resolução de problemas envolvendo classificação de séries temporais e em específico, a abordagem envolvendo aplicação de redes neurais convolucionais;
- Implementar os modelos de aprendizado profundo FCN e Resnet. E verificar como impactam seus desempenhos ao serem treinados sob conjunto de amostras balanceadas e desbalanceadas de classes;
- Definir métodos de validação para testar os experimentos de forma estatisticamente representativa;
- Definir métricas de avaliação de performance para os modelos na tarefa de classificação;
- Avaliar resultados e comparar desempenhos ao aplicar algoritmos e técnicas.

## 1.4 ORGANIZAÇÃO DO TRABALHO

Esse documento apresenta o desenvolvimento do projeto de pesquisa realizado em que inicialmente é apresentado o capítulo introdutório que relata o contexto da problemática adotada, bem como, os objetivos gerais e específicos do trabalho. O Capítulo 2 compreende as

revisões da literatura, afim de fundamentar os conhecimentos e conceitos principais visando o entendimento de futuras definições no processo de desenvolvimento do trabalho. O Capítulo 3 apresenta a metodologia utilizada, na qual discorre sobre os materiais que foram utilizados, bem como os métodos adotados para alcançar os objetivos propostos. No Capítulo 4 são mostrados os resultados obtidos, sendo discutidos e analisados os valores encontrados. Por fim, no Capítulo 5, são apresentadas as conclusões do trabalho.

## 2. REFERENCIAL TEÓRICO

Este Capítulo é dedicado à exposição e elucidação dos conceitos básicos necessários para a compreensão das soluções propostas, bem como, a citação de dados e estudos relacionados aos problemas e soluções abordados nesse trabalho.

### 2.1 ENERGIAS RENOVÁVEIS

A energia renovável vem de fontes que podem ser regeneradas ou reabastecidas naturalmente. As principais fontes são: água, vento, solar, biomassa e geotérmico (CENTER FOR CLIMATE AND ENERGY SOLUTIONS, 2021).

Costa et al. (2021, p. 1) ressalta que “As energias renováveis se tornaram as maiores aliadas para a geração de energia elétrica em todo o mundo livre de emissões de CO<sub>2</sub>”.

Segundo dados recentes da International Energy Agency (IEA), as adições anuais de capacidade renovável no mundo, bateram um novo recorde em 2021, aumentando 6% para quase 295 GW. Espera-se que a capacidade renovável aumente em mais de 8% em 2022, atingindo quase 320 GW.

No cenário nacional, a IEA prevê uma adição total de 12 GW de energias renováveis em 2022, sendo que a fonte solar fotovoltaica será responsável por mais de 8 GW, enquanto que a fatia restante virá de eólicas, hidrelétricas e outras fontes.

De acordo com dados divulgados no Seminário Energia, Desenvolvimento, Desafios e Oportunidades, promovido pela Federação das Indústrias do Rio de Janeiro (Firjan) e Cluster Tecnológico Naval do Rio de Janeiro em 30/05/2022. A produção de energia elétrica de origem renovável no Brasil é três vezes superior à mundial. No país, 83% da eletricidade produzida vem de fontes renováveis. Já no mundo, essa proporção chega a 27%.

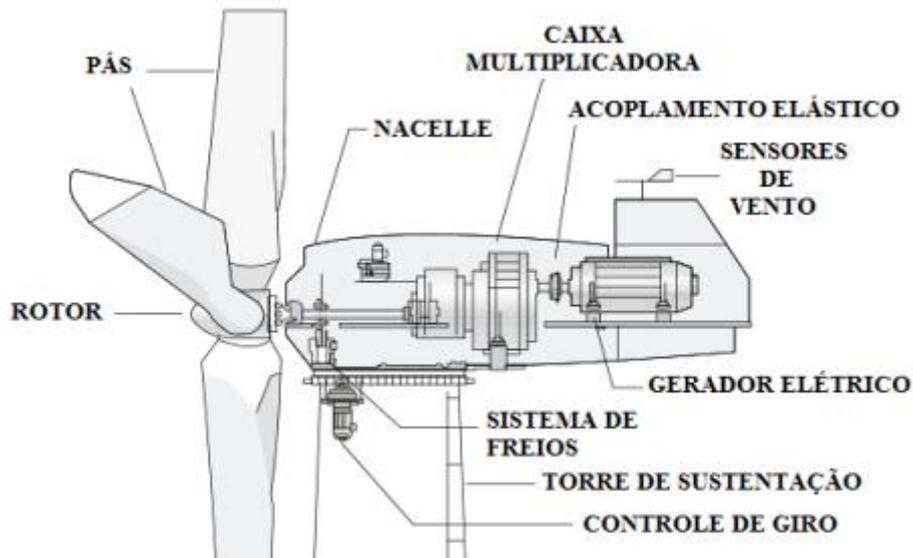
### 2.2 ENERGIA EÓLICA

Conforme Wizelius (2007), a energia eólica é a utilização do vento que é o ar em movimento. Como o ar tem massa ( $1,2 \text{ kg} / \text{m}^3$  a  $15 \text{ °C}$  e  $1 \text{ bar}$ ), o vento tem energia cinética. Essa energia pode ser transformada em energia elétrica, calor ou trabalho mecânico por meio de turbinas eólicas.

“Levando em consideração que a tecnologia hidrelétrica foi desenvolvida ao longo de século 20, a energia eólica está entre as tecnologias atuais para geração elétrica de fontes renováveis mais amplamente implementada em torno do mundo.” (COSTA et al., 2021, p. 1).

Uma torre de energia eólica possui diversos componentes, conforme Figura 5, os quais possuem sua relevância na tarefa de transformar energia mecânica em energia elétrica. Contudo, para a plena compreensão do trabalho em questão se faz necessário o descritivo dos conceitos de pás e rotor.

Figura 5 – Componentes turbina eólica



Fonte: Energia eólica: o que é, como funciona, vantagens e desvantagens. Portal Solar. Disponível em: <https://www.portalsolar.com.br/energia-solar-x-energia-eolica-precos.html>. Acesso em: 17 jul. 2022. Adaptado de (CBEE/UFPE, 2000).

As pás captam o vento, convertendo sua potência ao centro do rotor. São construídas em processo praticamente artesanal a partir de materiais como o plástico e a fibra de vidro. O desenho das pás emprega conceitos de aerodinâmica.

O rotor é o elemento de fixação das pás que transmite o movimento de rotação para o eixo de movimento lento. De acordo com de Oliveira (2013), existem diversas forças cíclicas que atuam sobre as pás de uma turbina eólica:

- Aerodinâmica (sustentação e arrasto das pás);
- Gravitacional (atração da terra);

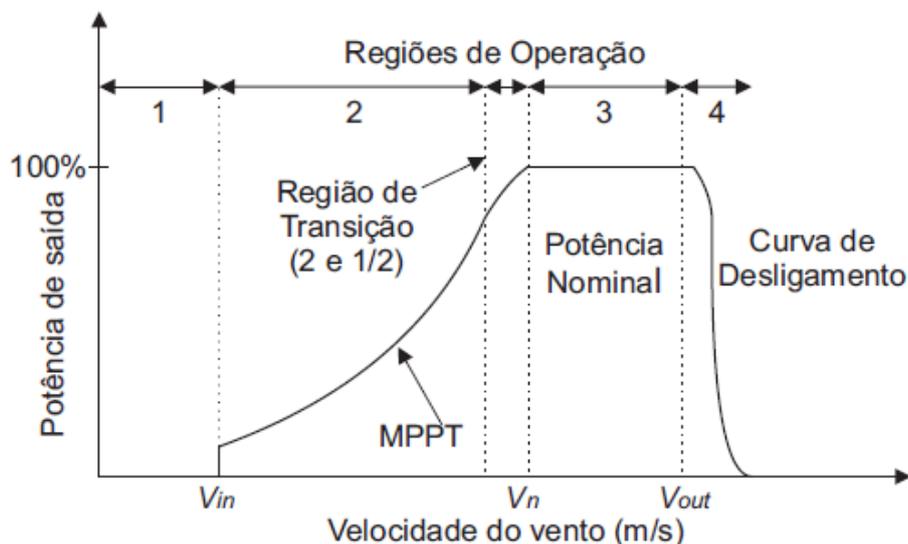
- Centrífuga (movimento rotacional das pás);
- Giroscópica (*yawing* durante a operação);
- Operacional (resultam do *pitching*, *yawing*, ligação ou desligamento do gerador).

### 2.3 REGIÕES DE OPERAÇÃO DE AEROGERADORES

Conforme Hübner (2021), em aerogeradores de velocidade variável com controlador regulado por *pitch*, as operações do aerogerador são divididas em regiões, conforme Figura 6. Isso ocorre devido as condições externas do meio ambiente, como por exemplo, efeitos da velocidade do vento turbulento nas condições de trabalho do aerogerador.

O foco do trabalho proposto é na região de operação 3 do aerogerador, chamada de Potência Nominal, é a etapa em que o aerogerador se encontra em condições nominais de potência e velocidade. Essa região se limita até uma determinada velocidade de vento estabelecida considerada arriscada para a integridade da turbina. Durante essa região como a velocidade do vento pode variar para que o aerogerador consiga manter uma velocidade constante, é realizado um controle de ângulo de *pitch* visando manter a geração de potência máxima nominal.

Figura 6 - Regiões de operação de turbinas eólicas



Fonte: (HÜBNER, 2021).

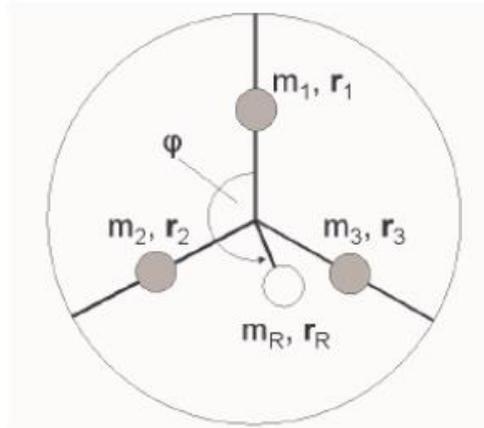
## 2.4 FALHA DE DESEQUILÍBRIO EM ROTORES DE AEROGERADORES

As falhas mais comuns em aerogeradores são citadas em de Souza (2018), no presente trabalho aprofunda-se a análise para a falha de desequilíbrio no rotor causado pelo desbalanceamento de massa entre as pás.

Com os problemas que surgem ao longo da vida útil das pás, variações de massa podem provocar o desbalanceamento do rotor, fazendo surgir vibrações indesejadas.

A Figura 7 mostra o modelo simplificado de um rotor de três pás, usado para estudar o comportamento das pás em situação de desequilíbrio de massas.

Figura 7 – Modelo de desequilíbrio de massa



Fonte: (CASELITZ; GIEBHARDT, 2005).

Em condições de operação normal, em que as massas ( $m_i$ ) estão perfeitamente equilibradas, verificam-se as condições representadas pela Equação 1.

$$m_1 r_1 = m_2 r_2 = m_3 r_3 \quad (1)$$

Nestas condições, o valor absoluto das forças centrífugas ( $FC_i$ ) é igual para cada pá e devido à geometria simétrica do rotor, a soma das três forças será zero, conforme a Equação 2.

$$FC_1 + FC_2 + FC_3 = 0 \quad (2)$$

Qualquer alteração dos parâmetros,  $(m_i, r_i)$ , origina o desequilíbrio do rotor. A massa  $(m_i)$  é alterada devido até mesmo uma situação em que algum objeto esteja preso à pá. Ao longo dos anos, a pá sofre com as intempéries climáticas, o que afeta seu peso e sua aerodinâmica, diminuindo o desempenho da turbina.

Dentre os principais problemas que podem afetar o desempenho e estrutura das pás, pode-se citar: a incidência de raios, onde ocorre perda de massa pela quebra; vazamentos de óleo, acúmulo de sujeira, insetos e gelo. Outro problema muito comum em regiões litorâneas, são os processos de erosão e corrosão que atacam, principalmente, os bordos de ataque das pás devido ao clima salino e a maresia que desgastam as regiões das pás que mantem contato direto com o vento.

O surgimento desses problemas pode afetar a aerodinâmica das pás, que está diretamente relacionada à potência gerada. (JÚNIOR, 2017)

Em estudos como de Sareen et al. (2014), Khalfallah e Koliub (2007) demonstraram a diminuição de 25% de potência elétrica gerada devido ao desbalanceamento de massa ocasionada pela erosão nos bordos de ataque das pás e acúmulo de sujeiras, insetos e gelo, respectivamente.

“O desequilíbrio de massas pode ser descrito por uma massa virtual  $(m_R)$  a uma distância virtual  $(r_R)$ . Esta massa irá induzir uma força centrífuga resultante no eixo do rotor” (de OLIVEIRA, 2013, p. 67).

Além disso, conforme Hubner (2021), o desequilíbrio de massa é diretamente proporcional à massa adicionada e ao quadrado da velocidade angular. Um rotor mal balanceado gera cargas excessivas que podem comprometer seriamente toda a estrutura do aerogerador.

## 2.5 SMOTE

Os algoritmos de aprendizagem de máquina precisam frequentemente lidar com conjuntos de dados altamente equilibrados. Um conjunto de dados de classificação desequilibrado significa dizer que as diferentes classes incluídas no conjunto de dados não estão uniformemente representadas (MANCHEV, 2021).

De acordo com o estudo realizado em Gupta et al. (2021) sob dados reais, 15% a 17% do total de reclamações pagas por seguros automóveis (terceiros) é fraudulento. Um grande

problema que é enfrentado na construção de um modelo de detecção de fraude para seguros é o desequilíbrio de dados. Por possuir menos dados de amostras fraudulentas, o modelo treinado é enviesado pelas observações não-fraudulentas. Dessa forma, possui desempenho insatisfatório em alegações fraudulentas, o que é indesejável.

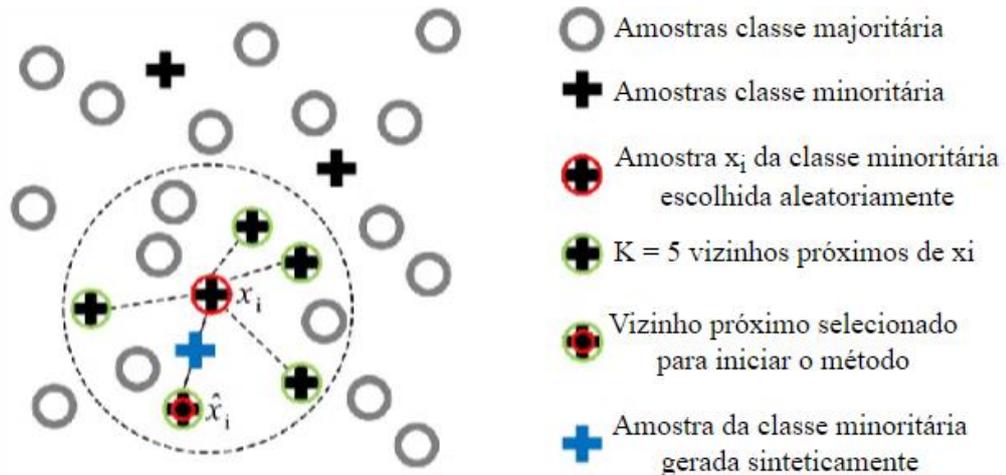
Caso não seja aplicada nenhuma técnica para conter esse problema, é muito provável que o modelo projetado tenha facilidade em identificar os dados referentes à classe majoritária (afinal, ele teve mais exemplos de dados dessa classe), mas tenha dificuldade na identificação dos dados da classe minoritária. Além disso, pode-se gerar uma má interpretação do modelo, por ele estar tendo um bom desempenho no acerto geral (acurácia). Essas e outras problemáticas em aplicações envolvendo dados desequilibrados estão descritas em Japkowicz e Stephen (2002).

Na literatura, existem diversos métodos para contornar a problemática do conjunto de dados desequilibrado, equalizando a quantidade das amostras por classe, enquadram-se basicamente em três abordagens: *Oversampling* (aumenta as observações da classe minoritária), *Undersampling* (diminui as observações da classe majoritária) e a combinação das anteriores.

Os métodos mais tradicionais dependem tipicamente da simples reamostragem por replicação ou exclusão de amostras do conjunto de dados original. No artigo de Chawla et al. (2002), propõe-se uma estratégia diferente em que a classe minoritária passa por sobreamostragem (*Oversampling*) através da geração de exemplos sintéticos.

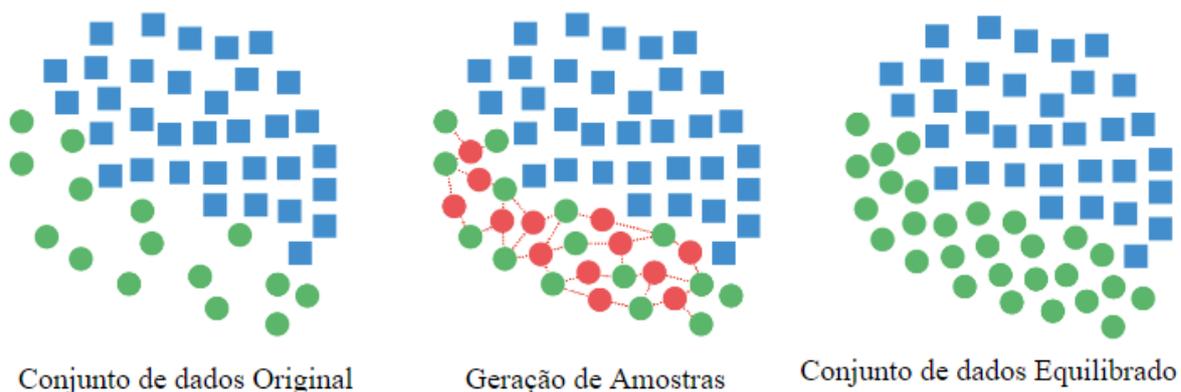
SMOTE baseia-se no algoritmo *K-Nearest Neighbors*, na qual uma amostra é gerada nas linhas que ligam um ponto e um dos seus  $k$  vizinhos mais próximos. Para isso, calcula-se a distância entre uma amostra e um de seus  $k$  vizinhos mais próximos e multiplica-se por algum valor aleatório no intervalo de  $(0, 1)$ , conforme é ilustrado em Figura 8. Isso é realizado para o outro vizinho mais próximo até o valor de  $k$ . O processo é repetido com a escolha de outra amostra e seus vizinhos até ocorrer o equilíbrio de classes ilustrado pela Figura 9.

Figura 8 - Geração de amostras sintéticas



Fonte: Autor. Adaptado de: [https://rikunert.com/smote\\_explained](https://rikunert.com/smote_explained). Acesso em: 19 jul. 2022.

Figura 9 - Efeito equilíbrio de classes em amostras



Fonte: Autor. Adaptado de (ORELLANA, 2020): Disponível em: <https://emilia-orellana44.medium.com/smote-2acd5dd09948>. Acesso em: 19 jul. 2022.

Em Chawla et al. (2002) realizou-se uma avaliação exaustiva do impacto do *Oversampling* baseada em SMOTE. Os resultados demonstram que os experimentos utilizando SMOTE superam consistentemente experimentos em que se utilizam apenas classificadores. Chawla et al., (2002) também explica porque uma simples sobreamostragem com replicação tem pior desempenho quando comparada com a sobreamostragem baseada no SMOTE. A simples replicação das observações minoritárias leva à redução das regiões de decisão, o que é o oposto do que é necessário para uma boa generalização. Por outro lado, o levantamento sintético da amostra expande as regiões de decisão, o que melhora o desempenho do classificador.

## 2.6 CLASSIFICAÇÃO DE SÉRIES TEMPORAIS

Nas últimas duas décadas, a classificação de séries temporais tem se revelado uma das problemáticas mais difíceis e desafiadoras na área de mineração de dados (YANG; WU, 2006; FAWAZ et al., 2019).

De acordo com Montanger (2021), a classificação de séries temporais consiste em prever a qual classe pertence uma determinada série temporal.

Classificação de séries temporais é amplamente aplicada em diferentes campos, tais como: reconhecimento de fala em Abdel-Hamid et al. (2013), ciência médica para diagnosticar doenças cardíacas em Wang et al. (2013) e reconhecimento de atividades humanas em Sadouk et al. (2018).

De acordo com Nakano e Chakraborty (2019), com a crescente utilização de vários sensores, o avanço das TIC (Tecnologia da Informação e Comunicação) e o menor custo de armazenamento, uma enorme quantidade de dados na forma de séries temporais são gerados e armazenados regularmente em vários domínios de aplicação.

Qualquer dado coletado de modo sequencial pode ser representado mediante uma série temporal. Uma série temporal pode ser compreendida como um conjunto ordenado de observações registradas cronologicamente (MORETTIN; TOLOI, 2004).

De acordo com Sadouk (2018), as séries temporais podem ser categorizadas em univariadas e multivariadas. Séries temporais univariadas possuem apenas uma única variável observada. Séries temporais multivariadas possuem duas ou mais variáveis observadas.

A representação de séries temporais consiste em utilizar um conjunto reduzido de valores para caracterizar as séries, de modo que as informações relevantes sejam preservadas ou até mesmo ressaltadas. No âmbito da literatura de séries temporais, as técnicas de representação estão organizadas de acordo com o tipo de transformação utilizada, sendo essas divididas em três abordagens: adaptativas, não adaptativas e baseadas em modelos.

Em Bagnall et al. (2012), demonstrou-se que transformar a representação para um espaço de dados alternativo onde as características discriminatórias são mais facilmente detectadas melhora o desempenho dos classificadores treinados em comparação com classificadores ajustados apenas no domínio do tempo. Ao transformar a representação em seu estudo, descreve que para os classificadores é mais fácil detectar no domínio *Power Spectrum*

(PS), mas menos óbvio no tempo, fornecendo assim, mais informações sobre as relações subjacentes que caracterizam o problema.

Sabendo que o ruído aleatório e as perturbações de alta frequência presentes nos dados das séries temporais podem interferir no processo de aprendizagem e que, é difícil captar características úteis com a presença de ruído nos dados brutos, em Rad et al. (2018), propõe-se a aplicação da Transformada Rápida de Fourier, conhecida no inglês como *Fast Fourier Transform* (FFT), para converter as séries temporais brutas num conjunto de sinais no domínio da frequência que servem como entrada para o processo de treinamento de Rede Neural Convolutiva (SADOUK, 2018).

Utilizou-se neste trabalho uma categoria de métodos da abordagem não adaptativa que consiste em transformar as séries em outros domínios. Dessa forma, utilizou-se o método *Power Spectral Density* (PSD).

## 2.7 POWER SPECTRAL DENSITY

De modo geral, os sistemas de monitoramento de condições trabalham sob sistemas e processos aleatórios, como os que são influenciados pela estocasticidade do vento, com isso, armazenam grandes quantidades de dados. Sendo assim, técnicas que revelam a periodicidade dos sinais podem contribuir com a análise da condição do aerogerador. Essa periodicidade pode ser analisada no domínio da frequência.

A densidade espectral - *Power Spectral Density* (PSD) define como a potência de um sinal ou série temporal está distribuída na frequência (HOWARD, 2004). É uma técnica de processamento de dados que realiza uma correlação entre as transformadas de Fourier de diferentes trechos no tempo do mesmo sinal, dessa forma, reduz a influência de sinais randômicos e avalia ocorrências periódicas presentes no sinal no domínio do tempo.

Ainda assim, na PSD ocorre efeitos do vazamento espectral (energia de uma frequência afeta outras) devido a técnica PSD funcionar melhor para sinais com número infinito de amostras, porém na prática é impossível, sendo utilizado um conjunto finito de amostras. Para reduzir os efeitos do vazamento espectral aplica-se a técnica do janelamento. O resultado disso é a redução da descontinuidade nas bordas do sinal (INSTRUMENTS, 2015).

Os parâmetros de configurações da técnica, segundo Hübner (2021) utilizado no presente trabalho, como o tipo da função de janelamento, o tamanho do sinal, a frequência de amostragem e a quantidade de sobre amostras são apresentados em Tabela 1.

Tabela 1 – Parâmetros configuração PSD

<b>Parâmetro</b>	<b>Valor</b>
Frequência de amostragem dos dados	50 Hz
Janelamento	Hann
Tempo de janelamento	30 segundos
<i>Overlap</i>	70% do tempo de janelamento

Fonte: Autor.

A Equação 3 apresenta a função de janelamento, conhecida como janelamento de Hann, a qual foi aplicada no presente trabalho.

$$w_{Hann}(n) = \frac{1}{2} \left( 1 - \cos \left( \frac{2\pi n}{N-1} \right) \right) \quad (3)$$

Em Gomes (2016) realizou-se estudo de modelo preditivo para detecção e caracterização de oscilações (*Oscillation, Detection and Characterization- ODC*) em sinais presentes em malhas de controle de processo industrial utilizando PSD para separar prováveis frequências de oscilação do sinal.

## 2.8 APRENDIZADO DE MÁQUINA E APRENDIZAGEM PROFUNDA

Aprendizado de máquina (do inglês: *Machine Learning*) é o estudo científico de algoritmos que computadores usam para realizar uma tarefa específica sem usar instruções explícitas, mediante extração de informações de dados e aplicação de algum tipo de modelo matemático.

Aprendizado Profundo, traduzido do inglês como é mais conhecido como *Deep Learning*, é um tipo de aprendizado automático, sendo uma subárea de Aprendizado de

Máquina que investiga técnicas para simular o comportamento do cérebro humano em tarefas como reconhecimento visual, reconhecimento de fala e processamento de linguagem natural.

Li et al. (2021) descreve que soluções envolvendo modelos de aprendizagem profunda para classificação de séries temporais, tipicamente aprendem de ponta a ponta, onde começam diretamente com dados brutos ou convertendo-os para o domínio da frequência. Fawaz et al. (2019) separa as abordagens em duas categorias principais: generativa e discriminativa. Os modelos aplicados no trabalho proposto estão dentro da abordagem discriminativa. Um modelo discriminativo de aprendizagem profunda é um classificador que aprende diretamente o mapeamento de uma série temporal original (ou as suas representações) e a saída com uma distribuição de probabilidade sobre as variáveis de classe num conjunto de dados.

Entre vários modelos de redes neurais profundas desenvolvidos para diferentes tarefas, *Convolutional Neural Network* (CNN) é a arquitetura mais amplamente aplicada para problemas de Classificação de Séries Temporais, provavelmente devido à sua robustez e menor tempo de treinamento em comparação com outras arquiteturas complexas (NAKANO; CHAKRABORTY, 2019).

## 2.9 REDES NEURAIAS CONVOLUCIONAIS

A arquitetura de uma Rede Neural Convolucional, do inglês *Convolutional Neural Network* (CNN), se baseia na alternância de camadas de convolução, camada que dá o nome a rede, camadas de normalização em lote e camadas de *pooling* (PENHA, 2018).

Em Fawaz et al. (2019) elenca-se duas razões potenciais para a elevada capacidade de generalização das CNNs sobre tarefas de classificação de séries temporais. Primeiro, tendo visto o sucesso das convoluções nas tarefas de classificação que requerem aprendizagem de características que são espacialmente invariantes num espaço bidimensional (tais como largura e altura em imagens), é natural pensar que a descoberta de padrões num espaço unidimensional (tempo) será uma tarefa mais fácil para as CNNs, exigindo assim menos dados com os quais aprender. A outra razão mais direta é a alta precisão das CNNs profundas em amostras de séries temporais e o seu sucesso em outros dados sequenciais tais como como reconhecimento da fala e classificação de frases, onde o texto e áudio, possuem semelhança aos dados das séries temporais, contendo uma ordenação temporal natural.

Uma convolução pode ser vista como a aplicação e o deslizamento de um filtro ao longo da série temporal. A convolução resulta na aplicação de uma média móvel com um deslizamento. De forma geral, pode ser definida pela Equação 4.

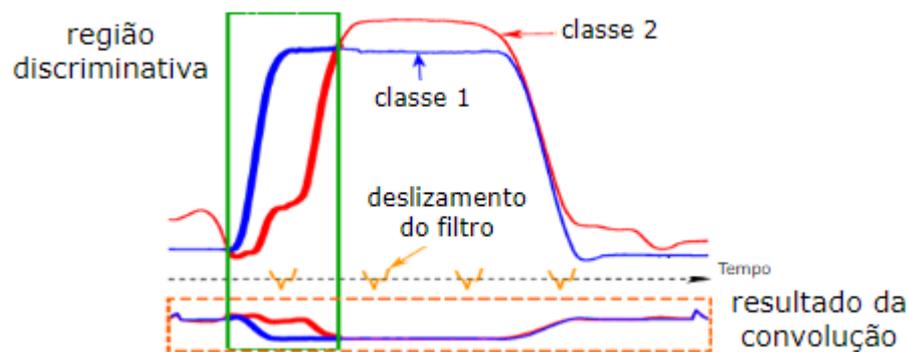
$$C_t = f(w \cdot x_{t-l/2:t+l/2} + b) \mid \forall t \in [1, T] \quad (4)$$

onde  $C$  denota o resultado de uma convolução aplicada sobre uma série temporal univariada  $X$  de comprimento  $T$  com um filtro  $w$  de comprimento  $l$ , um parâmetro de polarização  $b$  e uma função final não linear  $f$  como a *Rectified Linear Unit* (ReLU). Ela executa a operação sem alterar o tamanho de sua entrada, onde qualquer entrada de valor inferior a zero é definida como zero.

O resultado de uma convolução (um filtro) sobre uma série temporal de entrada  $X$  pode ser considerado como outra série temporal univariada que foi submetida a um processo de filtragem. Uma razão para aplicação de vários filtros sobre uma entrada de séries temporais é aprender múltiplas características discriminatórias úteis para a tarefa de classificação.

Entende-se como convolução a aplicação que permitirá o classificador discriminar facilmente entre as classes do conjunto de dados, conforme ilustrado em Figura 10.

Figura 10 - Resultado de convolução discriminativa sobre o conjunto de dados GunPoint



Fonte: Autor. Adaptado de (FAWAZ et al., 2019).

A convolução é geralmente precedida por uma operação de *pooling* que pode ser tanto local como global. Com uma operação de *pooling* global, as séries temporais serão agregadas ao longo de toda a dimensão temporal, resultando num único valor real. Isso é similar a aplicar

um *pooling* local com a janela de deslizamento de tamanho igual ao comprimento de entrada da série temporal. Normalmente é adotada uma agregação global para reduzir drasticamente o número de parâmetros do modelo, diminuindo assim o risco de sobreajuste (*overfitting*).

A camada discriminatória final toma a representação das séries temporais de entrada (o resultado das convoluções) e fornece uma distribuição de probabilidade sobre as variáveis de classe do conjunto de dados. Normalmente, isto é constituída por uma operação de *softmax*.

Sadouk (2018) faz uma revisão dos modelos de CNN aplicado a Classificação de Séries Temporais. Existem basicamente duas abordagens principais para a classificação de séries temporais por meio de redes neurais convolucionais. Numa abordagem, a CNN tradicional é modificada para aceitar séries temporais unidimensionais como entrada e, na outra abordagem, as séries temporais são convertidas numa imagem 2D para ser utilizada. Em Nakano e Chakraborty (2019) foi comparada essas duas abordagens e constatou-se que os resultados das redes profundas bidimensionais são muito melhores do que redes profundas unidimensionais com entrada de séries temporais brutas para a maioria dos conjuntos de dados testados.

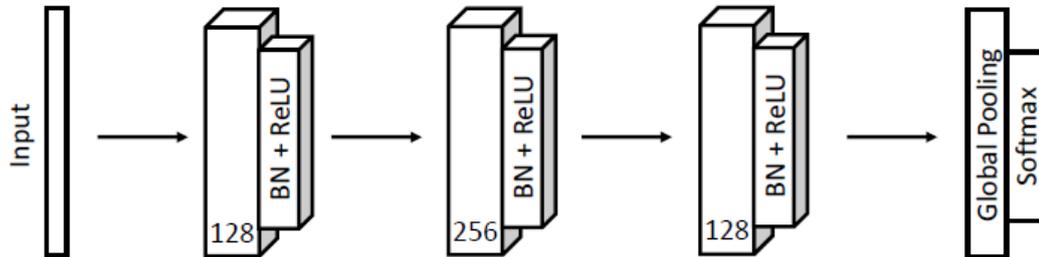
Em Fawaz et al. (2019), conjuntos de dados de séries temporais univariadas e multivariadas foram utilizadas para testar o desempenho de algumas redes neurais profundas na tarefa de classificação. ResNet e FCN obtiveram os melhores resultados.

Os algoritmos originais foram propostos por Wang et al. (2017) e estão implementados em Keras tensorflow e disponibilizados em GitHub com o endereço <<https://github.com/hfawaz/dl-4-tsc/blob/master/classifiers>>. Contudo, a biblioteca Tsai que fornece os algoritmos aplicados no trabalho em questão, adaptou a implementação original utilizando a biblioteca Pytorch. A arquitetura dos algoritmos implementada em Pytorch pela biblioteca Tsai pode ser acessada em: <<https://timeseriesai.github.io/tsai/models.FCN.html>> e <<https://timeseriesai.github.io/tsai/models.ResNet.html#ResNet>>.

## 2.10 FULLY CONVOLUTIONAL NETWORK

Rede Neural Totalmente Convolucional, traduzido do inglês *Fully Convolutional Network* (FCN), proposta por Wang et al. (2017), para classificação de séries temporais univariadas, ilustrada em Figura 11, é principalmente composta por camadas de convolução.

Figura 11 – Rede FCN



Fonte: (WANG et al., 2017)

A arquitetura do algoritmo é composta por três blocos de convolução e um bloco de saída. Cada bloco de convolução contém uma camada de convolução, uma camada de normalização de lotes (Ioffe e Szegedy, 2015) e a aplicação da função de ativação ReLU. A primeira convolução contém 128 filtros com um comprimento de filtro igual a 8, seguido de uma segunda convolução de 256 filtros com um comprimento de filtro igual a 5 que por fim, é alimentada a uma terceira e última camada convolutiva composta por 128 filtros, com um comprimento igual a 3 (FAWAZ, 2019).

A normalização em lotes, traduzido do inglês *Batch Normalization* (BN), é necessária em cada camada oculta da rede neural, pois acelera o processo de treinamento do modelo. Subtraindo o lote de dados da sua média e dividindo pela sua variação.

O bloco de convolução é dado pelas equações:

$$y = \mathbf{W} \cdot x + b \quad (5)$$

$$s = BN(y) \quad (6)$$

$$h = ReLu(s) \quad (7)$$

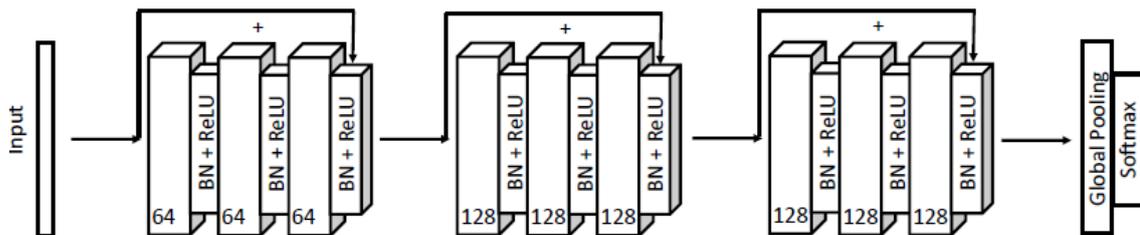
Após a série temporal ser extraída pelos blocos de convolução, as características extraídas são introduzidas na camada de *Global Average Pooling* (GAP) proposto em Lin et al. (2013) para produzir o resultado da classificação. Uma das características da FCN é não possui quaisquer camadas locais de *pooling* entre as convoluções, o que significa que o comprimento de uma série é mantido inalterado ao longo das convoluções.

Além disso, outra característica desta arquitetura é a substituição da camada tradicional final totalmente conectada por uma camada de GAP que reduz drasticamente o número de parâmetros numa rede neural.

## 2.11 RESIDUAL NETWORK

Rede Neural Residual, do inglês *Residual Network* (Resnet), originalmente proposta por Wang et al. (2017), ilustrada em Figura 12. É uma das redes mais profundas para classificação de séries temporais, das quais 9 são camadas convolucionais.

Figura 12 – Rede Resnet



Fonte: (WANG et al., 2017)

A rede do algoritmo é composta por três blocos residuais seguidos por uma camada GAP e uma *softmax* final. O bloco residual é composto inicialmente por três convoluções cuja saída é adicionada à entrada do bloco residual e depois alimentado para a camada seguinte. O número de filtros para todas as convoluções de cada bloco é fixado em  $\{64, 128, 128\}$ , com a função de ativação ReLU que é precedida por uma operação de normalização de lotes. Em cada bloco residual, o comprimento dos filtros é fixado em 8, 5 e 3 respectivamente para a primeira, segunda e terceira convolução (FAWAZ, 2019).

O bloco residual é dado pelas equações:

$$h_1 = \text{Block}_{k_1}(x) \quad (8)$$

$$h_2 = \text{Block}_{k_2}(h_1) \quad (9)$$

$$h_3 = \text{Block}_{k_3}(h_2) \quad (10)$$

$$y = h_3 + x \quad (11)$$

$$\hat{h} = \text{ReLU}(y) \quad (12)$$

$\text{Block}_k$  denota cada bloco convolucional com o número de filtros  $\{64, 128, 128\}$ .

A principal característica de ResNet é a ligação residual de atalho entre as camadas convolucionais consecutivas. A diferença entre outras redes convolucionais, como a FCN, é um atalho linear que é adicionado para ligar a saída de um bloco residual à sua entrada, permitindo assim o fluxo do gradiente diretamente através destas ligações, o que facilita o treinamento da rede neural profunda por reduzir o efeito do gradiente de fuga (FAWAZ, 2019).

### 3. METODOLOGIA

Neste capítulo são descritos os materiais e métodos escolhidos na execução do projeto, desde linguagens e interfaces de programação, *softwares*, estrutura do banco de dados, métodos e técnicas para realizar os experimentos de aprendizado de máquina profundo (*Deep learning*).

#### 3.1 AMBIENTE DE DESENVOLVIMENTO

Utilizou-se o *Google Colaboratory*, conhecido por *Colab*, como ambiente de desenvolvimento. O *Colab* é um serviço em nuvem gratuito hospedado pela empresa Google sem necessidade de qualquer configuração, para ter acesso é necessário apenas, conter um endereço eletrônico da empresa. Acessado via navegador, é uma ótima ferramenta para fins didáticos e documentação de conhecimento, pois contém uma série de células iterativas que podem conter textos explicativos ou códigos executáveis e suas respectivas saídas armazenadas.

Cada arquivo com células iterativas é chamado de *notebook*. O grande diferencial do serviço é que os *notebooks* do *Colab* são automaticamente armazenados na conta do Google Drive, sendo possível compartilhar os *notebooks* com outras pessoas, permitindo que elas façam comentários ou até editem o documento. Isso possibilita a difusão e evolução do conhecimento para o desenvolvimento da área de Aprendizado de Máquina e Inteligência Artificial. (GOOGLE, 2021)

No *Colab*, utilizou-se a programação de códigos na linguagem *Python* que atualmente está cada vez mais difundida em um modelo de desenvolvimento comunitário, aberto e gerenciado pela organização sem fins lucrativos *Python Software Foundation* também muito unida em aprimorar as suas funcionalidades.

Um dos aspectos que fazem a linguagem expandir-se para áreas do conhecimento cada vez mais específicas são as bibliotecas ou módulos de código aberto (*open source*) que foram desenvolvidas com o intuito de aprimorar as funcionalidades e tarefas a serem executadas. No projeto proposto foram utilizadas as seguintes bibliotecas/módulos para os devidos fins descritos:

- Pandas: Manipulação de dados tabulares importando-os de arquivos do Excel;
- Math: Operações matemáticas;
- Numpy: Álgebra linear para operações com matrizes e vetores;
- Seaborn :Visualização de dados;
- Scikit-learn: Ferramentas para repartição/divisão de dados para etapa de treinamento, validação e teste. E funções para avaliação do desempenho dos modelos nas tarefas de classificação;
- Imbalanced-learn: Ferramenta para tratar conjunto de dados com classes desequilibradas que serão submetidos a modelos de classificação;
- Tsai: aplicação de algoritmos de aprendizagem profunda (*deep learning*) voltado a resolver tarefa de classificação e previsão de séries temporais;
- Google.colab – Biblioteca própria do *Google Colaboratory* para integrar ao ambiente de desenvolvimento arquivos disponíveis em aplicativos *Google*.

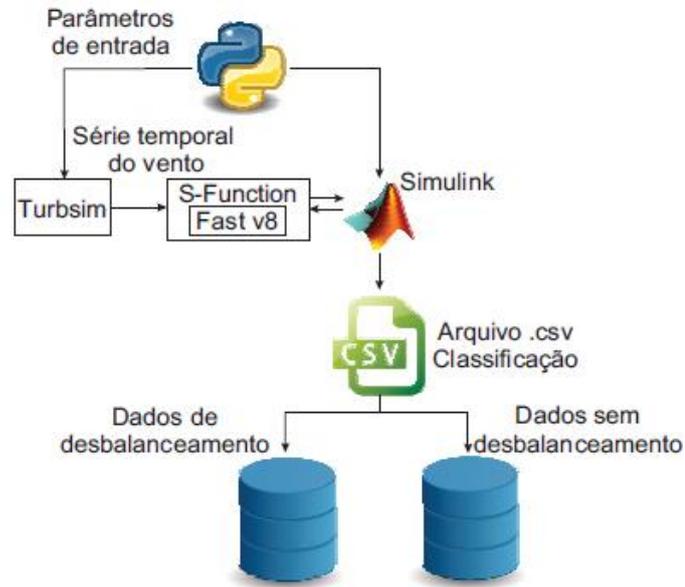
Os *notebooks* desenvolvidos para o trabalho estão armazenados no endereço eletrônico: <https://drive.google.com/drive/folders/1xOD4XQSpk3rZJZDDJ98mdtNPibrNdlkj?usp=sharing>

### 3.2 ORIGEM DO BANCO DE DADOS

O banco de dados de velocidade rotacional estimada do aerogerador no domínio da frequência utilizado no desenvolvimento do trabalho foi construído usando plataformas de simulação computacional. Essa decisão se dá por conta da indisponibilidade de dados experimentais de campo, além da maior conveniência que se consegue com plataformas de simulação computacional, as quais não necessitam sistemas complexos de instrumentação e podem simular diferentes cenários e operações que demandariam tempo em uma turbina eólica real. Ademais, os *softwares* de simulação utilizados são já consolidados por pesquisadores do tema de energia eólica e foram utilizados e validados na área de pesquisa por meio de trabalhos (MALIK, 2016; da ROSA, 2019; DELAZZERI, 2020; HÜBNER, 2021).

O fluxograma sequencial para o desenvolvimento do banco de dados é ilustrado conforme Figura 13.

Figura 13 - Etapas para geração dos dados



Fonte: (da ROSA; L. D., 2019).

O detalhamento da estrutura de simulação utilizada com Turbsim-FAST-Simulink está bem descrita nos trabalhos de da Rosa (2019) e Hübner (2021), não sendo o foco desse trabalho explicar as características e propriedades dos *softwares* de simulação computacional, bem como, as configurações realizadas nessas plataformas, pois o banco de dados aplicado no trabalho proposto foi gerado por Hübner (2021).

Para gerar os dados de velocidade rotacional estimado do aerogerador foram consideradas treze condições de velocidade média do vento na faixa de 12,5 a 24,5m/s espaçadas de 1 m/s e três condições de intensidade de turbulência (5% | 7,5% | 10%), formando combinações as quais são atreladas a modificação da densidade de massa do material em uma de suas pás, criando conjunto de dados com 105%, 104%, 103%, 102%, 101%, 99%, 98%, 97%, 96%, 95% de balanceamento de massa em uma das pás caracterizando o desequilíbrio de massa no rotor e um conjunto de dados na condição sem desbalanceamento de massa.

De acordo com Hübner (2021), os dados na forma de séries temporais utilizadas para o desenvolvimento do trabalho proposto possuem frequência de amostragem final de 50Hz no domínio do tempo, ou seja, 50 amostras geradas por segundo para cada simulação (combinação de condições) ou ainda uma amostra a cada 0,02 segundos com um tempo salvo de 120

segundos dos 180 segundos totais de simulação. Pois, foram removidos dados relativos aos 60 segundos de transitório inicial das amostras de velocidades de rotação do aerogerador.

Sob os dados aplicou-se também uma filtragem por faixa de velocidade rotacional para capturar apenas simulações/amostras pertencentes a região 3 de operação da turbina eólica, pois é nessa etapa apenas em que a velocidade do aerogerador em condições normais é mais constante. Uma variação de velocidade pode representar desbalanceamento de massa do rotor.

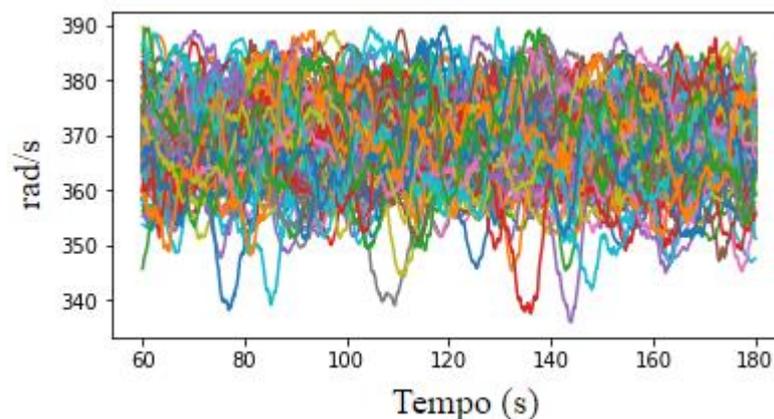
Conforme Hübner (2021), a razão pela qual é importante observar apenas os resultados pertencentes à região 3 de operação do aerogerador, é que desequilíbrios de massa do rotor resultam em distúrbios em  $1p$  (0,34Hz aproximadamente), e a velocidade do rotor é quase constante nesta região.

Por fim, o trabalho em questão, assim como o de Hübner (2021), utiliza-se de séries temporais no domínio da frequência para treinar os modelos e verificar seu desempenho. Dessa forma, o método *Power Spectral Density* (PSD) foi aplicado para modificar a perspectiva de análise da série do domínio do tempo para o domínio da frequência.

As amostras contêm 6000 características/atributos relativos a dados capturados no espaço temporal. Em contrapartida, no domínio da frequência a dimensão dos dados é reduzida para 751 características/atributos relativos à transformação realizada para o domínio da frequência. Esses valores foram observados ao manipular as variáveis no ambiente de desenvolvimento *Google Colab* utilizado para aplicação dos métodos.

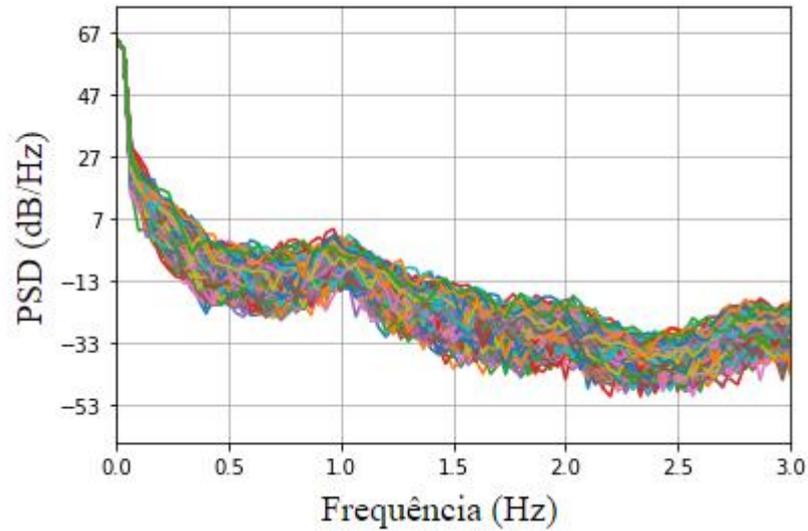
Para efeito de comparação, em Figura 14 e 15, são ilustradas 203 amostras/observações da classe 0% de desbalanceamento de massa no rotor (condição normal) no domínio do tempo e da frequência, respectivamente.

Figura 14 - Observações de condição sem falha gerada no domínio do tempo



Fonte: Autor.

Figura 15 - Representação da condição sem falha no domínio da frequência



Fonte: Autor.

### 3.3 MODELAGEM DO EXPERIMENTO COM 6 CLASSES DEFINIDAS

Nessa etapa, realizaram-se algumas considerações e tarefas preparatórias, sob as séries temporais segmentadas em 6 classes no domínio da frequência, visando a execução do experimento de treino e avaliação dos modelos de aprendizagem profunda (*deep learning*).

#### 3.3.1 Preparação

Definiu-se que os conjuntos de dados por meio das séries temporais caracterizadas por tipo/situação de funcionamento do aerogerador iriam ser englobados em classes de acordo com o valor absoluto de variação de massa em uma das pás. Assim, definiu-se as condições de contorno da tarefa de classificação a ser realizada conforme Tabela 2.

Tabela 2 - Definição de classes pelo agrupamento de cenários

<b>Tipo de Funcionamento</b>	<b>Classe no algoritmo</b>
Desbalanceamento 5% em uma das pás	5
Desbalanceamento 4% em uma das pás	4
Desbalanceamento 3% em uma das pás	3
Desbalanceamento 2% em uma das pás	2
Desbalanceamento 1% em uma das pás	1
Desbalanceamento 0% em uma das pás (Normal)	0
Desbalanceamento -1% em uma das pás	1
Desbalanceamento -2% em uma das pás	2
Desbalanceamento -3% em uma das pás	3
Desbalanceamento -4% em uma das pás	4
Desbalanceamento -5% em uma das pás	5

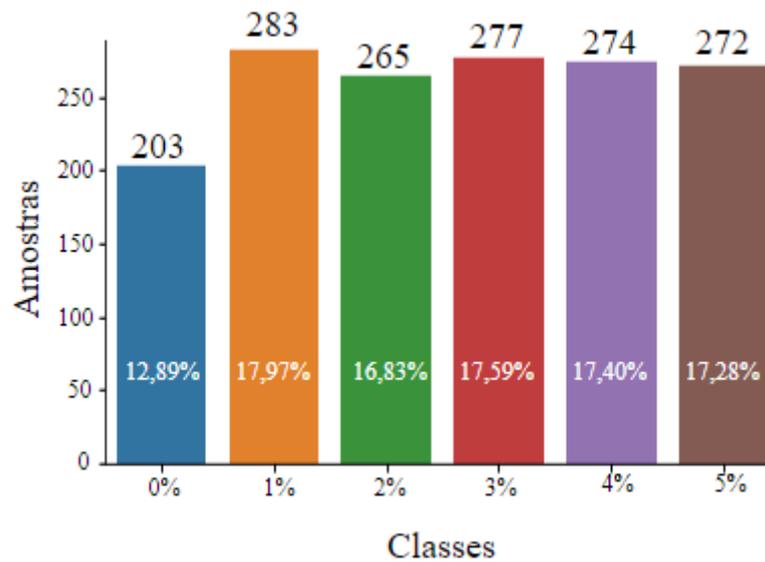
Fonte: Autor.

Em seguida, agrupou-se os dados de velocidade rotacional estimada do aerogerador contendo séries temporais univariadas relativo as 6 classes definidas. Então, criou-se um conjunto de dados composto por variáveis explicativas as quais são denominadas como variáveis de entrada com dimensão de 1574 x 751 no domínio da frequência, as quais são necessárias para os algoritmos realizarem os aprendizados.

A quantidade de amostras, bem como, a proporção de classes do conjunto total de 1574 amostras é ilustrada em Figura 16. Por consequência, criou-se também o vetor de variáveis dependentes conhecido como variáveis de saída, *target* ou alvo com tamanho de 1574. Essas variáveis representam a classificação das amostras/observações do conjunto de dados de entrada.

Pela distribuição de classes pode-se perceber que a classe que possui uma proporção menor de amostras é a 0% (comportamento normal) de falha de desbalanceamento de massa no rotor com aproximadamente 13% da totalidade de amostras.

Figura 16 - Distribuição de classe do conjunto total



Fonte: Autor

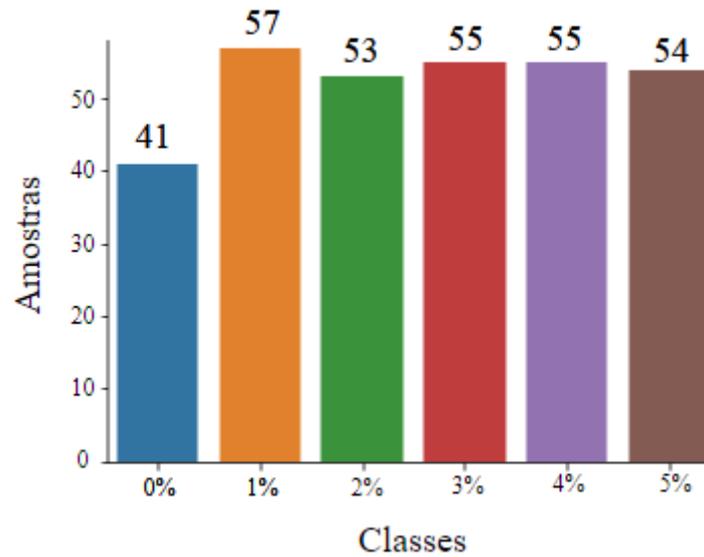
### 3.3.2 Pré-processamento

Nessa etapa, inicialmente, realizou-se a divisão dos dados. Em seguida, aplicou-se a técnica de balanceamento de classes nos dados de treino/validação. A ordem foi aplicada dessa forma, respeitando as práticas recomendadas pela biblioteca Scikit-learn, disponibilizadas em < [https://scikit-learn.org/stable/common\\_pitfalls.html](https://scikit-learn.org/stable/common_pitfalls.html)>, para testes experimentais de modo a evitar erros nos resultados devido a vazamento dos dados (*Data Leakage*).

A divisão de dados foi implementada particionando o conjunto em 80% de dados de treino/validação e 20% dados de teste. Além disso, definiu-se que a proporção de classes de cada partição seria estratificada, ou seja, dados de treino/validação e teste possuem a mesma proporção de classes do conjunto de dados total.

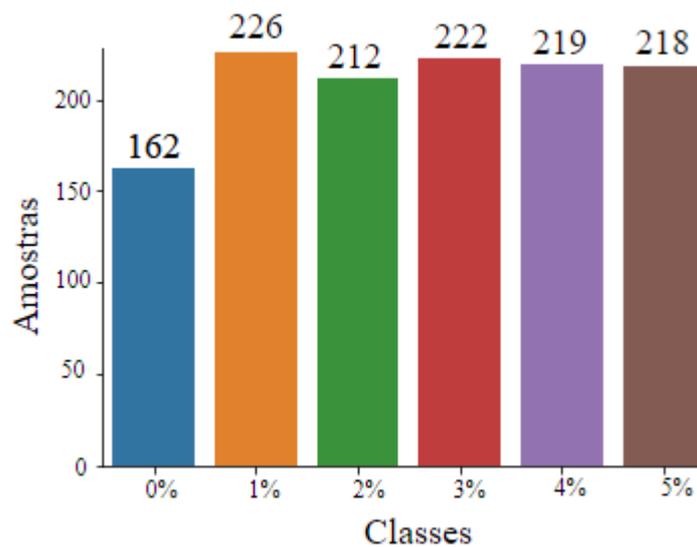
Para isso, utilizou-se a função *train\_test\_split* da biblioteca *Scikit-learn*. Assim, a quantidade de amostras por classe para a partição de dados de teste (315 amostras) e de treino/validação (1259 amostras) pode ser visualizada, respectivamente, nas Figura 17 e 18.

Figura 17 - Amostras por classe conjunto de teste



Fonte: Autor.

Figura 18 - Amostras por classe conjunto de treino/validação



Fonte: Autor.

Assim como o conjunto de dados total, pode-se notar pela natureza da divisão adotada que a classe minoritária é a 0% (sem falha) e das classes que representam níveis de falha, as classes 1% e 2% possuem mais amostras e menos amostras, respectivamente.

Após realizada a divisão dos dados, utilizou-se a função *to\_pickle* da biblioteca *Pandas* para gravar as séries temporais constituintes de cada partição no Google Drive para posteriormente serem utilizadas como referência.

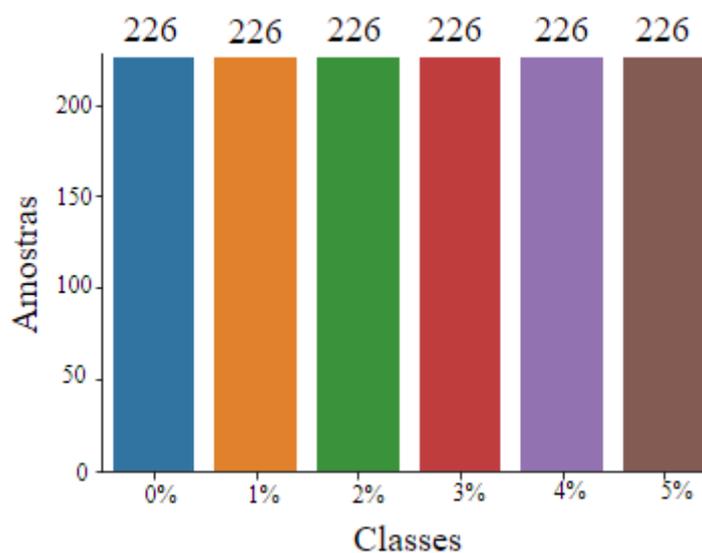
Com o intuito de verificar de forma comparativa o desempenho dos modelos de aprendizado profundo FCN e Resnet, treinados sob conjunto de dados com classes balanceadas, aplicou-se a técnica de balanceamento de classes SMOTE para obter uma quantidade de amostras igual para cada classe.

A estratégia de reamostragem parametrizada para o método foi de geração de amostras sintéticas para todas as classes minoritárias, de forma a igualar a quantidade de amostras em cada classe. Além disso, o parâmetro  $k$  definido para os  $k$  vizinhos próximos usados como referência para geração dos novos dados foi igual a 3. A definição desses parâmetros seguiu os parâmetros adotados em Aires (2021), na qual se obteve resultados satisfatórios de desempenho ao aplicar a técnica SMOTE sob as amostras.

Com a aplicação da técnica SMOTE para balancear as classes por meio de *Oversampling*, ou seja, aumento de amostras de classes minoritárias para igualar a quantidade da classe majoritária (classe 1% no experimento em questão), a quantidade de amostras de cada classe passou a ser 226 amostras. Assim, o conjunto de treino/validação passa a conter 1356 amostras, os quais representam 81% das amostras totais considerando o conjunto de teste.

A quantidade de amostras por classe do novo conjunto de treino/validação é ilustrada em Figura 19.

Figura 19 - Amostras por classe conjunto de treino/validação balanceado



Fonte: Autor.

### 3.4 MODELAGEM COM OUTROS EXPERIMENTOS

Além do experimento com conjunto de amostras distribuídas em 6 classes, em que se utilizou dados de treino/validação com e sem equilíbrio de classes, realizou-se outros sete experimentos por meio do mesmo processo realizado com o conjunto de dados de 6 classes, onde variou-se a quantidade de classes existente no conjunto de dados.

Da partição de dados de treino/validação e teste de 6 classes armazenada no Google Drive, excluiu-se as classes que não eram requeridas e, os conjuntos de dados para cada um dos experimentos foram concebidos. Na Tabela 3, pode-se visualizar a quantidade de amostras totais e das partições utilizadas em cada experimento, bem como, a quantidade quando aplicado balanceamento nos dados de treino/validação.

Tabela 3 - Amostras em experimentos com quantidade de classes menores que 6

Dados Nº classes	Treino/Validação		Teste	Total	
	Desbalanceado	Balanceado		Desbalanceado	Balanceado
4	828	904	207	1035	1111
3	602	666	150	752	816
2 (0% 1%)	388	452	98	486	550
2 (0% 2%)	374	424	94	468	518
2 (0% 3%)	384	424	96	480	520
2 (0% 4%)	381	438	96	477	534
2 (0% 5%)	380	436	95	475	531

Fonte: Autor.

Além disso, as classes que foram incluídas em cada experimento e a proporção dessas com relação a quantidade total de classes é apresentada em Tabela 4. Como pode-se perceber, as classes minoritárias e majoritárias em cada experimento seguem a mesma proporção de amostras por classe do experimento realizado com conjunto de 6 classes.

Tabela 4 - Classes em experimentos com quantidade de classes menores que 6

<b>Classes</b>	0%	1%	2%	3%	4%	5%
<b>N° classes</b>						
4	19,61%	27,34%		26,76%		26,28%
3	26,99%			36,83%		36,17%
2 (0% 1%)	41,76%	58,23%				
2 (0% 2%)	43,37%		56,62%			
2 (0% 3%)	42,29%			57,70%		
2 (0% 4%)	42,55%				57,44%	
2 (0% 5%)	42,73%					57,26%

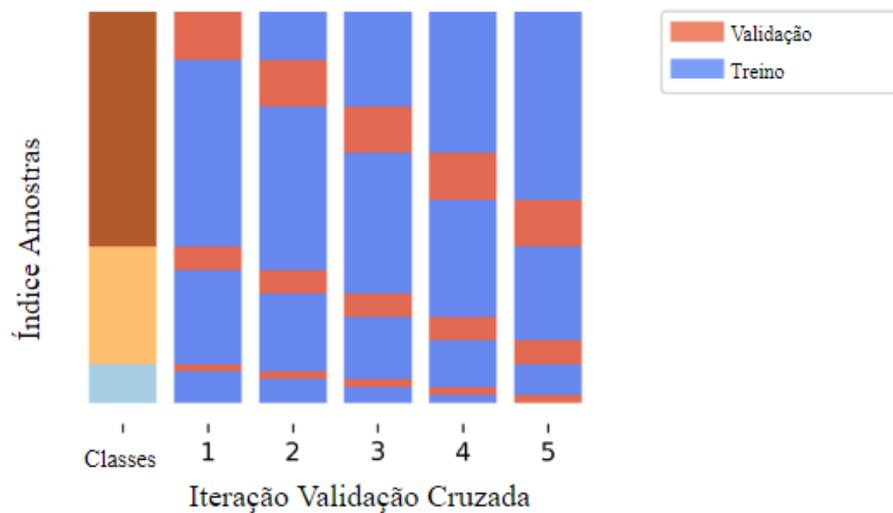
Fonte: Autor.

### 3.5 MÉTODO DE AVALIAÇÃO E AJUSTE DE MODELOS

Nessa etapa, aplicou-se o método de validação cruzada denominado *StratifiedKfold* em todos os experimentos. Métodos de validação são estratégias utilizadas para particionar amostras de dados e executar o treinamento do modelo almejando a máxima generalização do conjunto de amostras apresentado. Isso significa buscar o mínimo enviesamento do modelo no aprendizado dos dados.

O método aplicado no trabalho em questão, consistiu em gerar a partir do conjunto de dados de treino/validação, seja com ou sem equilíbrio de classes, K iterações ou combinações de amostras formadas por partições de dados de treino e validação. As partições de dados a cada K iteração da validação cruzada, são formadas por amostragem estratificada, ou seja, as partições de treino e validação manterão a proporção de amostras por classe do conjunto de dados geral, conforme é ilustrado exemplo da Figura 20.

Figura 20 - Funcionamento validação *StratifiedKfold*



Fonte: Autor. Adaptado de (MÜLLER, 2020). Disponível em: <https://amueller.github.io/aml/04-model-evaluation/1-data-splitting-strategies.html>. Acesso em: 14 jul. 2022.

O  $K = 5$  do exemplo em Figura 20 foi o mesmo definido na aplicação do método do trabalho em questão. Ou seja, 5 iterações formadas por partições divididas em  $1/5$  de amostras de validação e o restante em amostras de treino. A proporção das partições em todos os experimentos os quais a partição de treino/validação não possui equilíbrio de classes é aproximadamente 64% dados de treino, 16% dados de validação e 20% dados de teste. Em contrapartida, nos experimentos em que a partição de treino/validação é equilibrada, a proporção é aproximadamente 65% dados de treino, 16% dados de validação e 19% dados de teste devido a geração sintética de amostras das classes minoritárias.

A escolha do método de validação cruzada *StratifiedKfold* foi realizada devido a duas características positivas que o método oferece. A primeira é a repetição do processo de repartição em conjuntos de treino e validação o que reduz o efeito da aleatoriedade das amostras separadas em uma única vez, aumentando as possibilidades de análises estatísticas sob o desempenho dos algoritmos.

A segunda é a forma de repartição ocorrer por amostragem estratificada, o que garante que em ambos conjuntos a distribuição de amostras por classe seja a mesma do conjunto de dados original. Com isso, evita-se que o desbalanceamento entre as classes possa ser maior a cada repartição, podendo assim, enviesar os resultados de treinamento e teste.

Em seguida, realizou-se o treinamento dos modelos de aprendizado profundo FCN e Resnet, os quais foram ajustados sob os dados de treino inseridos. Parâmetros relativos a características do treinamento da rede neural são apresentados em Tabela 5.

Tabela 5 – Parâmetros das redes neurais treinamento

<b>Parâmetro</b>	<b>Valor</b>
Tamanho do lote	64
Taxa de aprendizagem	0,001
Épocas	20
Otimizador	Adam
Função de custo	entropia cruzada

Fonte: Autor.

O otimizador e a função de custo/perda foram definidos por padrão pela biblioteca que implementa os algoritmos. O número de épocas foi definido em função de análise do comportamento gráfico de minimização da função de perda e desempenho nos dados de validação por época os quais não obtiveram melhora de performance com valores maiores de épocas. Para o restante dos parâmetros manteve-se valores sugeridos pelos tutoriais da biblioteca Tsai, visto que variar as configurações de treinamento dos modelos por meio dos parâmetros não são o foco do estudo abordado por esse trabalho.

Dessa forma, os classificadores de aprendizado profundo FCN e Resnet, foram treinados em cada experimento, por 5 iterações ou conjunto de amostras diferentes as quais geraram 5 modelos ajustados que foram avaliados sob os dados de teste. A avaliação final da performance de cada algoritmo se deu pela média das métricas de avaliação relativo aos modelos de predição.

Para retratar o comportamento dos modelos na etapa de testes, utilizou-se como ferramenta de visualização a Matriz de Confusão. Ela permite uma medida efetiva do modelo de classificação, apresentando o número de classificações corretas versus as classificações preditas para cada classe, sobre um conjunto de exemplos. A diagonal principal apresenta para cada classe o número de classificação corretas (PROVOST; KOHAVI, 1998).

Utilizou-se a Acurácia, Precisão, *Recall* (Sensibilidade ou Revocação) e o F1-Score como métricas de avaliação de desempenho para fins de comparação de modelos e técnicas implementadas.

Além disso, para analisar o desempenho de modelos por meio das métricas de avaliação para problemas multiclasse, os dados são tratados como uma coleção de problemas binários, um para cada classe. Existem algumas maneiras de calcular a média de cálculos de métricas binárias no conjunto de classes, cada uma das quais pode ser útil em algum cenário. A forma utilizada no trabalho proposto é a “macro” que calcula a média das métricas binárias, dando peso igual a cada classe.

A acurácia é uma métrica de desempenho que avalia a performance geral de algoritmos de classificação, ela mensura a quantidade de classificações corretas sob todas as classificações realizadas. Essa métrica não analisa o desempenho da tarefa de classificação para uma classe em específico, e sim o desempenho de acertos para todas as classes existentes no experimento. Conforme Equação 13, em que  $V_{c1}$  é a quantidade de classificações corretas da classe 1 e  $F_{c1}$  é a quantidade de classificações erradas que foram apontadas como sendo da classe 1.

$$A = \frac{V_{c1} + \sum_2^n V_{cn}}{V_{c1} + F_{c1} + \sum_2^n V_{cn} + F_{cn}} \quad (13)$$

A precisão é uma métrica de desempenho que avalia o poder preditivo do algoritmo em apontar cada classe, ela mensura a quantidade de classificações corretas de uma determinada classe sob todas as predições feitas sob essa classe, conforme Equação 14.

$$P = \frac{V_{c1}}{V_{c1} + F_{c1}} \quad (14)$$

Recall ou Revocação é uma métrica de desempenho que avalia o poder de classificação do algoritmo em identificar todas as amostras de uma determinada classe, ela mensura a quantidade de classificações corretas de uma determinada classe sob todas as amostras existentes no conjunto dessa classe, conforme Equação 15.

$$R = \frac{V_{c1}}{V_{c1} + (\sum_2^n F_{cn})} \quad (15)$$

Como as métricas Precisão e Recall são resultados de avaliação que retratam o desempenho da classificação para cada classe, para ter-se o desempenho geral do algoritmo por meio delas, calcula-se a média dos resultados para cada classe existentes no experimento.

Dessa forma, pode-se compreender que essas duas métricas fornecem análises complementares sobre experimentos com a tarefa de classificação. Enquanto a Precisão fornece a capacidade preditiva em apontar determinada classe e acertar, não confundindo amostras de outras classes como sendo dessa. A Recall, avalia a capacidade do algoritmo em acertar todas as amostras que realmente são de uma determinada classe.

Por fim, F1-Score é uma métrica de avaliação que simplesmente combina as métricas de Precisão e Recall em uma só. É calculada pela média harmônica entre as duas, conforme Equação 16. Por se tratar de média harmônica em comparação com a média aritmética por exemplo, visa penalizar o desempenho final do algoritmo, trazendo a média para valores baixos em caso de Precisão ou Recall serem baixos.

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \quad (16)$$

## 4. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

Neste capítulo destacam-se os principais resultados obtidos durante o desenvolvimento desta pesquisa. Assim, é organizada por conjuntos de dados de treino/validação os quais diferenciam-se pela quantidade de classes existentes e foram tomados como entrada para o treinamento dos algoritmos de aprendizado profundo baseado em redes neurais FCN e Resnet.

Além de variar a quantidade de classes existente nas amostras da base de dados, outra combinação que foi experimentada foi o balanceamento de amostras entre as classes do conjunto de dados através da aplicação da técnica de SMOTE. Salienta-se que o SMOTE foi aplicado no conjunto de dados de treino/validação com o intuito de equilibrar as classes nos experimentos.

Com isso, buscou-se compreender o impacto da performance média dos algoritmos de aprendizado profundo de redes neurais na tarefa de classificação com conjuntos de dados de variadas quantidades de classes e, em cada uma aplicando o equilíbrio de amostras por classe.

Ressalta-se que devido à natureza da metodologia adotada com a geração de cinco modelos mediante cinco combinações de repartição de dados de treino e validação (iterações), um volume significativo de resultados sobre o experimento foi gerado. Com isso, dedicou-se os apêndices deste trabalho para melhor alocar o comportamento no treinamento ilustrando a função de perda/custo dos modelos treinados, bem como, a performance destes na etapa de testes mediante apresentação da matriz de confusão.

### 4.1 CONJUNTO DE DADOS COM SEIS CLASSES

Experimentos realizados com conjunto de dados de seis classes, utilizam-se a totalidade de classes rotuladas existentes para a problemática adotada, conforme apresentado no Capítulo 3, dessa forma, são o foco de análise do presente trabalho. Assim, a apresentação e análise de resultados dos algoritmos de aprendizado profundo de redes neurais contemplam para esses uma maior riqueza em detalhes em comparação com experimentos com quantidades inferiores de classes.

### 4.1.1 Conjunto sem equilíbrio de classes

O conjunto de dados de treino/validação desbalanceado empregado possui 1259 amostras, conforme apresentado na Figura 18. Desse conjunto, foram geradas cinco combinações de repartições de dados de treino e validação com distribuições média de classes, conforme Figura 16. A proporção de amostras do experimento foi de: 64% de dados de treino, 16% de dados de validação e 20% de dados de teste.

#### 4.1.1.1 Fully Convolutional Network

O comportamento do modelo de aprendizado profundo (*deep learning*) FCN durante o ajuste do algoritmo em 20 épocas tendo como entrada os dados de treino via técnica de minimização de custo/perda *Backpropagation* é apresentado no Apêndice A.

Pode-se notar que ao longo do treinamento para cada modelo gerado em cada repartição a capacidade de minimização de perda do modelo é de aproximadamente 0,1, com perda inicial de aproximadamente 1,8 para uma perda final de aproximadamente 1,7.

O desempenho médio dos modelos por classe na etapa de validação, com algoritmo FCN o qual foi gerado pelas cinco combinações de repartição de dados de treino e validação, é exposto em Tabela 6.

Tabela 6 – Validação por classe FCN treinado sob 6 classes desbalanceadas

<b>Métricas</b>			
<b>Classes</b>	Precisão	Recall	F1-Score
0%	0% ± 0%	0% ± 0%	0% ± 0%
1%	35,6% ± 4,16%	79,8% ± 13,7%	48,8% ± 5,5%
2%	36,6% ± 9,45%	16,2% ± 11,84%	20,2% ± 10,35%
3%	38,6% ± 7,89%	12,6% ± 6,58%	18,2% ± 7,5%
4%	19,6% ± 11,46%	25,6% ± 21,59%	21% ± 13,96%
5%	39,4% ± 8,20%	59% ± 25,54%	43,6% ± 8,32%

Fonte: Autor.

Após, realizou-se a etapa de teste dos modelos treinados, na qual utilizou-se como entrada os dados de teste (315 amostras), apresentados no Capítulo 3. Para ilustrar o desempenho dos modelos FCN, utilizou-se a Matriz de Confusão representando o comportamento preditivo dos modelos ao classificar cada amostra do conjunto de dados de teste, conforme apresentado no Apêndice B.

Com isso, ao analisar os resultados ilustrados pelas Matrizes de Confusão de teste, pode-se perceber que ocorre um padrão de resultados para as classes 0% e 1% em todos os testes realizados.

A capacidade de predição do algoritmo FCN para a classe 0% é a pior entre classes, não conseguindo classificar nenhum dado dessa classe corretamente e indicando ao invés, amostras dessa classe como sendo da classe 1% majoritariamente e 2 e 3% minoritariamente em média.

Entretanto, a capacidade de predição do algoritmo FCN para a classe 1% é satisfatório em termos de acertos de amostras que realmente são da classe 1%, porém ele confunde amostras que são de outras classes como sendo a classe 1%, fazendo com que a precisão do algoritmo para essa classe não seja satisfatória.

Visando sintetizar por meio de métricas de desempenho a performance média do algoritmo FCN por classe, em Tabela 7 é exposto o desempenho médio do modelo ao predizer as classes do conjunto de dados de teste.

Pode-se verificar que o algoritmo FCN consegue, dentre as classes existentes, obter uma melhor performance na classificação das classes 1% e 5%. Em contrapartida, destaca-se que o modelo não conseguiu aprender amostras da classe 0% pois não obteve nenhum acerto na classificação posterior na etapa de teste, zerando todas as métricas de desempenho.

Tabela 7 - Teste por classe FCN treinado sob 6 classes desbalanceadas

<b>Métricas</b>			
<b>Classes</b>	Precisão	Recall	F1-Score
0%	0% ± 0%	0% ± 0%	0% ± 0%
1%	34,0% ± 1,87%	80,4% ± 13,24%	47,4% ± 3,65%
2%	26,8% ± 11,45%	12,6% ± 7,23%	15,6% ± 6,31%
3%	25,2% ± 10,26%	9,2% ± 5,12%	13% ± 6,36%
4%	36,6% ± 9,45%	27,6% ± 18,04%	26,4% ± 13,96%
5%	34,6% ± 4,56%	55,6% ± 26,6%	40,8% ± 12,52%

Fonte: Autor.

Por fim, a performance média final e desvio padrão das métricas dos modelos, de aprendizado profundo (*Deep learning*) FCN, treinados por 5 repartições de dados de treino e validação combinadas de maneira diferente, é exposto conforme Tabela 8.

Dessa forma, pode-se notar que o desempenho médio do modelo em questão é insatisfatório pois fica abaixo de 50% de acurácia o que significa que o modelo não conseguiu em média, sob os dados do trabalho proposto, acertar ao menos metade das amostras a serem classificadas.

Tabela 8 - Teste geral FCN treinado sob 6 classes desbalanceadas

<b>Etapa</b>	Validação	Teste
<b>Métricas</b>		
Acurácia	33,8% ± 3,42%	32,6% ± 2,41%
Precisão	28,2% ± 4,15%	26,2% ± 3,35%
Recall	32% ± 3,46%	30,8% ± 2,59%
F1-Score	25,4% ± 2,97%	23,6% ± 2,97%

Fonte: Autor.

#### 4.1.1.2 Residual Network

O comportamento do modelo de aprendizado profundo (*deep learning*) Resnet durante o ajuste do algoritmo em 20 épocas tendo como entrada os dados de treino via técnica de minimização de custo/perda *Backpropagation* é apresentado no Apêndice A.

Pode-se notar que ao longo do treinamento para cada modelo gerado em cada repartição a capacidade de minimização de custo do modelo é de aproximadamente 1,2, com perda inicial de aproximadamente 1,8 para uma perda final de aproximadamente 0,6. Desempenho esse que é melhor se comparado ao algoritmo FCN.

Com relação ao comportamento da função de perda/custo na etapa de treino ilustrados no Apêndice A e C, pode-se notar que existe uma oscilação maior na função de custo/perda para os dados de validação em comparação com dados de treino. Isso ocorre pois a cada época de treinamento os algoritmos de aprendizado profundo visualizam cada amostra de treino uma vez e são ajustados com base neles. Ao passo que, os dados de validação são apenas tomados como entrada do modelo após serem ajustados pelos dados de treino a cada época para serem

preditos, e assim acompanhar o desempenho de aprendizado sob amostras diferentes das ajustadas pelos modelos.

Dessa forma, é natural que em determinadas épocas, o modelo possa sobreajustar seus parâmetros com base nos dados de treino. Contudo, o interessante é observar que ao longo do treinamento, ao ver cada vez mais dados de treino, ele consegue generalizar os parâmetros do modelo e minimizar o custo/perda de dados de validação de forma que no final, a função de perda esteja aproximadamente em mesmos níveis para dados de treino e validação.

O desempenho médio dos modelos por classe na etapa de validação, com algoritmo Resnet o qual foi gerado pelas cinco combinações de repartição de dados de treino e validação, é exposto em Tabela 9. Pode-se perceber um desempenho significativamente superior em comparação ao algoritmo FCN apresentado, com as mesmas combinações de dados de treino e validação utilizadas no treinamento.

Tabela 9 - Validação por classe Resnet treinado sob 6 classes desbalanceadas

<b>Métricas</b>			
<b>Classes</b>	Precisão	Recall	F1-Score
0%	84,6% ± 4,83%	73,6% ± 7,37%	78,4% ± 4,88%
1%	77,8% ± 4,92%	83,2% ± 5,26%	80,2% ± 3,49%
2%	85,4% ± 5,94%	86,4% ± 7,44%	85,8% ± 6,3%
3%	83,2% ± 6,02%	84% ± 5,52%	83,4% ± 5,77%
4%	82% ± 7,21%	75,8% ± 7,12%	78,6% ± 4,67%
5%	86% ± 6,12%	91,2% ± 8,20%	88% ± 3,74%

Fonte: Autor.

Após, realizou-se a etapa de teste dos modelos treinados, na qual utilizou-se como entrada os dados de teste desbalanceados (315 amostras) apresentados no Capítulo 3. Para ilustrar o desempenho dos modelos Resnet, utilizou-se a Matriz de Confusão representando o comportamento preditivo dos modelos ao classificar cada amostra do conjunto de dados de teste, conforme apresentado no Apêndice B.

Com isso, ao analisar os resultados ilustrados pelas Matrizes de Confusão para o teste, pode-se perceber que ocorre uma tendência na relação de predições entre as classes próximas.

Pode-se notar que as maiores dificuldades do modelo foram em diferenciar as classes 2% e 3% entre si, bem como, as classes 4% e 5%, ocorrendo predição de uma classe enquanto a correta seria a outra e vice-versa. Isso ocorre também entre as classes 0% e 1%.

Entretanto, as classes 1% e 2%, bem como, 3% e 4% obtiveram melhor diferenciação por parte do algoritmo e assim, as trocas de classificações entre essas classes não foram evidenciadas.

Visando sintetizar por meio de métricas de desempenho a performance média do algoritmo Resnet por classe, em Tabela 10 expõe-se o desempenho médio do modelo ao prever as classes do conjunto de dados de teste.

Pode-se notar que o algoritmo Resnet consegue obter um desempenho significativamente melhor em classificar as classes do conjunto de dados de teste em comparação ao algoritmo FCN. Além disso, destaca-se a homogeneidade dos resultados de performance para cada classe, tanto com relação a média das métricas como a dispersão da média evidenciada pela baixa variação dos desvios padrão se comparada ao algoritmo FCN.

Tabela 10 - Teste por classe Resnet treinado sob 6 classes desbalanceadas

Métricas	Precisão	Recall	F1-Score
Classes			
0%	87,8% ± 1,1%	73% ± 3,54%	79,8% ± 2,17%
1%	76% ± 2,45%	83% ± 1,41%	79,4% ± 1,82%
2%	75,8% ± 1,92%	79% ± 2,45%	77,6% ± 2,3%
3%	78,6% ± 1,52%	82,6% ± 2,41%	80,6% ± 1,67%
4%	82,4% ± 1,82%	71,4% ± 7,02%	76,2% ± 4,32%
5%	79,4% ± 5,5%	83,8% ± 3,35%	81,4% ± 2,19%

Fonte: Autor.

Por fim, a performance média geral e desvio padrão das métricas dos modelos, de aprendizado profundo (*deep learning*) Resnet, treinados por 5 repartições de dados de treino e validação combinadas de maneira diferente, é exposta conforme Tabela 11.

Assim sendo, pode-se verificar que o desempenho médio do modelo em questão de aproximadamente 80%. Além disso, observa-se que o desvio padrão do desempenho médio dos modelos Resnet são menores que os modelos FCN, principalmente na performance na etapa de teste.

Tabela 11 - Teste geral Resnet treinado sob 6 classes desbalanceadas

<b>Etapa</b>	<b>Validação</b>	<b>Teste</b>
<b>Métricas</b>		
Acurácia	82,6% ± 3,58%	79,2% ± 1,10%
Precisão	82,8% ± 3,27%	80,0% ± 1,00%
Recall	82,2% ± 3,27%	78,8% ± 0,84%
F1-Score	82,4% ± 3,21%	79,2% ± 1,10%

Fonte: Autor.

## 4.1.2 Conjunto com equilíbrio de classes

O conjunto de dados de treino/validação balanceado empregado possui 1356 amostras, conforme apresentado em Figura 19.

### 4.1.2.1 Fully Convolutional Network

O comportamento do modelo de aprendizado profundo (*deep learning*) FCN durante o ajuste do algoritmo em 20 épocas tendo como entrada os dados de treino via técnica de minimização de custo/perda *Backpropagation* é apresentado no Apêndice C.

Pode-se notar que ao longo do treinamento para cada modelo gerado, em cada repartição, a capacidade de minimização da função de custo/perda do modelo é de aproximadamente 0,15. Perda inicial de aproximadamente 1,8 para uma perda final de aproximadamente 1,65. Dessa forma, verifica-se que o treinamento do algoritmo FCN com dados balanceados aumentou a minimização da função de custo/perda em comparação com dados desbalanceados.

O desempenho médio dos modelos por classe na etapa de validação, com algoritmo FCN o qual foi gerado pelas cinco combinações de repartição de dados de treino e validação, é exposto em Tabela 12. Assim, nota-se uma melhora significativa do algoritmo FCN na predição da classe 0% em comparação com o mesmo algoritmo treinado com dados desbalanceados.

Tabela 12 - Validação por classe FCN treinado sob 6 classes balanceadas

<b>Métricas</b>	Precisão	Recall	F1-Score
<b>Classes</b>			
0%	47% ± 5,24%	69% ± 9,54%	55,8% ± 6,18%
1%	31,8% ± 10,21%	36,8% ± 11,5%	33,4% ± 8,62%
2%	35,8% ± 14,20%	18,2% ± 9,47%	23,2% ± 9,52%
3%	28,8% ± 44,13%	4,4% ± 7,8%	6,8% ± 10,89%
4%	29,8% ± 3,96%	38,6% ± 21,11%	31,8% ± 9,09%
5%	32,6% ± 2,61%	47,6% ± 29,13%	35,8% ± 13,52%

Fonte: Autor.

Após, realizou-se a etapa de teste dos modelos treinados, na qual utilizou-se como entrada os dados de teste (315 amostras), apresentados no Capítulo 3. Para ilustrar o desempenho dos modelos FCN, utilizou-se a matriz de confusão representando o comportamento preditivo dos modelos ao classificar cada amostra do conjunto de dados de teste, conforme apresentado no Apêndice D.

Com isso, ao analisar os resultados ilustrados pelas Matrizes de Confusão no teste para os modelos gerados, pode-se perceber que ocorrem desempenhos recorrentes (padrões de performance) para determinadas classes. A capacidade de predição do algoritmo FCN para a classe 0% melhora significativamente se comparado aos modelos gerados com dados desbalanceados, entretanto existem divergências entre classe 0% e 1% recorrentes de predição o que é demarcado pelas cores mais claras formando um quadrado na matriz de confusão na região de 0% e 1%.

Erros de predição entre si ocorrem também com as classes 4% e 5%, revelando a baixa capacidade de diferenciação de amostras dessas duas classes. Além disso, pode-se perceber a insatisfatória capacidade de predição do algoritmo FCN para a classe 3% por meio dos testes realizados.

Visando sintetizar por meio de métricas de desempenho a performance média do algoritmo FCN por classe, apresenta-se em Tabela 13, o desempenho médio do modelo ao prever as classes do conjunto de dados de teste. Assim como evidenciado nos resultados da etapa de validação e nas matrizes de confusão de teste, a classe 0% não só teve uma melhora significativa, entre as classes preditas, como também foi a classe com melhor desempenho. A classe 3% obteve resultados bastante insatisfatórios de desempenho médio.

Tabela 13 - Teste por classe FCN treinado sob 6 classes balanceadas

<b>Métricas</b>	Precisão	Recall	F1-Score
<b>Classes</b>			
0%	44,4% ± 4,83%	63,2% ± 5,22%	52% ± 4,64%
1%	39,2% ± 4,92%	47,6% ± 10,06%	42,8% ± 6,14%
2%	25,4% ± 2,51%	14% ± 7,07%	17,2% ± 4,66%
3%	30,4% ± 39,66%	2,6% ± 2,61%	4% ± 3,67%
4%	30,4% ± 3,29%	43,6% ± 15,44%	35,2% ± 4,32%
5%	37,4% ± 5,77%	45,2% ± 26,85%	36,4% ± 11,97%

Fonte: Autor.

Por fim, a performance média geral e desvio padrão das métricas dos modelos, de aprendizado profundo (*deep learning*) FCN, treinados por 5 repartições de dados de treino e validação balanceadas combinadas, é apresentada conforme Tabela 14.

Dessa forma, verifica-se que o desempenho médio geral do algoritmo FCN, assim como quando foi treinado por dados desbalanceados, é insatisfatório pois a performance de classificação através das métricas permanecem abaixo de 50%. Entretanto, pode-se afirmar que em todas as etapas avaliadas o desempenho médio geral do algoritmo FCN treinado a partir de dados balanceados obteve melhores resultados do que treinado mediante dados desbalanceados.

Tabela 14 - Teste geral FCN treinado sob 6 classes balanceadas

<b>Etapa</b>	Validação	Teste
<b>Métricas</b>		
Acurácia	35,6% ± 3,91%	34,8% ± 3,49%
Precisão	34,2% ± 8,58%	34,6% ± 6,73%
Recall	35,8% ± 3,56%	36% ± 3,08%
F1-Score	31,2% ± 4,32%	31% ± 1,58%

Fonte: Autor

#### 4.1.2.2 Residual Network

O comportamento do modelo de aprendizado profundo (*deep learning*) Resnet durante o ajuste do algoritmo em 20 épocas tendo como entrada os dados de treino via técnica de minimização de custo/perda *Backpropagation* é apresentado no Apêndice C.

Pode-se notar que ao longo do treinamento para cada modelo gerado em cada repartição a capacidade de minimização da função de custo/perda do algoritmo é semelhante ao comportamento do mesmo para dados desbalanceados, conforme exposto no Apêndice A. Dessa forma, pode-se verificar que o algoritmo Resnet possui maior capacidade de minimização da função de custo/perda do que o algoritmo FCN tanto para dados desbalanceados como balanceados.

O desempenho médio dos modelos por classe na etapa de validação, do algoritmo Resnet gerado por cinco combinações de repartição de dados de treino e validação é apresentado conforme Tabela 15. Além do desempenho do algoritmo nessa etapa ser superior ao algoritmo FCN, o mesmo apresenta melhora em comparação ao próprio algoritmo aplicado a dados desbalanceados, apresentado em Tabela 9.

Tabela 15 - Validação por classe Resnet treinado sob 6 classes balanceadas

Métricas	Precisão	Recall	F1-Score
<b>Classes</b>			
0%	84,6% ± 2,97%	84,8% ± 8,04%	84,6% ± 4,34%
1%	78,8% ± 6,53%	77,6% ± 1,67%	78% ± 3,67%
2%	86,6% ± 3,91%	89,2% ± 6,14%	87,8% ± 3,35%
3%	85,6% ± 3,44%	85,8% ± 2,77%	85,8% ± 2,17%
4%	84,4% ± 4,51%	79,2% ± 6,06%	81,8% ± 4,92%
5%	88,6% ± 2,88%	92% ± 5,43%	90,2% ± 3,7%

Fonte: Autor.

Após, realizou-se a etapa de teste dos modelos treinados, na qual utilizou-se como entrada os dados de teste (315 amostras) apresentados no Capítulo 3. Para ilustrar o desempenho dos modelos Resnet, utilizou-se a matriz de confusão representando o comportamento preditivo dos modelos ao classificar cada amostra do conjunto de dados de teste, conforme apresentado no Apêndice D.

Com isso, ao analisar as matrizes de confusão no teste, pode-se perceber que ocorre uma recorrência no padrão de predições entre classes próximas, de forma bem similar aos padrões descritos em 4.1.1.2 em que se analisou resultados de teste para os modelos treinados com dados desbalanceados.

Visando sintetizar por meio de métricas de desempenho a performance média do algoritmo Resnet por classe, na Tabela 16 expõe-se o desempenho médio do modelo ao predizer as classes do conjunto de dados de teste.

Dessa forma, verifica-se que o algoritmo Resnet não somente obtêm um desempenho significativamente melhor em predizer classes em comparação ao algoritmo FCN, como também seu desempenho é levemente superior ao próprio algoritmo treinado com dados desbalanceados. Além disso, destaca-se a homogeneidade dos resultados de performance para cada classe, tanto relacionado a média das métricas como também a dispersão da média (desvio padrão) se comparado aos resultados obtidos pelo algoritmo FCN.

Tabela 16 - Teste por classe Resnet treinado sob 6 classes balanceadas

Métricas	Precisão	Recall	F1-Score
Classes			
0%	84,4% ± 1,67%	79,2% ± 5,02%	81,4% ± 3,36%
1%	78% ± 1,73%	80,6% ± 0,89%	79,2% ± 1,1%
2%	77% ± 3,16%	78,6% ± 1,67%	77,8% ± 1,64%
3%	79,8% ± 3,9%	83,2% ± 4,55%	81,6% ± 4,16%
4%	82,6% ± 3,36%	76,4% ± 7,09%	79,4% ± 4,98%
5%	82,6% ± 3,91%	83,8% ± 2,28%	83,2% ± 2,59%

Fonte: Autor.

Por fim, a performance média geral e desvio padrão das métricas dos modelos, de aprendizado profundo (*deep learning*) Resnet, treinados por 5 repartições de dados de treino e validação balanceados combinadas de maneira diferente, é apresentado em Tabela 17.

Assim sendo, compreende-se que o desempenho médio final do algoritmo em questão, não somente obtêm um desempenho significativamente melhor em comparação ao algoritmo FCN, para mesmos dados treinados, mas também seu desempenho médio final supera 80% de F1-Score, o que é levemente superior ao próprio algoritmo treinado com dados desbalanceados, ilustrado em Tabela 11.

Tabela 17 - Teste geral Resnet treinado sob 6 classes balanceadas

<b>Etapa</b>	<b>Validação</b>	<b>Teste</b>
<b>Métricas</b>		
Acurácia	84,4% ± 2,61%	80,4% ± 2,30%
Precisão	84,8% ± 2,49%	80,6% ± 1,95%
Recall	84,4% ± 2,61%	80,4% ± 1,82%
F1-Score	84,4% ± 2,61%	80,6% ± 1,95%

Fonte: Autor.

## 4.2 CONJUNTOS COM QUANTIDADE DE CLASSES INFERIOR A SEIS

Com intuito de verificar o comportamento dos algoritmos para cenários contendo amostras com quantidades inferiores de classes, realizou-se experimentos com conjunto de dados possuindo 4 classes (0% | 1% | 3% | 5%), 3 classes (0% | 3% | 5%) e combinações de 2 classes com uma das classes sendo sempre a 0% e variando a classe que representa o nível de falha. Para esses experimentos foram aplicados os mesmos procedimentos adotados com experimentos de conjunto de dados contendo 6 classes, contudo, a exposição e análise dos resultados por classe se resumirão ao desempenho médio dos modelos durante a etapa de teste.

### 4.2.1 Conjuntos sem equilíbrio de classes

O resultado médio geral expresso pelas métricas de desempenho Acurácia, Precisão, Recall e F1-Score, das etapas de validação e teste, dos modelos de algoritmos treinados sob 5 repartições combinadas de dados de treino e validação com desequilíbrio de classe, é apresentada em Tabela 18 para o algoritmo FCN e em Tabela 19 para o Resnet.

As quantidades de amostras utilizadas em cada experimento, bem como, as repartições de treino/validação e teste aplicada foi descrita no Capítulo 3. A proporção de amostras para todos os experimentos é de aproximadamente 64% de dados de treino, 16% de dados de validação e 20% de dados de teste.

Com o intuito de compreender a capacidade de predição dos algoritmos FCN e Resnet para as classes observadas, essa seção estará segmentada por resultados médios das métricas de desempenho por classe na etapa de teste para cada experimento contendo quantidades distintas de classes.

Tabela 18 - FCN sob conjuntos desbalanceados com quantidade de classes inferior a 6

N° Classes	Validação				Teste			
	Acurácia	Precisão	Recall	F1 - Score	Acurácia	Precisão	Recall	F1 - Score
4 (0% 1% 3% 5%)	51,4% ± 3,21%	39,2% ± 3,77%	47,8% ± 2,86%	40% ± 2,92%	48,4% ± 3,21%	37% ± 3,94%	45,2% ± 2,77%	37,4% ± 3,21%
3 (0% 3% 5%)	53,4% ± 7,54%	61% ± 14,39%	50,8% ± 9,5%	48,4% ± 12,24%	53,4% ± 8,88%	68,2% ± 3,96%	50,4% ± 10,53%	48% ± 12,83%
2 (0% 1%)	58,2% ± 0,45%	29% ± 0%	50% ± 0%	37% ± 0%	58% ± 0%	29% ± 0%	50% ± 0%	37% ± 0%
2 (0% 2%)	52,6% ± 4,04%	37,4% ± 8,88%	47,4% ± 3,21%	39,4% ± 3,85%	56,2% ± 2,49%	44,6% ± 15,87%	51,4% ± 1,95%	43,6% ± 6,99%
2 (0% 3%)	58,2% ± 1,3%	49,2% ± 27,66%	50,8% ± 1,1%	37,8% ± 2,05%	57,6% ± 0,89%	49% ± 27,39%	50,6% ± 0,89%	37,8% ± 2,68%
2 (0% 4%)	68,4% ± 7,83%	71,6% ± 24%	63% ± 9,27%	58,8% ± 14,65%	65,6% ± 5,55%	80,6% ± 3,13%	60% ± 6,04%	54,4% ± 10,14%
2 (0% 5%)	79,8% ± 10,78%	87% ± 6,36%	76% ± 12,65%	75,8% ± 14,34%	76,2% ± 11,76%	80,4% ± 8,88%	73% ± 13,66%	71,4% ± 16,59%

Fonte: Autor.

Tabela 19 - Resnet sob conjuntos desbalanceados com quantidade de classes inferior a 6

N° Classes	Validação				Teste			
	Acurácia	Precisão	Recall	F1 - Score	Acurácia	Precisão	Recall	F1 - Score
4 (0% 1% 3% 5%)	91% ± 1,22%	90,6% ± 1,67%	90% ± 1,73%	90,4% ± 1,67%	90,4% ± 1,14%	90,8% ± 0,84%	89,4% ± 1,14%	90% ± 1%
3 (0% 3% 5%)	97,8% ± 1,3%	98% ± 1,22%	97,8% ± 1,48%	98% ± 1,22%	96,8% ± 0,45%	97,6% ± 0,55%	97,4% ± 0,89%	97,4% ± 0,89%
2 (0% 1%)	82% ± 3,67%	83,2% ± 4,87%	80,4% ± 3,78%	81% ± 4,06%	82,6% ± 3,97%	83,8% ± 2,77%	80,2% ± 4,44%	81,2% ± 4,44%
2 (0% 2%)	97,6% ± 2,79%	97,6% ± 2,3%	97,4% ± 2,7%	97,6% ± 2,79%	96% ± 0%	96% ± 0%	95,4% ± 0,55%	96% ± 0%
2 (0% 3%)	99,2% ± 1,3%	99,2% ± 1,3%	99% ± 1,41%	99,2% ± 1,3%	99,6% ± 0,55%	99,6% ± 0,55%	99,6% ± 0,55%	99,6% ± 0,55%
2 (0% 4%)	99,8% ± 0,45%	99,8% ± 0,45%	99,6% ± 0,89%	99,8% ± 0,45%				
2 (0% 5%)	99,8% ± 0,45%	99,6% ± 0,89%	99,8% ± 0,45%	99,8% ± 0,45%	98,6% ± 0,55%	98,6% ± 0,55%	98,6% ± 0,55%	98,6% ± 0,55%

Fonte: Autor.

#### 4.2.1.1 Conjunto com quatro classes

Em Tabela 20 verifica-se que o algoritmo FCN possui desempenho significativamente inferior ao algoritmo Resnet para todas as classes previstas no experimento com 4 classes. Sendo que para a classe 0%, que representa funcionamento do aerogerador normal sem falha, o algoritmo FCN para nenhum dos cinco modelos gerados acerta uma previsão apontada como 0% corretamente. Em contrapartida, o algoritmo Resnet possui uma homogeneidade no desempenho médio das classes previstas, sendo que para as classes 3% e 5% atinge um desempenho médio de mais de 95% de F1-Score.

Tabela 20 - Desempenho por classe sob conjunto desbalanceado com 4 classes

Métricas	Precisão		Recall		F1-Score	
	FCN	Resnet	FCN	Resnet	FCN	Resnet
0%	0% ±	87,2% ±	0% ±	74,6% ±	0% ±	80,4% ±
	0%	1,1%	0%	3,21%	0%	1,67%
1%	43% ±	82,8% ±	82% ±	91,4% ±	56,6% ±	87% ±
	1,58%	1,79%	4,53%	0,89%	1,52%	1%
3%	47,2% ±	93,4% ±	19,2% ±	98% ±	26,6% ±	95,8% ±
	14,48%	1,67%	7,5%	1,41%	8,82%	1,1%
5%	57,2% ±	98,8% ±	79,6% ±	93,6% ±	66,8% ±	96% ±
	3,42%	1,1%	4,72%	0,55%	3,7%	1%

Fonte: Autor.

#### 4.2.1.2 Conjunto com três classes

Em Tabela 21 verifica-se que o algoritmo FCN possui desempenho inferior ao algoritmo Resnet para todas as classes previstas para o experimento com 3 classes. Entretanto a diferença de performance entre os modelos é menor do que o experimento realizado com 4 classes principalmente, pela melhora do FCN em prever corretamente os apontamentos realizados para a classe 0%, embora tenha predito corretamente uma média de  $20,8\% \pm 27,49\%$  das amostras totais da classe 0%. O algoritmo Resnet teve um desempenho semelhante ao obtido no experimento anterior com destaque para uma melhora significativa nos resultados preditivos da classe 0%.

Tabela 21 - Desempenho por classe sob conjunto desbalanceado com 3 classes

Métricas	Precisão		Recall		F1-Score	
	FCN	Resnet	FCN	Resnet	FCN	Resnet
0%	99,4% ± 1,34%	100% ± 0%	20,8% ± 27,49%	99,6% ± 0,89%	28,8% ± 31,03%	99,8% ± 0,45%
3%	41,6% ± 8,71%	93,8% ± 1,64%	58,8% ± 10,26%	98,4% ± 0,89%	48,4% ± 7,44%	95,8% ± 0,45%
5%	63,4% ± 5,59%	98,4% ± 0,89%	71,8% ± 4,38%	93,2% ± 1,3%	67,2% ± 2,86%	95,6% ± 0,55%

Fonte: Autor.

#### 4.2.1.3 Conjunto com duas classes

Realizou-se 5 experimentos com conjunto de dados contendo duas classes, tendo em todos a presença da classe 0% e outra de porcentagem de falha. Dessa forma, apresenta-se em Tabela 22 o desempenho médio por classe em que os algoritmos obtiveram os piores resultados dentre os experimentos realizados com duas classes. Em contrapartida, em Tabela 23 é exposto o desempenho médio por classe em que os algoritmos obtiveram os melhores resultados dentre os experimentos realizados com duas classes.

Isso comprova por meio dos resultados do experimento que os algoritmos conseguiram realmente discernir mais facilmente amostras de velocidade estimada do vento entre classes que representam operações de falha da turbina eólica mais distantes fisicamente, o que era expectativa de resultados na teoria.

Tabela 22 - Pior desempenho por classe sob conjunto desbalanceado com 2 classes

Métricas	Precisão		Recall		F1-Score	
	FCN	Resnet	FCN	Resnet	FCN	Resnet
0%	0% ± 0%	87,6% ± 2,07%	0% ± 0%	68% ± 9,46%	0% ± 0%	76,2% ± 6,5%
1%	58% ± 0%	80,4% ± 4,34%	100% ± 0%	93% ± 1,41%	74% ± 0%	86% ± 2,35%

Fonte: Autor.

Tabela 23 - Melhor desempenho por classe sob conjunto desbalanceado com 2 classes

Métricas	Precisão		Recall		F1-Score	
	FCN	Resnet	FCN	Resnet	FCN	Resnet
0%	88,2% ± 6,76%	96,8% ± 1,64%	49,8% ± 27,31%	100% ± 0%	60,2% ± 27,22%	98,6% ± 0,55%
5%	72,8% ± 11,43%	100% ± 0%	95,8% ± 1,79%	97,2% ± 1,1%	82,6% ± 7,09%	98,6% ± 0,55%

Fonte: Autor.

#### 4.2.2 Conjuntos com equilíbrio de classes

O resultado médio final expresso pelas métricas de desempenho Acurácia, Precisão, Recall e F1-Score, das etapas de validação e teste, dos modelos de algoritmos treinados sob 5 repartições combinadas de dados de treino e validação com classes balanceadas, é apresentada em Tabela 24 para o algoritmo FCN e em Tabela 25 para o Resnet.

Tabela 24 - FCN sob conjuntos balanceados com quantidade de classes inferior a 6

N° Classes	Validação				Teste			
	Acurácia	Precisão	Recall	F1 - Score	Acurácia	Precisão	Recall	F1 Score
4 (0% 1% 3% 5%)	52,2% ± 3,9%	49,6% ± 4,56%	52,4% ± 4,16%	47,8% ± 4,38%	53% ± 3,39%	51,8% ± 5,07%	53,4% ± 3,78%	49,6% ± 4,51%
3 (0% 3% 5%)	63,4% ± 8,26%	63,6% ± 8,82%	63,4% ± 8,26%	61,4% ± 7,27%	57,8% ± 5,72%	57,8% ± 5,45%	59,2% ± 6,38%	57% ± 4,64%
2 (0% 1%)	61,2% ± 5,36%	61,8% ± 5,31%	61% ± 4,95%	60,2% ± 5,54%	56,2% ± 1,1%	53% ± 1,73%	52,8% ± 1,3%	51,2% ± 1,79%
2 (0% 2%)	68,2% ± 5,76%	69,4% ± 5,73%	68,2% ± 5,76%	67,6% ± 6,07%	65,6% ± 2,97%	65,8% ± 3,03%	62,6% ± 2,97%	61,8% ± 3,96%
2 (0% 3%)	73,6% ± 4,93%	74,8% ± 4,27%	73,4% ± 5,13%	73,2% ± 5,36%	68,4% ± 3,91%	68,2% ± 4,09%	66% ± 3,94%	66% ± 4,3%
2 (0% 4%)	83,8% ± 7,82%	84,8% ± 6,22%	83,8% ± 7,82%	83,6% ± 8,26%	77,6% ± 6,5%	78,8% ± 6,1%	75,4% ± 7,73%	76% ± 8,49%
2 (0% 5%)	84,4% ± 5,68%	87% ± 3,46%	84,4% ± 5,68%	83,8% ± 6,53%	77,8% ± 3,35%	79,6% ± 2,88%	75,6% ± 4,04%	76,2% ± 4,55%

Fonte: Autor.

Tabela 25 - Resnet sob conjuntos balanceados com quantidade de classes inferior a 6

N° Classes	Validação				Teste			
	Acurácia	Precisão	Recall	F1 - Score	Acurácia	Precisão	Recall	F1 Score
4 (0% 1% 3% 5%)	91% ± 1,87%	91,6% ± 1,95%	91% ± 1,87%	91% ± 1,87%	90,4% ± 1,14%	90,4% ± 1,14%	89,6% ± 0,89%	89,6% ± 0,89%
3 (0% 3% 5%)	98,2% ± 0,45%	98,4% ± 0,55%	98,4% ± 0,55%	98,4% ± 0,55%	96,8% ± 0,45%	97% ± 0%	96,8% ± 0,45%	96,8% ± 0,45%
2 (0% 1%)	86% ± 2,12%	86,2% ± 2,49%	86% ± 2,12%	86% ± 2,12%	85,8% ± 1,79%	85,4% ± 1,67%	85% ± 2,35%	85% ± 2%
2 (0% 2%)	98,2% ± 1,48%	98,4% ± 1,14%	98,2% ± 1,48%	98,2% ± 1,48%	97,2% ± 0,84%	97,2% ± 0,84%	97,2% ± 0,84%	97,2% ± 0,84%
2 (0% 3%)	99,4% ± 0,55%	99,4% ± 0,55%	99,4% ± 0,55%	99,4% ± 0,55%	99,6% ± 0,55%	99,6% ± 0,55%	99,6% ± 0,55%	99,6% ± 0,55%
2 (0% 4%)	100% ± 0%							
2 (0% 5%)	99,6% ± 0,55%	99,6% ± 0,55%	99,6% ± 0,55%	99,6% ± 0,55%	98,4% ± 0,55%	98,4% ± 0,55%	98,4% ± 0,55%	98,4% ± 0,55%

Fonte: Autor.

As quantidades de amostras utilizadas em cada experimento, bem como, as repartições de treino/validação e teste aplicada foi descrita no Capítulo 3. A proporção de amostras para todos os experimentos é de aproximadamente 65% de dados de treino, 16% de dados de validação e 19% de dados de teste.

Com o intuito de compreender a capacidade de predição dos algoritmos FCN e Resnet para as classes observadas, essa seção estará segmentada por resultados médios das métricas de desempenho por classe sob dados de teste para cada experimento contendo quantidades distintas de classes.

#### 4.2.2.1 Conjunto com quatro classes

Em Tabela 26 verifica-se que o algoritmo FCN possui desempenho médio inferior em comparação ao algoritmo Resnet no teste para todas as classes preditas no experimento com 4 classes balanceado. Entretanto, pode-se perceber uma melhora significativa do algoritmo FCN

se comparado com o mesmo modelo treinado em um conjunto de dados de 4 classes desbalanceado.

Tabela 26 - Desempenho por classe sob conjunto balanceado com 4 classes

Métricas	Precisão		Recall		F1-Score	
	FCN	Resnet	FCN	Resnet	FCN	Resnet
0%	47,8% ± 3,83%	84,8% ± 1,79%	52% ± 8,94%	76,8% ± 1,1%	50% ± 5,83%	80,6% ± 1,52%
1%	50,8% ± 3,7%	84,2% ± 0,84%	57% ± 4,64%	88,8% ± 1,79%	53,6% ± 2,88%	86,2% ± 1,48%
3%	49,8% ± 12,15%	93% ± 2,12%	17% ± 7,42%	98,8% ± 1,1%	25,2% ± 9,86%	95,8% ± 0,84%
5%	58,8% ± 3,7%	98,8% ± 1,1%	87,2% ± 4,71%	93,2% ± 1,3%	70,2% ± 3,7%	95,8% ± 0,84%

Fonte: Autor.

#### 4.2.2.2 Conjunto com três classes

Em Tabela 27 verifica-se que o algoritmo FCN possui desempenho médio inferior ao algoritmo Resnet na etapa de teste para todas as classes preditas para o experimento com 3 classes balanceado. Contudo, a diferença de desempenho médio para a classe 0% é a menor observada entre os modelos sob os experimentos realizados.

Tabela 27 - Desempenho por classe sob conjunto balanceado com 3 classes

Métricas	Precisão		Recall		F1-Score	
	FCN	Resnet	FCN	Resnet	FCN	Resnet
0%	65,8% ± 4,87%	100% ± 0%	76% ± 12,63%	100% ± 0%	70,2% ± 6,26%	100% ± 0%
3%	49% ± 12,57%	92,4% ± 0,55%	32,2% ± 8,79%	98,8% ± 1,1%	37,6% ± 6,58%	95,8% ± 0,45%
5%	58,4% ± 2,97%	98,8% ± 1,1%	69,8% ± 14,62%	91,8% ± 1,1%	63,2% ± 7,05%	94,8% ± 0,45%

Fonte: Autor.

### 4.2.2.3 Conjunto com duas classes

Com processo de experimentos semelhante ao realizado em 4.2.1.3., apresenta-se em Tabela 28 o desempenho médio por classe em que os algoritmos obtiveram os piores resultados dentre os experimentos realizados com duas classes balanceadas. Em contrapartida, em Tabela 29 é exposto o desempenho médio por classe em que os algoritmos obtiveram os melhores resultados dentre os experimentos realizados com duas classes balanceadas.

Novamente, pode-se verificar que o algoritmo FCN possui desempenho médio inferior em comparação ao algoritmo Resnet no teste para todas as classes preditas. Além disso, evidencia-se pelos resultados do experimento a expectativa de resultados teóricos como observado em 4.2.1.3.

Tabela 28 - Pior desempenho por classe sob conjunto balanceado com 2 classes

Métricas	Precisão		Recall		F1-Score	
	FCN	Resnet	FCN	Resnet	FCN	Resnet
0%	46% ± 2,24%	84,2% ± 3,56%	28,6% ± 6,39%	80,2% ± 5,72%	35,6% ± 4,93%	82% ± 2,92%
1%	59,6% ± 0,89%	86,6% ± 2,88%	76% ± 4,8%	89% ± 3,08%	67,2% ± 1,79%	87,8% ± 1,3%

Fonte: Autor.

Tabela 29 - Melhor desempenho por classe sob conjunto desbalanceado com 2 classes

Métricas	Precisão		Recall		F1-Score	
	FCN	Resnet	FCN	Resnet	FCN	Resnet
0%	83,4% ± 4,93%	96,8% ± 1,64%	61% ± 11,16%	99,6% ± 0,89%	69,8% ± 7,05%	98,4% ± 0,55%
5%	75,6% ± 4,72%	99,6% ± 0,89%	90,2% ± 4,55%	97,2% ± 1,1%	82,4% ± 2,3%	98,4% ± 0,55%

Fonte: Autor.

### 4.2.3 Comparação entre experimentos

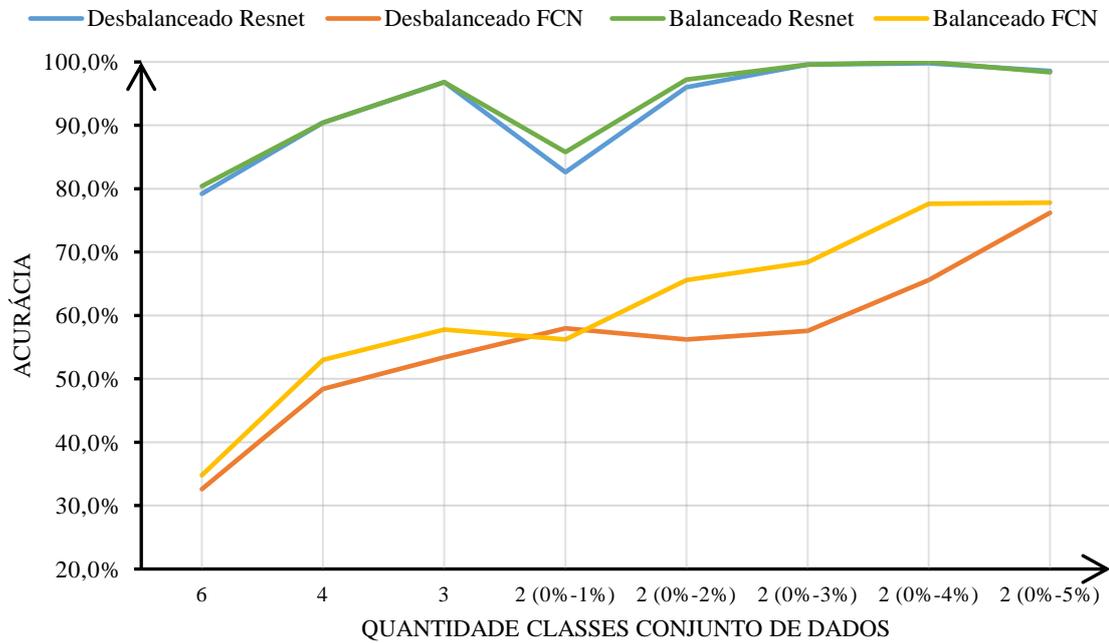
Em virtude da grande quantidade de resultados gerados, em consequência de variados experimentos realizados, possibilitando mensurar os impactos de utilizar-se técnica de balanceamento de classes entre amostras combinado aos algoritmos de aprendizado profundo de redes neurais FCN e Resnet ajustados sob conjunto de dados com quantidade de classes distintas, se faz necessário a comparação, por meio de um gráfico, do desempenho médio na etapa de teste dos experimentos o qual é ilustrado pela Figura 21, utilizando a métrica Acurácia, e pela Figura 22 utilizando a métrica F1-Score.

Dessa forma, pode-se verificar que os experimentos realizados com o algoritmo Resnet possuem padrões de desempenho médio similares ao utilizar a mensuração através das métricas F1-Score e Acurácia. Ressaltando-se apenas uma leve diferença de melhor desempenho utilizando a métrica F1-Score ao ser treinado com conjunto de dados de 4 e 3 classes desbalanceado.

Em contrapartida, os experimentos realizados com o algoritmo FCN possuem padrões de desempenho médio que divergem mais no tamanho da diferença entre o treinamento ser realizado com dados balanceados e desbalanceados ao utilizar a mensuração de F1-Score e Acurácia. Ressaltando-se que apenas quando treinado sob conjunto de dados de duas classes (0% e 1%) desbalanceadas, utilizando a métrica de Acurácia, o desempenho foi melhor que balanceado.

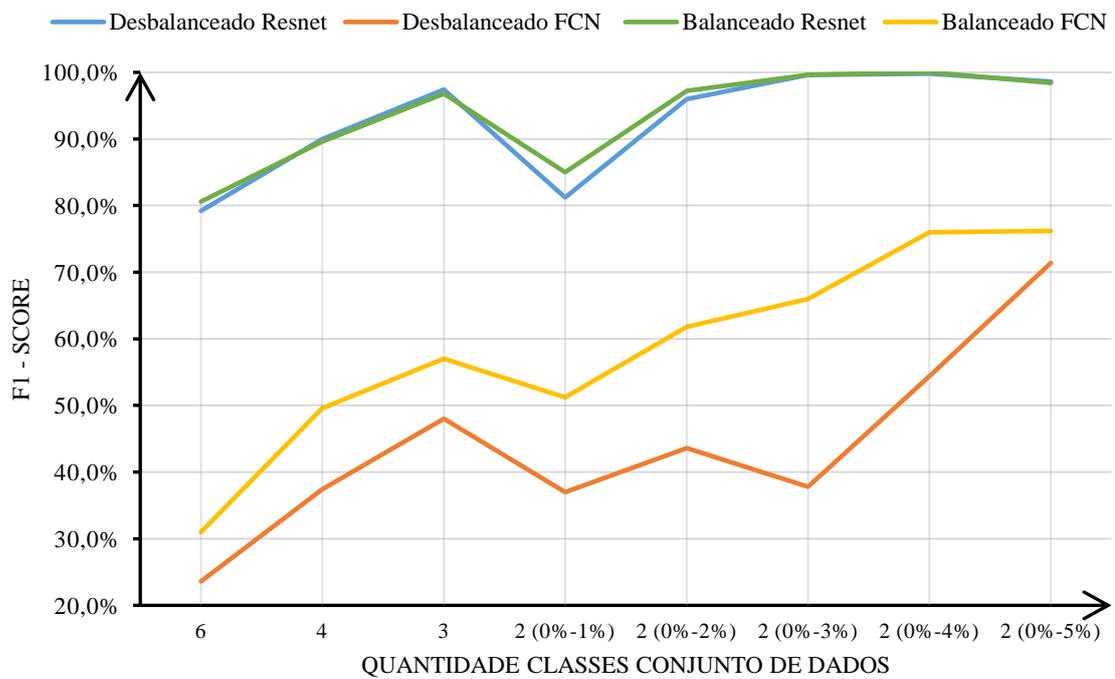
O resultado médio dos modelos, na etapa de teste, para todos os experimentos realizados que foram treinados sob 5 repartições combinadas de dados de treino e validação desbalanceados é apresentado no ÂPENDICE E. Além disso, o resultado médio, na etapa de teste, para os mesmos experimentos utilizando balanceamento de classes ao treinar os modelos é apresentado em ÂPENDICE F.

Figura 21 - Comparação entre experimentos utilizando a métrica acurácia



Fonte: Autor.

Figura 22 - Comparação entre experimentos utilizando a métrica F1- score



Fonte: Autor

## 5. CONCLUSÃO

Neste trabalho foi apresentado o fluxo de métodos implementados, bem como, a forma de funcionamento de cada um para realizar a classificação de falha de desbalanceamento de massa no rotor de aerogerador de características apresentadas na introdução.

Relativo à função de custo/perda a qual ilustra o comportamento dos modelos na etapa de treinamento, pode-se observar que o algoritmo de aprendizado profundo FCN ao ser ajustado através dos dados de treino/validação, tanto desbalanceados como balanceados, possuem uma redução mais lenta em comparação com o algoritmo Resnet. Isso fornece indícios os quais se comprovam posteriormente da melhor performance média e por classe nos experimentos realizados os quais o Resnet teve em comparação ao FCN.

Foram experimentados conjuntos de amostras com diferentes quantidades de classes constituintes, e pode-se verificar, a partir das métricas de avaliação, que à medida que o número de classes do conjunto a ser experimentado é menor, por meio da consequente exclusão de amostras contendo classes que caracterizam operações menos próximas fisicamente, o desempenho médio dos modelos de aprendizado profundo aumenta.

Ao comparar experimentos realizados com amostras contendo equilíbrio de classes e sem, pode-se constatar que os algoritmos de aprendizado profundo aplicado sob dados equilibrados conseguem aprender de forma homogênea todas as classes, conseguindo assim distingui-las com maior facilidade e dessa forma, obtêm melhores resultados nos testes com relação ao treinamento realizado sob o conjunto de amostras sem a aplicação do equilíbrio. O impacto da aplicação da técnica de balanceamento de classes mostrou-se ainda maior para o algoritmo FCN.

Além disso, observou-se que em todos experimentos contendo amostras da classe 0% e 1% de falha de desbalanceamento de massa nos quais não foram realizados o equilíbrio de amostras por classe e utilizou-se o algoritmo FCN, a classe 0% (condição normal) não conseguiu ser identificada pelo classificador em nenhuma observação testada.

Pode-se observar por meio dos desvios padrão dos resultados médios de desempenho que o algoritmo FCN obteve maior variabilidade na tarefa de classificação por meio de seus modelos treinados sob cinco combinações segmentadas entre treino e validação, em comparação com o algoritmo Resnet. O que demonstra maior capacidade de generalização do

algoritmo de aprendizado profundo Resnet, ou seja, o poder de responder de maneira adequada a situações que o algoritmo não teve acesso em seu aprendizado.

O algoritmo de aprendizado profundo Resnet foi o que obteve os melhores desempenhos médios nos experimentos realizados sob diferentes cenários de quantidade de classes e balanceamento.

Por fim, o trabalho desenvolvido atingiu os objetivos gerais propostos, bem como, os específicos inicialmente. Para isso foram seguidas práticas recomendadas da biblioteca *Scikitlearn* para testes experimentais de modo a tornar os resultados significativos de acordo com a real robustez dos modelos e técnicas empregadas, evitando erros de ordem sequencial dos métodos aplicados.

Dessa forma, foi possível concluir que os experimentos utilizando de conjunto de amostras equilibradas com o algoritmo de aprendizado profundo Resnet conseguiu obter os melhores desempenhos médios para os testes realizados.

Para trabalhos futuros é sugerido em caso de utilizar-se a mesma base de dados e problemática adotada no trabalho em questão, alterar a modelagem para dividir as tarefas em duas: primeiramente apenas a detecção de falha mediante o agrupamento de todas as amostras contendo algum nível de falha sendo rotuladas com mesma variável de saída e amostras normais agrupadas com outro rótulo, gerando dessa forma a necessidade de classificação binária. Como segundo passo, a sugestão seria treinar um outro modelo para classificar apenas as observações que o primeiro modelo indicou como sendo com falha para assim indicar que nível de falha existe, gerando dessa forma a necessidade de classificação multiclasse.

## REFERÊNCIAS

ABEEólica; **Infovento 25**, Disponível em <[https://abeeolica.org.br/wp-content/uploads/2022/04/2022\\_03\\_InfoVento25.pdf](https://abeeolica.org.br/wp-content/uploads/2022/04/2022_03_InfoVento25.pdf)>. Acesso em: 15 julho 2022.

ABEEólica; **Infovento 21**, Disponível em <[http://abeeolica.org.br/wp-content/uploads/2021/06/2021\\_06\\_InfoVento21.pdf](http://abeeolica.org.br/wp-content/uploads/2021/06/2021_06_InfoVento21.pdf)>. Acesso em: 04 setembro 2021.

ABDEL-HAMID, O.; DENG, L.; YU, D. **Exploring convolutional neural network structures and optimization techniques for speech recognition**. Interspeech. 2013;2013:1173-1175.

AIRES, L.F. **Utilização de redes neurais convolucionais para diagnóstico de desbalanço de massa em rotores de turbinas eólicas**. 2021. 70 p. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Universidade Federal de Santa Maria, Santa Maria, RS, 2021.

BAGNALL, A.; DAVIS, L.M.; HILLS, J.; LINES, J. **Transformation Based Ensembles for Time Series Classification**. SDM. 2012.

CASELITZ, P.; GIEBHARDT, J. **Rotor condition monitoring for improved operational safety of offshore wind energy converters**. Journal of Solar Energy Engineering-Transactions of the Asme, vol. 127, pp. 253-261, May 2005.

CENTER FOR CLIMATE AND ENERGY SOLUTIONS. **Types of Renewable Energy**. Disponível em: <<https://www.c2es.org/content/renewable-energy/>>. Acesso em: 8 jun. 2021.

CHAWLA, N.V.; BOWYER, K.W.; HALL, L.O.; KEGELMEYER, W.P. **SMOTE: Synthetic Minority Over-sampling Technique**. Journal of Artificial Intelligence Research 16, pp. 321–357, 2002.

COSTA, Á.M.; OROSA, J.A.; VERGARA, D.; FERNÁNDEZ-ARIAS, P. **New Tendencies in Wind Energy Operation and Maintenance**. Appl. Sci. 2021.

da ROSA, L. D. **Detecção de desbalanceamento de massa entre pás de aerogerador utilizando sinais elétricos de um gerador síncrono de ímãs permanentes aplicados a métodos de aprendizagem de máquina**. 2019. 83 p. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Santa Maria, Santa Maria, RS, 2019.

da SILVA, A. B. F. **Manutenção de turbinas eólicas: levantamento e análise da incidência de falhas nos seus subsistemas**. 2017. 120 p. Trabalho de Conclusão de Curso (Graduação em Tecnologia em Energias Renováveis) – Instituto Federal de Educação do Rio Grande do Norte, João Câmara, RN, 2017.

de OLIVEIRA, S. F. P. S. **Análise do comportamento dos aerogeradores em situação de anomalia**. 2013. 122 p. Dissertação (Mestrado em Engenharia Electrotécnica - Sistemas Eléctricos de Energia) – Instituto Superior de Engenharia do Porto, Porto, Portugal, 2013.

de SOUZA, A. J. **Comportamento dinâmico em torres de aerogeradores**. 2018. 90 p. Dissertação (Pós-Graduação em Engenharia Mecânica) – Universidade Federal do Rio Grande do Norte, Natal, RN, 2018.

DELAZZERI, K.A. **Máquina de vetores de suporte e técnicas estatísticas para previsão e classificação de falha de turbinas eólicas**. 2020. 99 p. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Santa Maria, Santa Maria, RS, 2020.

FAWAZ, H. I.; FORESTIER, G.; WEBER, J.; IDOUMGHAR, L.; MULLER, P.A. **Deep Learning for time series classification: A review**. *Data Min. Knowl. Discov.* 2019, 33, 917–963.

GOMES, F. da S. V. **Modelos de series temporais para previsão de nível de líquidos em cadinho de altos-fornos**. Dissertação (Doutorado) – Universidade Federal do Espírito Santo. 2016.

GOOGLE. Disponível em: <<https://colab.research.google.com/notebooks/intro.ipynb>>. Acesso em: 12 de julho de 2021.

GUPTA, R.Y.; MUDIGONDA, S.; BARUAH, P.K. **Integrating actuarial models with neural networks for building a fraud detection model for automobile insurance**. 2021.

GWEC; **Annual Wind Report 2022**, Disponível em < [https://gwec.net/wp-content/uploads/2022/04/Annual-Wind-Report-2022\\_screen\\_final\\_April.pdf](https://gwec.net/wp-content/uploads/2022/04/Annual-Wind-Report-2022_screen_final_April.pdf)>. Acesso em: 15 julho 2022.

HAHN, B.; DURSTEWITZ, M.; ROHRIG, K. **Reliability of wind turbines in Wind energy**. pp. 329–332, Springer: Berlin/Heidelberg, Germany, 2007.

HOWARD, R. M. **Principles of random signal analysis and low noise design: The power spectral density and its applications**. [S.l.]: John Wiley & Sons, 2004.

HÜBNER, G. R. **Diagnóstico de desequilíbrio de massa para rotores de aerogeradores utilizando máquina de vetores de suporte**. 2021. 132 p. Dissertação (Pós-Graduação em Engenharia Elétrica, Processamento de Energia Elétrica) – Universidade Federal de Santa Maria, Santa Maria, RS, 2021.

IEA. **Renewable Energy Market Update**. 2022. Disponível em: <<https://www.iea.org/reports/renewable-energy-market-update-may-2022>>. Acesso em: 17 jul. 2022.

INSTRUMENTS, N. **Understanding fft and windowing**. White Pap In, 2015.  
JAPKOWICZ, N.; STEPHEN, S. The class imbalance problem: A systematic study. *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, Nov. 2002, doi: 10.3233/IDA-2002-6504.

IOFFE, S.; SZEGEDY, C. **Batch normalization: Accelerating deep network training by reducing internal covariate shift.** arXiv preprint arXiv:1502.03167, 2015.

JÚNIOR, F. E. de A. M. **Avaliação das condições de operação das pás de uma turbina eólica em escala: aplicação da dfa combinada com técnicas de reconhecimento de padrões dos sinais de vibração.** 2017. 95 p. Dissertação (Pós- Graduação em Engenharia Mecânica) – Universidade Federal do Ceará, Fortaleza, CE, 2017.

JUNIOR, D. **Precision, Recall ou F1-Score ? Qual a melhor métrica para utilizar em seu modelo?**, 9 Junho de 2022. Disponível em: < <https://medium.com/@djalmajr07/precision-recall-ou-f1-score-qual-a-melhor-m%C3%A9trica-para-utilizar-em-seu-modelo-e0d87877ed55>>. Acesso em: 25 de julho de 2022.

KHALFALLAH, M. G.; KOLIUB, A. M. **Effect of dust on the performance of wind turbines.** Desalination, 209-220, 2007.

LI, S.; CHOWDHURY, R.R.; SHANG, J.; GUPTA, R. K.; HONG, D. **UniTS: Short-Time Fourier Inspired Neural Networks for Sensory Time Series Classification.** SenSys. 2021.

LIN, M.; CHEN, Q.; YAN, S. **Network in network.** arXiv preprint arXiv:1312.4400, 2013.

MALIK, H.; MISHRA, S. **Artificial neural network and empirical mode decomposition based imbalance fault diagnosis of wind turbine using TurbSim, FAST and Simulink.** IET Renewable Power Generation, v. 11, n. 6, p. 889-902, 2016.

MANCHEV, N. **ML internals: Synthetic Minority Oversampling (SMOTE) Technique.** Domino Data Lab. May 20, 2021. Disponível em: < <https://www.dominodatalab.com/blog/smote-oversampling-technique> >. Acesso em: 18 de julho de 2022.

MONTANGER, P.O. **Detecção automática de exoplanetas por meio de algoritmos de aprendizado de máquina baseados em séries temporais.** 2021. 89 p. Trabalho de Conclusão de Curso (Graduação em Engenharia Física) – Universidade Federal da Integração Latino-Americana, Foz do Iguaçu, PR, 2021.

MORETTIN, P.A.; TOLOI, C. M. de C. **Análise de séries temporais.** São Paulo: Edgard Blucher. 2004. Acesso em: 28 jul. 2022.

MÜLLER, A.C. **Cross-Validation Strategies.** 2020. Disponível em: <https://amueller.github.io/aml/04-model-evaluation/1-data-splitting-strategies.html>. Acesso em: 14 julho 2022.

NAKANO, K.; CHAKRABORTY, B. **Effect of Data Representation for Time Series Classification - A Comparative Study and a New Proposal.** Mach. Learn. Knowl. Extr., 1, 1100-1120. 2019.

ORELLANA, E. **SMOTE.** Medium, 2020. Disponível em: <https://emilia-orellana44.medium.com/smote-2acd5dd09948>. Acesso em: 19 julho 2022.

PENHA, D. de P. **Rede neural convolucional aplicada à identificação de equipamentos residenciais para sistemas de monitoramento não-intrusivo de carga**. 71 p. Dissertação (Pós-Graduação em Engenharia Elétrica) – Universidade Federal do Pará, Belém, PA, 2018.

PEREIRA, A. de L. (Brasília). Senai (Org.). **Operação de Parques Eólicos**: Brasília: Cooperação Alemã, 2015. 60 slides, color.

PÉREZ, J. M. P.; MÁRQUEZ, F. P. G.; TOBIAS, A.; PAPAELIAS, M. **Wind turbine reliability analysis**. *Renewable and Sustainable Energy Reviews*, vol. 23, pp. 463 – 472, 2013. *Proceedings. 2003 Symposium on*, p. 209–216, January 2003.

PROVOST, F.; KOHAVI, R. **Guest editors' introduction: On applied research in machine learning**. *Machine learning*, v. 30, n. 2, p. 127-132, 1998.

RAD, N.M.; KIA, S.M.; ZARBO, C.; van LAARHOVEN, T.; JURMAN, G.; VENUTI, P.; et al. **Deep learning for automatic stereotypical motor movement detection using wearable sensors in autism spectrum disorders**. *Signal Processing*. 2018;144:180-191.

RIBRANT, J.; BERTLING, L. M. **Survey of failures in wind power systems with focus on swedish wind power plants during 1997-2005**. *IEEE Transactions on Energy conversion*, vol. 22, no. 1, pp. 167–173, 2007.

RODRIGUES, V. **Métricas de Avaliação: acurácia, precisão, recall... quais as diferenças?**, 12 Abril de 2019. Disponível em: <<https://vitorborbarodrigues.medium.com/m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-acur%C3%A1cia-precis%C3%A3o-recall-quais-as-diferen%C3%A7as-c8f05e0a513c>>. Acesso em: 25 de julho de 2022.

SADOUK, L. **CNN Approaches for Time Series Classification**. In: Ngan, C., editor. *Time Series Analysis - Data, Methods, and Applications* [Internet]. London: IntechOpen; 2018. Acesso em: 20 julho de 2022. Disponível em: <<https://www.intechopen.com/chapters/64216>> doi: 10.5772/intechopen.81170.

SADOUK, L.; GADI, T.; ESSOUFI, E.H. **A novel deep learning approach for recognizing stereotypical motor movements within and across subjects on the autism spectrum disorder**. *Computational Intelligence and Neuroscience*. 2018;16. DOI: 10.1155/2018/7186762. Article ID 7186762.

SAREEN, A.; SAPRE, C. A.; SELIG, M. S. **Effects of leading edge erosion on wind turbine blade performance**. *Wind Energy*, 1531-1542, 2014.

SCIKIT-LEARN. **StratifiedKfold**. Disponível em: <[https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.StratifiedKfold.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKfold.html)>. Acesso em: 17 jul. 2022.

SINHA, Y.; STEEL, J. **A progressive study into offshore wind farm maintenance optimisation using risk based failure analysis**. *Renewable and Sustainable Energy Reviews*, vol. 42, pp. 735 – 742, 2015.

TCHAKOUA, P.; WAMKEUE, R.; OUHROUCHE, M.; SLAOUI-HASNAOUI, F.; TAMEGHE, T. A.; EKEMB, G.; **Wind Turbine Condition Monitoring: State-of-the-Art Review, New Trends, and Future Challenges**. MDPI, Journal, Energies, 2014.

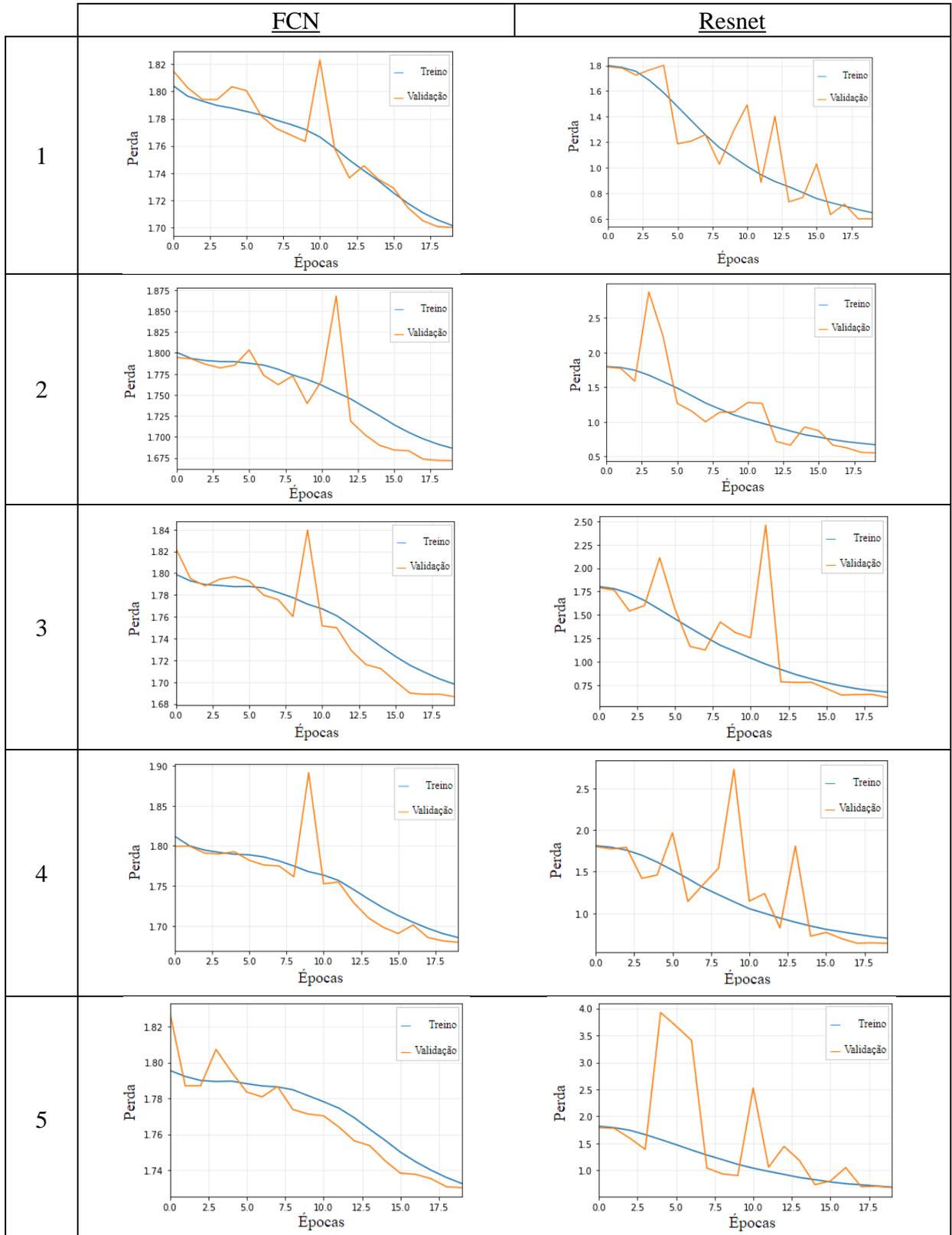
WANG, S.; LIU, P.; SHE, M.F.H.; et al. **Bagof-words representation for biomedical time series classification**. *Biomedical Signal Processing and Control*. 2013; 8(6):634-644.

WANG, Z.; YAN, W.; OATES, T. **Time series classification from scratch with deep neural networks: A strong baseline**. 1578-1585. 10.1109/IJCNN.2017.7966039. 2017.

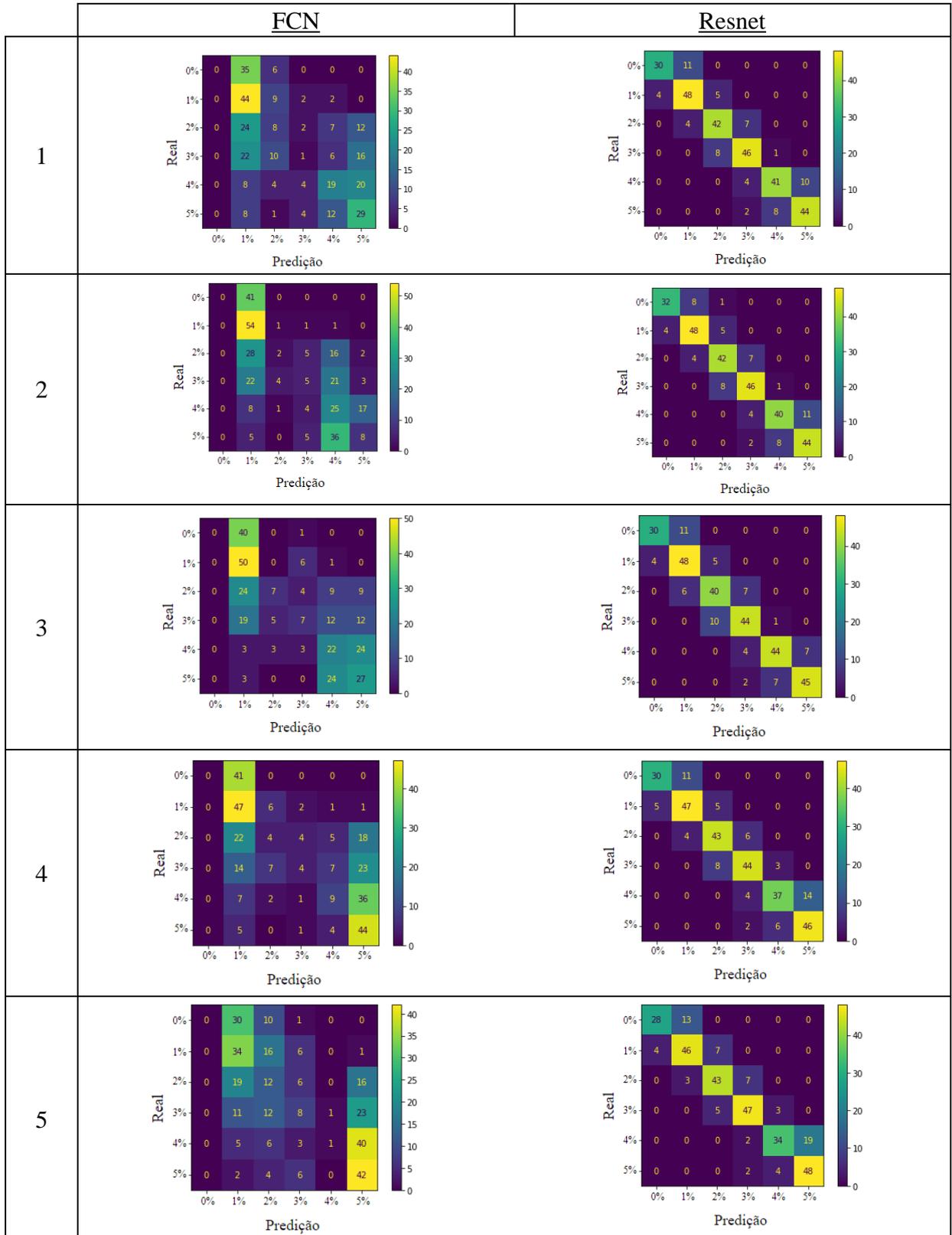
WIZELIUS T. **Developing Wind Power Projects: Theory and Practice**. 1st edn. London: Routledge, 2007.

YANG, Q.; WU, X. **10 Challenging problems in data mining research**. *International Journal of Information Technology and Decision Making* 5 (4), 597–604. 2006.

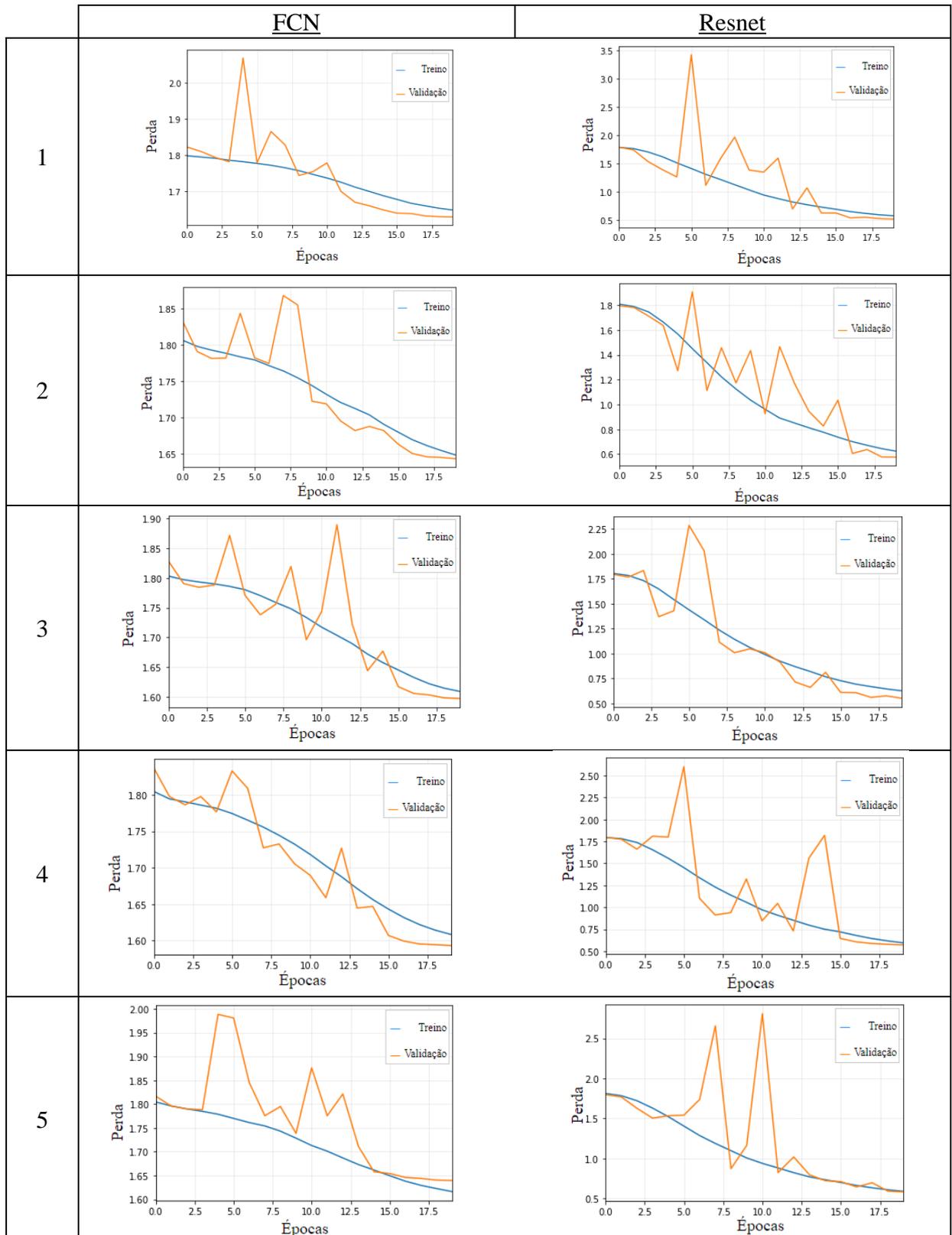
**APÊNDICE A – FUNÇÃO DE CUSTO/PERDA PARA SEIS CLASSES  
DESBALANCEADAS**



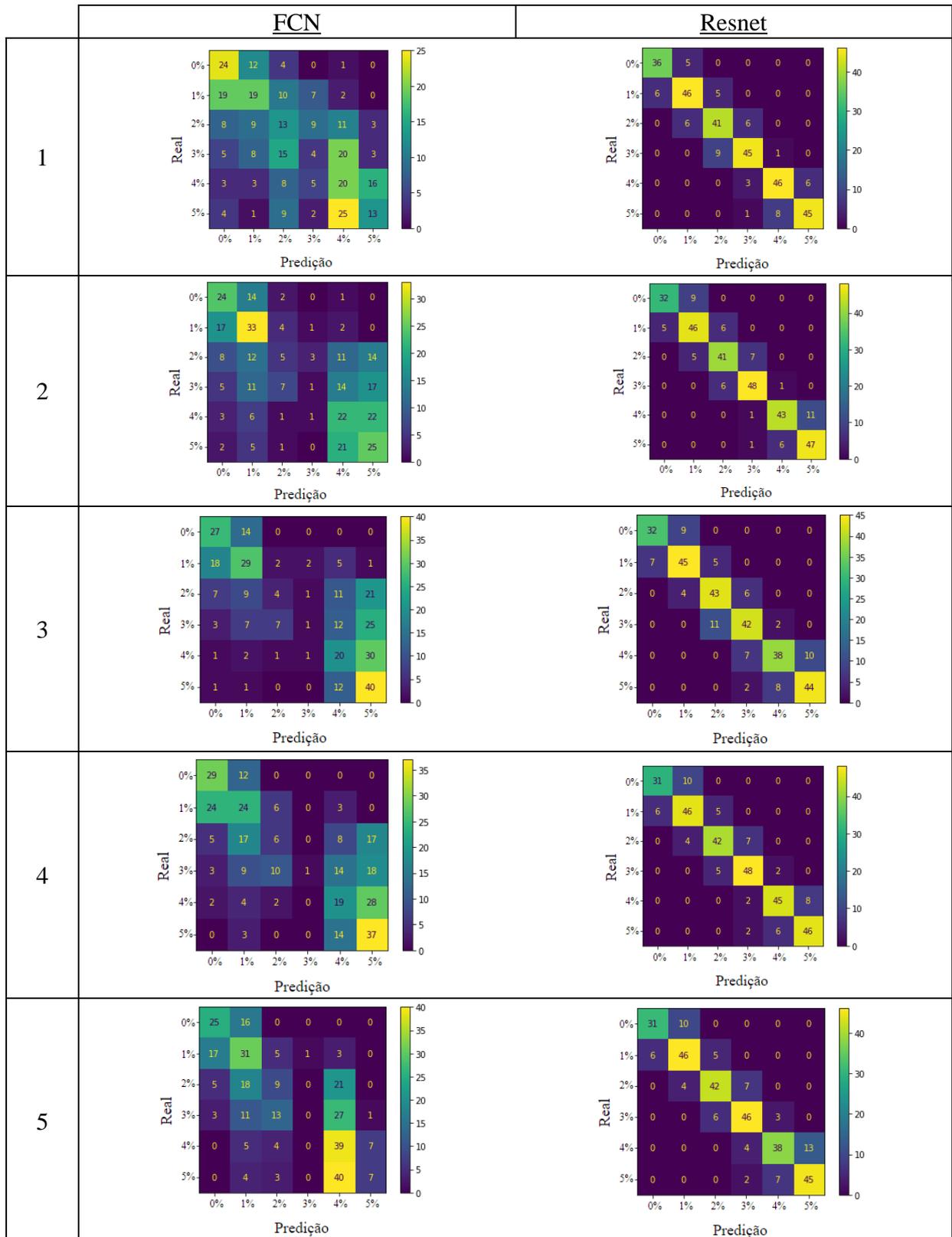
## APÊNDICE B – MATRIZ DE CONFUSÃO TESTE PARA SEIS CLASSES DESBALANCEADAS



**APÊNDICE C – FUNÇÃO DE CUSTO/PERDA PARA SEIS CLASSES  
BALANCEADAS**



## APÊNDICE D – MATRIZ DE CONFUSÃO TESTE PARA SEIS CLASSES BALANCEADAS



**APÊNDICE E – COMPARAÇÃO GERAL EXPERIMENTOS TREINADOS COM  
DADOS DESBALANCEADOS**

N° Classes	Algoritmo	Teste			
		Acurácia	Precisão	Recall	F1 - Score
6 (0% 1% 2% 3% 4% 5%)	FCN	32,6% ± 2,41%	26,2% ± 3,35%	30,8% ± 2,59%	23,6% ± 2,97%
	Resnet	79,2% ± 1,10%	80% ± 1,00%	78,8% ± 0,84%	79,2% ± 1,10%
4 (0% 1% 3% 5%)	FCN	48,4% ± 3,21%	37% ± 3,94%	45,2% ± 2,77%	37,4% ± 3,21%
	Resnet	90,4% ± 1,14%	90,8% ± 0,84%	89,4% ± 1,14%	90% ± 1%
3 (0% 3% 5%)	FCN	53,4% ± 8,88%	68,2% ± 3,96%	50,4% ± 10,53%	48% ± 12,83%
	Resnet	96,8% ± 0,45%	97,6% ± 0,55%	97,4% ± 0,89%	97,4% ± 0,89%
2 (0% 1%)	FCN	58% ± 0%	29% ± 0%	50% ± 0%	37% ± 0%
	Resnet	82,6% ± 3,97%	83,8% ± 2,77%	80,2% ± 4,44%	81,2% ± 4,44%
2 (0% 2%)	FCN	56,2% ± 2,49%	44,6% ± 15,87%	51,4% ± 1,95%	43,6% ± 6,99%
	Resnet	96% ± 0%	96% ± 0%	95,4% ± 0,55%	96% ± 0%
2 (0% 3%)	FCN	57,6% ± 0,89%	49% ± 27,39%	50,6% ± 0,89%	37,8% ± 2,68%
	Resnet	99,6% ± 0,55%	99,6% ± 0,55%	99,6% ± 0,55%	99,6% ± 0,55%
2 (0% 4%)	FCN	65,6% ± 5,55%	80,6% ± 3,13%	60% ± 6,04%	54,4% ± 10,14%
	Resnet	99,8% ± 0,45%	99,8% ± 0,45%	99,8% ± 0,45%	99,8% ± 0,45%
2 (0% 5%)	FCN	76,2% ± 11,76%	80,4% ± 8,88%	73% ± 13,66%	71,4% ± 16,59%
	Resnet	98,6% ± 0,55%	98,6% ± 0,55%	98,6% ± 0,55%	98,6% ± 0,55%

**APÊNDICE F – COMPARAÇÃO GERAL EXPERIMENTOS TREINADOS COM  
DADOS BALANCEADOS**

N° Classes	Algoritmo	Teste			
		Acurácia	Precisão	Recall	F1 Score
6 (0% 1% 2% 3% 4% 5%)	FCN	34,8% ± 3,49%	34,6% ± 6,73%	36% ± 3,08%	31% ± 1,58%
	Resnet	80,4% ± 2,30%	80,6% ± 1,95%	80,4% ± 1,82%	80,6% ± 1,95%
4 (0% 1% 3% 5%)	FCN	53% ± 3,39%	51,8% ± 5,07%	53,4% ± 3,78%	49,6% ± 4,51%
	Resnet	90,4% ± 1,14%	90,4% ± 1,14%	89,6% ± 0,89%	89,6% ± 0,89%
3 (0% 3% 5%)	FCN	57,8% ± 5,72%	57,8% ± 5,45%	59,2% ± 6,38%	57% ± 4,64%
	Resnet	96,8% ± 0,45%	97% ± 0%	96,8% ± 0,45%	96,8% ± 0,45%
2 (0% 1%)	FCN	56,2% ± 1,1%	53% ± 1,73%	52,8% ± 1,3%	51,2% ± 1,79%
	Resnet	85,8% ± 1,79%	85,4% ± 1,67%	85% ± 2,35%	85% ± 2%
2 (0% 2%)	FCN	65,6% ± 2,97%	65,8% ± 3,03%	62,6% ± 2,97%	61,8% ± 3,96%
	Resnet	97,2% ± 0,84%	97,2% ± 0,84%	97,2% ± 0,84%	97,2% ± 0,84%
2 (0% 3%)	FCN	68,4% ± 3,91%	68,2% ± 4,09%	66% ± 3,94%	66% ± 4,3%
	Resnet	99,6% ± 0,55%	99,6% ± 0,55%	99,6% ± 0,55%	99,6% ± 0,55%
2 (0% 4%)	FCN	77,6% ± 6,5%	78,8% ± 6,1%	75,4% ± 7,73%	76% ± 8,49%
	Resnet	100% ± 0%	100% ± 0%	100% ± 0%	100% ± 0%
2 (0% 5%)	FCN	77,8% ± 3,35%	79,6% ± 2,88%	75,6% ± 4,04%	76,2% ± 4,55%
	Resnet	98,4% ± 0,55%	98,4% ± 0,55%	98,4% ± 0,55%	98,4% ± 0,55%