



GABRIEL APARECIDO FONSECA

**CLASSIFICAÇÃO DE FALTAS EM LINHAS DE
TRANSMISSÃO UTILIZANDO MÉTODOS DE
APRENDIZADO DE MÁQUINA**

LAVRAS – MG

2021

GABRIEL APARECIDO FONSECA

**CLASSIFICAÇÃO DE FALTAS EM LINHAS DE TRANSMISSÃO UTILIZANDO
MÉTODOS DE APRENDIZADO DE MÁQUINA**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, área de concentração em Sistemas Inteligentes, para a obtenção do título de Mestre.

Prof. DSc. Danton Diego Ferreira
Orientador

Prof. DSc. Aryfrance Rocha Almeida
Coorientador

Prof. DSc. Flávio Bezerra Costa
Coorientador

LAVRAS – MG

2021

**Ficha Catalográfica preparada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca
Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).**

Fonseca, Gabriel Aparecido

Classificação de faltas em linhas de transmissão utilizando métodos de aprendizado de máquina / . 1^a ed. rev., atual. e ampl. – Lavras : UFLA, 2021.

231 p. : il.

Dissertação (Mestrado acadêmico)–Universidade Federal de Lavras, 2021.

Orientador: Prof. DSc. Danton Diego Ferreira.

Bibliografia.

1. Classificação de faltas. 2. Linhas de transmissão. 3. Rocket. I. Ferreira, Danton Diego. II. Almeida, Aryfrance Rocha. III. Costa, Flávio Bezerra

GABRIEL APARECIDO FONSECA

**CLASSIFICAÇÃO DE FALTAS EM LINHAS DE TRANSMISSÃO UTILIZANDO
MÉTODOS DE APRENDIZADO DE MÁQUINA
CLASSIFICATION OF TRANSMISSION LINE FAULTS USING MACHINE
LEARNING METHODS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, área de concentração em Sistemas Inteligentes, para a obtenção do título de Mestre.

APROVADA em 03 de Dezembro de 2021.

Prof. DSc. Danton Diego Ferreira	UFLA
Prof. DSc. Leandro Rodrigues Manso Silva	UFJF
Profa. DSc. Silvia Costa Ferreira	UFLA
Prof. DSc. Aryfrance Rocha Almeida	UFPI
Prof. DSc. Flávio Bezerra Costa	UFRN



Prof. DSc. Danton Diego Ferreira
Orientador



Prof. DSc. Aryfrance Rocha Almeida
Co-Orientador



Prof. DSc. Flávio Bezerra Costa
Co-Orientador

**LAVRAS – MG
2021**

A minha família, amigos e equipe do projeto

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus pelo auxílio nos momentos difíceis dessa jornada. Sem Ele eu não teria chegado até aqui.

Em segundo lugar, agradeço aos meus pais Antônio e Janete por terem me dado suporte, me ajudado e apoiado minhas decisões durante essa vida. Sem eles eu não teria o direcionamento necessário para chegar até onde cheguei.

Por fim, gostaria de agradecer ao meu orientador Danton, e coorientadores Flávio e Aryfrance, por todo suporte que me deram. As dúvidas que me ajudaram a sanar e o apoio que possibilitou a realização desse projeto.

Uma longa viagem começa com um único passo. (Lao-Tsé)

RESUMO

As linhas de transmissão de energia elétrica são componentes altamente susceptíveis a faltas. Diversos fatores como animais, falha humana e descargas elétricas podem levar a essas ocorrências. Além disso, a crescente demanda de geração, distribuição e transmissão de energia elétrica tem contribuído cada vez mais para que esse se torne um problema recorrente. Diversos trabalhos já exploraram o uso de inteligência computacional, processamento de sinais e outras técnicas na construção de métodos protetivos para rápida verificação e atuação na ocorrência de faltas nas linhas de transmissão. Muitos desses trabalhos focam em abordagens utilizando processamento de sinais como transformada de Fourier ou *wavelet*. Com o avanço do aprendizado de máquina, algumas técnicas clássicas passaram a ser utilizadas nessa área obtendo êxito. Esse trabalho foca na classificação *offline* de dez (AT, BT, CT, AB, AC, BC, ABT, ACT, BCT e ABC) tipos de faltas que surgem quando ocorre um curto-circuito na linha de transmissão, investigando o uso de técnicas clássicas como filtro *notch* e florestas aleatórias. Para efeito comparativo, foram utilizadas técnicas criadas recentemente, chamadas de *Rocket* e *MiniRocket*, para a extração de características em séries temporais e bons resultados foram obtidos na identificação das faltas ocorridas na linha de transmissão. Como resultado dessa dissertação, acurácias superiores a 93% foram alcançadas considerando-se até 1/16 ciclo pós falta. Para sinais com 1 e 1/2 ciclo pós falta foram obtidas acurácias superiores a 97%.

Palavras-chave: Classificação de faltas. Linhas de transmissão. Filtro *notch*. *MiniRocket*. *Rocket*.

ABSTRACT

Power transmission lines are components highly susceptible to faults. Several factors such as animals, human failure and lightning can lead to the occurrence of a fault. In addition, the increasing demand for electricity generation, distribution and transmission has contributed to this becoming a recurrent problem. Several works have already explored the use of computational intelligence, signal processing and other techniques in the construction of protective methods for quick verification and action in the occurrence of transmission line fault. Many of these works focus on approaches using signal processing such as Fourier or wavelet transforms. With the advance of machine learning, some classic techniques started to be used in this area with success. This work focuses on the offline classification of ten (AG, BG, CG, AB, AC, BC, ABG, ACG, BCG and ABC) types of faults that arise when a short circuit occurs in the transmission line, investigating the use of classical techniques such as notch filter and random forests. For comparative purposes, recently created techniques, called Rocket and MiniRocket, were used to extract features in time series and good results were obtained in the identification of faults that occurred in the transmission line. As a result of this dissertation, accuracies greater than 93% were obtained considering up to 1/16 cycle post-fault. For signals with 1 and 1/2 cycle post-fault, accuracies higher than 97% were obtained.

Keywords: Rocket. MiniRocket. Fault classification. Transmission lines. Notch filter. Random Forests.

LISTA DE FIGURAS

Figura 1.1 – Número de Perturbações por Tipo de Equipamento Origem.	25
Figura 1.2 – Tipos de faltas em linhas de transmissão.	27
Figura 3.1 – Resposta em frequência de um filtro <i>notch</i>	42
Figura 3.2 – Diagrama de Árvore de Decisão.	53
Figura 3.3 – Fluxograma do processo de <i>bagging</i>	56
Figura 3.4 – Arquitetura de um único neurônio.	57
Figura 3.5 – Arquitetura de uma Rede Neural Artificial.	58
Figura 3.6 – Fluxograma para validação cruzada.	62
Figura 3.7 – Validação cruzada 5-fold.	64
Figura 3.8 – t-SNE aplicado ao <i>dataset</i> MNIST	70
Figura 4.1 – Diagrama unifilar para o sistema de energia trifásico de 500 kV.	73
Figura 4.2 – Comportamento da tensão em uma falta ABC 20km 0° 1 Ω.	74
Figura 4.3 – Comportamento da corrente em uma falta ABC 20km 0° 1 Ω.	74
Figura 4.4 – Sistema de linha de transmissão com compensação em série.	75
Figura 4.5 – Diagrama de um sistema compensado em série.	76
Figura 4.6 – Comportamento da tensão em uma falta ABC 20 km 0° 1 Ω 0,7 de compen- sação.	78
Figura 4.7 – Comportamento da corrente em uma falta ABC 20 km 0° 1 Ω 0,7 de com- pensação.	79
Figura 4.8 – Fluxograma da validação cruzada para o método A.	81
Figura 4.9 – Fluxograma da validação cruzada para os métodos B, C, D, E, F e G.	82
Figura 4.10 – Fluxograma das etapas do método A.	83
Figura 4.11 – Comportamento da tensão de uma falta ABC 20km 0° 1 Ω após aplicação do filtro <i>notch</i>	84
Figura 4.12 – Fluxograma das etapas do método B.	87
Figura 4.13 – Fluxograma das etapas do método C.	88
Figura 4.14 – Fluxograma das etapas do método F.	90
Figura 4.15 – Fluxograma das etapas do método G.	91
Figura 5.1 – t-SNE para dados de treinamento com 1 ciclo pós falta.	93
Figura 5.2 – t-SNE para dados de treinamento com 1/2 ciclo pós falta.	93
Figura 5.3 – t-SNE para dados de treinamento com 1/4 ciclo pós falta.	94

Figura 5.4 – t-SNE para dados de treinamento com 1/8 ciclo pós falta.	94
Figura 5.5 – t-SNE para dados de treinamento com 1/16 ciclo pós falta.	95
Figura 5.6 – t-SNE para dados de treinamento com 1/32 ciclo pós falta.	95
Figura 5.7 – t-SNE para dados de treinamento com 1/64 ciclo pós falta.	96
Figura 5.8 – t-SNE para dados de treinamento com 1/128 ciclo pós falta.	96

LISTA DE TABELAS

Tabela 3.1 – Comparativo entre métodos <i>Rocket</i> e <i>MiniRocket</i>	47
Tabela 4.1 – Parâmetros utilizados classificador <i>random forest</i> implementado no método A.	85
Tabela 4.2 – Parâmetros utilizados na rede neural implementada no método A.	85
Tabela 4.3 – Parâmetros utilizados nos classificadores <i>random forest</i> implementados no método B.	87
Tabela 4.4 – Parâmetros utilizados nos classificadores <i>random forest</i> implementados no método C.	88
Tabela 4.5 – Parâmetros utilizados nos classificadores <i>random forest</i> implementados no método D.	89
Tabela 4.6 – Parâmetros utilizados nos classificadores <i>random forest</i> implementados no método E.	89
Tabela 5.1 – Resumo das principais diferenças de cada modelo implementado.	97
Tabela 5.2 – Matriz de confusão para o modelo de florestas aleatórias no conjunto de validação.	98
Tabela 5.3 – Matriz de confusão para o modelo de redes neurais no conjunto de validação.	98
Tabela 5.4 – Resultados obtidos pelo método B.	99
Tabela 5.5 – Resultados obtidos pelo método C.	99
Tabela 5.6 – Resultados obtidos pelo método D.	100
Tabela 5.7 – Resultados obtidos pelo método E.	100
Tabela 5.8 – Resultados obtidos pelo método F utilizando-se <i>Rocket</i>	101
Tabela 5.9 – Resultados obtidos pelo método F utilizando-se <i>MiniRocket</i>	102
Tabela 5.10 – Resultados obtidos pelo método G utilizando-se <i>Rocket</i>	103
Tabela 5.11 – Resultados obtidos pelo método G utilizando-se <i>MiniRocket</i>	103
Tabela 5.12 – Comparação entre os resultados dos métodos B e D.	106
Tabela 5.13 – Comparação entre os resultados dos métodos C e E.	106
Tabela 5.14 – Comparação entre os resultados dos métodos F e D.	108
Tabela 5.15 – Comparação entre os tempos operacionais (para classificar uma única amostra) mínimo e máximo para os métodos F e D.	108
Tabela 5.16 – Comparação entre os resultados dos métodos G e E.	109

Tabela 5.17 – Comparação entre os tempos operacionais (para classificar uma única amostra) mínimo e máximo para os métodos G e E.	109
Tabela 1 – Matriz de confusão do método B com 1 ciclo pós falta.	115
Tabela 2 – Matriz de confusão do método B com 1/2 ciclo pós falta.	117
Tabela 3 – Matriz de confusão do método B com 1/4 ciclo pós falta.	118
Tabela 4 – Matriz de confusão do método B com 1/8 ciclo pós falta.	120
Tabela 5 – Matriz de confusão do método B com 1/16 ciclo pós falta.	121
Tabela 6 – Matriz de confusão do método B com 1/32 ciclo pós falta.	123
Tabela 7 – Matriz de confusão do método B com 1/64 ciclo pós falta.	124
Tabela 8 – Matriz de confusão do método B com 1/128 ciclo pós falta.	126
Tabela 9 – Matriz de confusão do método C com 1 ciclo pós falta.	127
Tabela 10 – Matriz de confusão do método C com 1/2 ciclo pós falta.	129
Tabela 11 – Matriz de confusão do método C com 1/4 ciclo pós falta.	130
Tabela 12 – Matriz de confusão do método C com 1/8 ciclo pós falta.	132
Tabela 13 – Matriz de confusão do método C com 1/16 ciclo pós falta.	133
Tabela 14 – Matriz de confusão do método C com 1/32 ciclo pós falta.	135
Tabela 15 – Matriz de confusão do método C com 1/64 ciclo pós falta.	136
Tabela 16 – Matriz de confusão do método C com 1/128 ciclo pós falta.	138
Tabela 17 – Matriz de confusão do método D com 1 ciclo pós falta.	139
Tabela 18 – Matriz de confusão do método D com 1/2 ciclo pós falta.	141
Tabela 19 – Matriz de confusão do método D com 1/4 ciclo pós falta.	142
Tabela 20 – Matriz de confusão do método D com 1/8 ciclo pós falta.	144
Tabela 21 – Matriz de confusão do método D com 1/16 ciclo pós falta.	145
Tabela 22 – Matriz de confusão do método D com 1/32 ciclo pós falta.	147
Tabela 23 – Matriz de confusão do método D com 1/64 ciclo pós falta.	148
Tabela 24 – Matriz de confusão do método D com 1/128 ciclo pós falta.	150
Tabela 25 – Matriz de confusão do método E com 1 ciclo pós falta.	151
Tabela 26 – Matriz de confusão do método E com 1/2 ciclo pós falta.	153
Tabela 27 – Matriz de confusão do método E com 1/4 ciclo pós falta.	154
Tabela 28 – Matriz de confusão do método E com 1/8 ciclo pós falta.	156
Tabela 29 – Matriz de confusão do método E com 1/16 ciclo pós falta.	157
Tabela 30 – Matriz de confusão do método E com 1/32 ciclo pós falta.	159

Tabela 31 – Matriz de confusão do método E com 1/64 ciclo pós falta.	160
Tabela 32 – Matriz de confusão do método E com 1/128 ciclo pós falta.	162
Tabela 33 – Matriz de confusão do método F com 1 ciclo pós falta e extração de características com <i>Rocket</i>	164
Tabela 34 – Matriz de confusão do método F com 1/2 ciclo pós falta e extração de características com <i>Rocket</i>	165
Tabela 35 – Matriz de confusão do método F com 1/4 ciclo pós falta e extração de características com <i>Rocket</i>	167
Tabela 36 – Matriz de confusão do método F com 1/8 ciclo pós falta e extração de características com <i>Rocket</i>	168
Tabela 37 – Matriz de confusão do método F com 1/16 ciclo pós falta e extração de características com <i>Rocket</i>	170
Tabela 38 – Matriz de confusão do método F com 1/32 ciclo pós falta e extração de características com <i>Rocket</i>	171
Tabela 39 – Matriz de confusão do método F com 1/64 ciclo pós falta e extração de características com <i>Rocket</i>	173
Tabela 40 – Matriz de confusão do método F com 1/128 ciclo pós falta e extração de características com <i>Rocket</i>	174
Tabela 41 – Matriz de confusão do método F com 1 ciclo pós falta e extração de características com <i>MiniRocket</i>	176
Tabela 42 – Matriz de confusão do método F com 1/2 ciclo pós falta e extração de características com <i>MiniRocket</i>	178
Tabela 43 – Matriz de confusão do método F com 1/4 ciclo pós falta e extração de características com <i>MiniRocket</i>	179
Tabela 44 – Matriz de confusão do método F com 1/8 ciclo pós falta e extração de características com <i>MiniRocket</i>	181
Tabela 45 – Matriz de confusão do método F com 1/16 ciclo pós falta e extração de características com <i>MiniRocket</i>	182
Tabela 46 – Matriz de confusão do método F com 1/32 ciclo pós falta e extração de características com <i>MiniRocket</i>	184
Tabela 47 – Matriz de confusão do método F com 1/64 ciclo pós falta e extração de características com <i>MiniRocket</i>	185

Tabela 48 – Matriz de confusão do método F com 1/128 ciclo pós falta e extração de características com <i>MiniRocket</i>	187
Tabela 49 – Matriz de confusão do método G com 1 ciclo pós falta e extração de características com <i>Rocket</i>	189
Tabela 50 – Matriz de confusão do método G com 1/2 ciclo pós falta e extração de características com <i>Rocket</i>	190
Tabela 51 – Matriz de confusão do método G com 1/4 ciclo pós falta e extração de características com <i>Rocket</i>	192
Tabela 52 – Matriz de confusão do método G com 1/8 ciclo pós falta e extração de características com <i>Rocket</i>	193
Tabela 53 – Matriz de confusão do método G com 1/16 ciclo pós falta e extração de características com <i>Rocket</i>	195
Tabela 54 – Matriz de confusão do método G com 1/32 ciclo pós falta e extração de características com <i>Rocket</i>	196
Tabela 55 – Matriz de confusão do método G com 1/64 ciclo pós falta e extração de características com <i>Rocket</i>	198
Tabela 56 – Matriz de confusão do método G com 1/128 ciclo pós falta e extração de características com <i>Rocket</i>	199
Tabela 57 – Matriz de confusão do método G com 1 ciclo pós falta e extração de características com <i>MiniRocket</i>	201
Tabela 58 – Matriz de confusão do método G com 1/2 ciclo pós falta e extração de características com <i>MiniRocket</i>	203
Tabela 59 – Matriz de confusão do método G com 1/4 ciclo pós falta e extração de características com <i>MiniRocket</i>	204
Tabela 60 – Matriz de confusão do método G com 1/8 ciclo pós falta e extração de características com <i>MiniRocket</i>	206
Tabela 61 – Matriz de confusão do método G com 1/16 ciclo pós falta e extração de características com <i>MiniRocket</i>	207
Tabela 62 – Matriz de confusão do método G com 1/32 ciclo pós falta e extração de características com <i>MiniRocket</i>	209
Tabela 63 – Matriz de confusão do método G com 1/64 ciclo pós falta e extração de características com <i>MiniRocket</i>	210

Tabela 64 – Matriz de confusão do método G com 1/128 ciclo pós falta e extração de características com <i>MiniRocket</i>	212
Tabela 65 – Resultados obtidos usando <i>Rocket</i> no método F variando-se a quantidade de <i>kernels</i>	213
Tabela 66 – Resultados obtidos usando <i>MiniRocket</i> no método F variando-se a quantidade de <i>features</i>	216
Tabela 67 – Resultados obtidos usando <i>Rocket</i> no método G variando-se a quantidade de <i>kernels</i>	219
Tabela 68 – Resultados obtidos usando <i>MiniRocket</i> no método G variando-se a quantidade de <i>features</i>	222

LISTA DE QUADROS

Quadro 3.1 – Matriz de confusão para 4 classes	65
Quadro 3.2 – Matriz de confusão para 2 classes	65

LISTA DE SIGLAS

ANN	<i>Artificial neural network</i>
CBA	Congresso Brasileiro de Automática
CFBPN	<i>Cascaded correlation feed forward network</i>
CNN	<i>Convolutional neural network</i>
DFT	<i>Discrete Fourier transform</i>
DWT	<i>Discrete wavelet transform</i>
FIS	<i>Fuzzy Inference Systems</i>
FT	<i>Fourier transform</i>
GPS	<i>Global Positioning System</i>
HOS	<i>Higher-order statistics</i>
ICA	<i>Independet component analysis</i>
IIR	<i>Infinite impulse response</i>
JCAE	<i>Journal of Control, Automation and Electrical Systems</i>
LDA	<i>Linear discriminant analysis</i>
MDS	<i>Multidimensional scaling</i>
MLP	<i>Multilayer perceptron</i>
MODWT	<i>Maximal overlap discrete wavelet transform</i>
ONS	Operador Nacional do Sistema Elétrico
pdf	<i>Probability density function</i>
ppv	<i>Proportion of positive values</i>
RBF	<i>Radial basis function</i>
RMS	<i>Root mean square</i>
RNA	Redes neurais artificiais

RTDS	<i>Real time digital simulator</i>
SEP	Sistema elétrico de potência
SIN	Sistema Interligado Nacional
SNR	<i>Signal-to-noise ratio</i>
t-SNE	<i>t-Distributed Stochastic Neighbor Embedding</i>
TW	<i>Traveling Wave</i>
WT	<i>Wavelet transform</i>

LISTA DE SÍMBOLOS

C	resultado da operação de convolução
H	entropia
Hz	Hertz, unidade de frequência
W	matriz de pesos
X	matriz de entrada
Y	admitância
Z	impedância
\log	logaritmo
$\mathcal{N}(0, 1)$	distribuição normal com média 0 e desvio padrão 1
$\mathcal{U}(-1, 1)$	distribuição uniforme entre -1 e 1
b	bias
d	dilatação
exp	exponencial
f_s	frequência de amostragem
k	<i>kernel</i>
l	comprimento
rad	radianos
y	vetor de saída

LISTA DE SÍMBOLOS GREGOS

γ	parâmetro de escala
ω	frequência angular
ϕ	função de ativação
π	constante com valor igual a 3,14159...
ρ_0	fator notch
Σ	símbolo de somatório
σ	desvio padrão
σ^2	variância

SUMÁRIO

1	INTRODUÇÃO	24
1.1	Objetivos	29
1.2	Organização do texto	29
2	ESTADO DA ARTE	30
3	REFERENCIAL TEÓRICO	42
3.1	Filtro <i>Notch</i>	42
3.2	<i>Rocket</i>	43
3.3	<i>MiniRocket</i>	46
3.4	Classificadores usados com <i>MiniRocket</i> e <i>Rocket</i>	49
3.4.1	Regressão <i>ridge</i>	50
3.5	Florestas Aleatórias	51
3.5.1	Árvores de Decisão	52
3.5.1.1	Ganho de Informação	54
3.5.1.2	Taxa de ganho	54
3.5.1.3	Índice de Gini	55
3.5.2	<i>Bagging</i>	55
3.6	Redes neurais artificiais	57
3.7	Validação Cruzada	61
3.7.1	Validação <i>k-fold</i>	63
3.8	Matriz de Confusão	64
3.9	t-Distributed Stochastic Neighbor Embedding	66
4	MATERIAIS E MÉTODOS	72
4.1	Bases de dados	72
4.1.1	Primeiro conjunto de dados	72
4.1.2	Segundo conjunto de dados	75
4.2	Ferramentas utilizadas	79
4.3	Metodologia	80
4.3.1	Avaliação dos modelos	80
4.3.2	Métodos treinados usando sinais de tensão	83
4.3.2.1	Método A	83
4.3.2.2	Método B	86

4.3.2.3	Método C	87
4.3.3	Métodos treinados usando sinais de corrente	88
4.3.3.1	Método D	89
4.3.3.2	Método E	89
4.3.3.3	Método F	89
4.3.3.4	Método G	91
5	RESULTADOS E DISCUSSÕES	92
5.1	Comparação entre os bancos de dados	92
5.2	Métodos implementados	96
5.3	Método A	97
5.4	Método B	99
5.5	Método C	99
5.6	Método D	100
5.7	Método E	100
5.8	Método F	101
5.8.1	<i>Rocket</i>	101
5.8.2	<i>MiniRocket</i>	101
5.9	Método G	102
5.9.1	<i>Rocket</i>	102
5.9.2	<i>MiniRocket</i>	103
5.10	Discussões	103
6	CONCLUSÃO	110
6.1	Propostas de trabalhos futuros	111
6.2	Publicações	112
7	EQUIPE TÉCNICA	113
	APENDICE A – Resultados dos métodos implementado	114
A.1	Método B	114
A.1.1	1 Ciclo Pós Falta	114
A.1.2	1/2 Ciclo Pós Falta	115
A.1.3	1/4 Ciclo Pós Falta	117
A.1.4	1/8 Ciclo Pós Falta	119
A.1.5	1/16 Ciclo Pós Falta	120

A.1.6	1/32 Ciclo Pós Falta	122
A.1.7	1/64 Ciclo Pós Falta	123
A.1.8	1/128 Ciclo Pós Falta	125
A.2	Método C	126
A.2.1	1 Ciclo Pós Falta	126
A.2.2	1/2 Ciclo Pós Falta	128
A.2.3	1/4 Ciclo Pós Falta	129
A.2.4	1/8 Ciclo Pós Falta	130
A.2.5	1/16 Ciclo Pós Falta	132
A.2.6	1/32 Ciclo Pós Falta	134
A.2.7	1/64 Ciclo Pós Falta	135
A.2.8	1/128 Ciclo Pós Falta	136
A.3	Método D	138
A.3.1	1 Ciclo Pós Falta	138
A.3.2	1/2 Ciclo Pós Falta	140
A.3.3	1/4 Ciclo Pós Falta	141
A.3.4	1/8 Ciclo Pós Falta	142
A.3.5	1/16 Ciclo Pós Falta	144
A.3.6	1/32 Ciclo Pós Falta	145
A.3.7	1/64 Ciclo Pós Falta	147
A.3.8	1/128 Ciclo Pós Falta	148
A.4	Método E	150
A.4.1	1 Ciclo Pós Falta	150
A.4.2	1/2 Ciclo Pós Falta	152
A.4.3	1/4 Ciclo Pós Falta	153
A.4.4	1/8 Ciclo Pós Falta	154
A.4.5	1/16 Ciclo Pós Falta	156
A.4.6	1/32 Ciclo Pós Falta	157
A.4.7	1/64 Ciclo Pós Falta	159
A.4.8	1/128 Ciclo Pós Falta	160
A.5	Método F	162
A.5.1	<i>Rocket</i>	162

A.5.1.1	1 Ciclo Pós Falta	162
A.5.1.2	1/2 Ciclo Pós Falta	164
A.5.1.3	1/4 Ciclo Pós Falta	166
A.5.1.4	1/8 Ciclo Pós Falta	167
A.5.1.5	1/16 Ciclo Pós Falta	169
A.5.1.6	1/32 Ciclo Pós Falta	170
A.5.1.7	1/64 Ciclo Pós Falta	172
A.5.1.8	1/128 Ciclo Pós Falta	173
A.5.2	<i>MiniRocket</i>	175
A.5.2.1	1 Ciclo Pós Falta	175
A.5.2.2	1/2 Ciclo Pós Falta	176
A.5.2.3	1/4 Ciclo Pós Falta	178
A.5.2.4	1/8 Ciclo Pós Falta	180
A.5.2.5	1/16 Ciclo Pós Falta	181
A.5.2.6	1/32 Ciclo Pós Falta	183
A.5.2.7	1/64 Ciclo Pós Falta	184
A.5.2.8	1/128 Ciclo Pós Falta	186
A.6	Método G	187
A.6.1	<i>Rocket</i>	187
A.6.1.1	1 Ciclo Pós Falta	188
A.6.1.2	1/2 Ciclo Pós Falta	189
A.6.1.3	1/4 Ciclo Pós Falta	191
A.6.1.4	1/8 Ciclo Pós Falta	192
A.6.1.5	1/16 Ciclo Pós Falta	194
A.6.1.6	1/32 Ciclo Pós Falta	195
A.6.1.7	1/64 Ciclo Pós Falta	197
A.6.1.8	1/128 Ciclo Pós Falta	198
A.6.2	<i>MiniRocket</i>	200
A.6.2.1	1 Ciclo Pós Falta	200
A.6.2.2	1/2 Ciclo Pós Falta	201
A.6.2.3	1/4 Ciclo Pós Falta	203
A.6.2.4	1/8 Ciclo Pós Falta	205

A.6.2.5	1/16 Ciclo Pós Falta	206
A.6.2.6	1/32 Ciclo Pós Falta	208
A.6.2.7	1/64 Ciclo Pós Falta	209
A.6.2.8	1/128 Ciclo Pós Falta	211
	APENDICE B – Tabelas de resultados	213
	REFERÊNCIAS	225

1 INTRODUÇÃO

Todo sistema elétrico de potência (SEP) está sujeito a faltas. A construção e utilização de um sistema completamente imune à faltas exigiria um alto investimento cujo o custo ao ser repassado ao consumidor inviabilizaria o seu uso. Por esse motivo, o Brasil, assim como outros países, segue o critério de confiabilidade n-1 no planejamento dos sistemas elétricos, segundo o qual, eles devem ser capazes de suportar a perda de qualquer elemento sem a interrupção do fornecimento. Ou seja, mesmo na ocorrência de uma contingência simples o sistema deve continuar em operação sem interromper o fornecimento de energia, perder a estabilidade, violar os padrões de frequência ou tensão e sobrecarregar equipamentos e instalações (ONS, 2019b).

Segundo o Operador Nacional do Sistema Elétrico (ONS, 2019b), diversas perturbações ocorrem nos sistemas elétricos durante a operação em tempo real, muitas delas não provocam medidas graves como o corte de carga e sequer são percebidas pelo consumidor. Por sua vez, aproximadamente 10% das perturbações interrompem o fornecimento de energia. Inúmeras causas podem levar a tais interrupções: defeitos de equipamentos, eventos meteorológicos (tempestades, ventos, chuvas etc), incêndios, faltas dos sistemas protetivos, erros humanos e operacionais, ação de animais, dentre diversas outras causas.

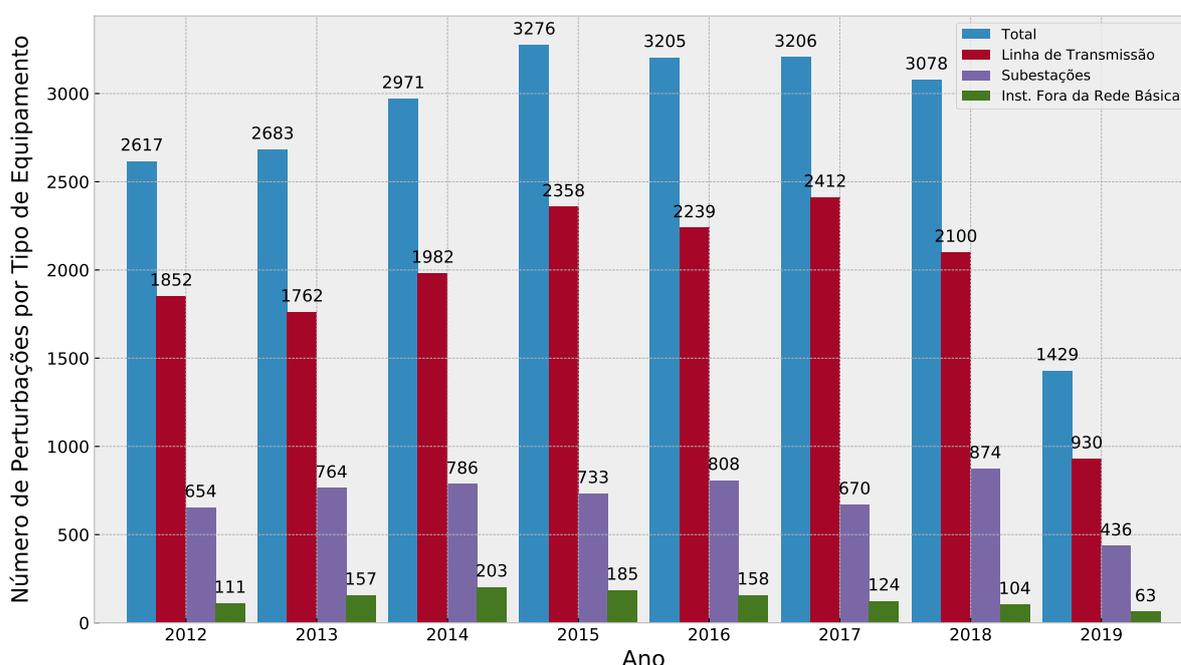
Para assegurar aos usuários do Sistema Interligado Nacional (SIN) a segurança, a continuidade, a qualidade e a economicidade do suprimento de energia elétrica por meio das instalações de transmissão integrantes do SIN com nível de tensão de operação igual ou superior a 230kV, o ONS estabeleceu alguns indicadores de desempenho como a robustez da rede básica, a duração equivalente de interrupção de carga, a frequência equivalente de interrupção de carga e a energia não suprida (ONS, 2019a).

Nos sistemas elétrico de potência, um dos componentes mais susceptíveis a faltas é a linha de transmissão. Uma vez que ela, em geral, está exposta ao ambiente externo os fatores anteriormente citados podem interferir no seu perfeito funcionamento fazendo que uma falta ocorra e, por conseguinte, o fornecimento de energia seja interrompido. É importante destacar também que há um crescimento da extensão das linhas de transmissão nos últimos anos devido ao acréscimo da demanda de energia elétrica e a necessidade de aumentar os locais que a recebem. Segundo ONS (2017) é possível verificar anualmente a crescente demanda para ampliação da extensão das linhas de transmissão até o ano de 2023.

Esse acréscimo de linhas de transmissão no país demonstra a necessidade de se implementar sistemas protetivos que minimizem a ocorrência de faltas e interrupções na transmissão

de energia elétrica. No gráfico apresentado na Figura 1.1 é possível verificar que a maioria das ocorrências de interrupções e faltas nos sistemas elétricos de potência ocorrem nas linhas de transmissão. Nele são apresentadas informações do ano de 2012 até maio de 2019. As linhas de transmissão (representadas pela barra vermelha do gráfico) são responsáveis, em média, por 70% da origem das perturbações envolvendo a Rede Básica, o que se justifica por sua maior exposição em relação aos demais componentes do sistema elétrico (ONS, 2019c). Esse fato ratifica a necessidade de se desenvolver sistemas eficientes capazes de proteger esse componente que é altamente vulnerável, visando evitar ou minimizar os prejuízos da concessionária e do consumidor final.

Figura 1.1 – Número de Perturbações por Tipo de Equipamento Origem.



Fonte: Adaptado de ONS (2019c).

A interrupção do fornecimento de energia elétrica acarreta em inúmeros prejuízos para o consumidor e fornecedor. Diversas causas podem levar à sua ocorrência, dentre elas pode-se citar as condições climáticas como tempestades, descargas elétricas, nevascas, além disso, certas regiões são afetadas por incêndios constantes que também podem levar à interrupção no fornecimento de energia. Tais interrupções têm efeitos extremamente negativos podendo levar ao desligamento de sistemas de refrigeração de alimentos e remédios, das redes de comunicação, além da interrupção no funcionamento de bombas de água que abastecem edifícios, escolas e residências.

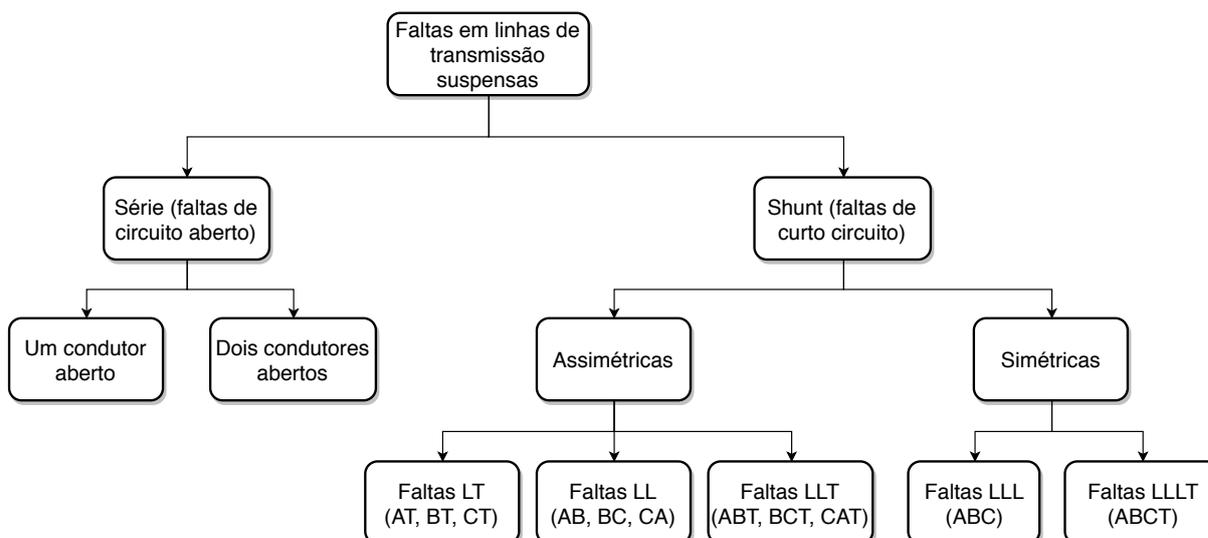
Outros locais no mundo estão sujeitos a condições climáticas e eventos naturais distintos em relação ao Brasil, mas que também podem levar à interrupção de energia elétrica. Em fevereiro de 2019, por exemplo, nove torres de transmissão de energia localizadas entre as cidades de Batote e Kishtwar na Índia foram severamente danificadas após uma nevasca. Isso levou à interrupção do fornecimento de energia elétrica durante dias nas cidades (TANTRAY, 2019).

Em geral, torres e linhas de transmissão antigas e que não passam por um processo de manutenção constante possuem um risco maior de sofrerem danos durante eventos naturais. Porto Rico, por exemplo, sofre com constantes interrupções no fornecimento de energia elétrica seja pela ineficiência das empresas responsáveis ou por eventos naturais como furacões e terremotos. Em janeiro de 2020, o maior terremoto do século atingiu o país danificando linhas de transmissão e interrompendo o fornecimento de energia por alguns dias para uma parte da população (MAZZEI; PENN; ROBLES, 2020).

As situações citadas corroboram a necessidade da criação de sistemas de proteção eficientes para que danos graves sejam evitados ou minimizados. Segundo Carvalho et al. (2014), as etapas de um sistema protetivo contra faltas em linhas de transmissão podem ser basicamente divididas em três partes distintas. A primeira delas é a detecção da falta que corresponde à identificação da ocorrência da falta e consequente amostragem do sinal. Em seguida é realizada a classificação da falta cuja função é identificar que tipo de falta ocorreu, ou seja, quais fases estão presentes na falta e se existe ou não presença do terra. Por fim, é executada a localização da falta que baseia-se em estimar a distância entre o ponto de ocorrência da falta e o local do relé.

Existem dois grupos de faltas que podem ocorrer nas linhas de transmissão: as de circuito aberto (ou faltas do tipo série) e as de curto-circuito (também chamadas de faltas *shunt*). As faltas de circuito aberto podem ser identificadas observando a tensão em cada fase, um aumento nos valores de tensão pode ser um indicativo de que ela ocorreu, apesar desse comportamento ocorrer em outros eventos elétricos. Elas podem ser classificadas em dois tipos: um circuito aberto e dois circuitos abertos. Essas faltas ocorrem raramente e são pouco abordadas na literatura. Por sua vez, as faltas de curto circuito podem ser identificadas de duas formas: pela diminuição no nível de tensão na fase ou elevação dos valores de corrente. Elas podem ser divididas em dois tipos: assimétricas e simétricas, conforme a Figura 1.2 (PRASAD; EDWARD; RAVI, 2017a).

Figura 1.2 – Tipos de faltas em linhas de transmissão.



Fonte: Adaptado de Prasad, Edward e Ravi (2017a).

As faltas mais comuns são aquelas que envolvem apenas uma fase e o terra (AT, BT, CT). Faltas bifásicas (AB, BC e CA) são um pouco mais raras e severas. Já as faltas que envolvem duas fases e o terra (ABT, BCT e CAT) são ainda menos comuns e mais nocivas. Por fim, faltas trifásicas (ABC) e trifásicas-terra (ABCT) quando ocorrem podem levar o sistema inteiro ao colapso, mas também possuem uma frequência de ocorrência menor (PRASAD; EDWARD; RAVI, 2017a). A partir de observações práticas verificou-se que entre 70% e 80% das faltas em linha de transmissão envolvem apenas uma fase (STEVENSON, 1978). Por sua vez, faltas nas três fases ocorrem em apenas 5% dos casos.

Ao realizar uma pesquisa sobre trabalhos desenvolvidos na área, verificou-se que diversos pesquisadores já propuseram a utilização de técnicas variadas em sistemas de proteção de faltas em linhas de transmissão. Pelos trabalhos analisados, na área de classificação de faltas são utilizados em sua maioria técnicas de processamento de sinais, que podem ser utilizadas isoladamente, ou em conjunto com métodos de reconhecimento de padrões e inteligência computacional (CARVALHO et al., 2014; KUMAR; SWAMY.R; VENKATESH, 2014; MAHESHWARI; AGARWAL; SHARMA, 2019; SARAVANAN; RATHINAM, 2012; CHEN; FINK; SANSAVINI, 2018; ALMEIDA et al., 2017).

Dentre as técnicas mais utilizadas para pré-processamento de sinais estão a transformada de Fourier e *wavelet*. Enquanto a primeira fornece informações sobre todas as frequências presentes no sinal ela não fornece nenhuma indicação de quando esses sinais estavam presentes no tempo. Por sua vez, a transformada *wavelet* fornece uma análise multi-resolução na frequên-

cia e no tempo (DAS; SINGH; SINHA, 2005). No entanto, sabe-se que os seus parâmetros são muito sensíveis ao ruído, que é gerado devido a muitos fenômenos relacionados à linha de transmissão, isto é, campos eletromagnéticos, correntes de fuga em isoladores e condições ambientais (ALMEIDA et al., 2017).

Diversas técnicas podem ser usadas em processamento estatístico de sinais visando extrair informações dos mesmos ou torná-los mais apropriados para uma aplicação específica. É comum encontrar a aplicação da transformada de Fourier e *wavelet* em problemas envolvendo o processamento de sinais. Mas além delas, outras técnicas já foram utilizadas ao se abordar o problema de faltas em linhas de transmissão, cabe destacar o uso de duas: estatísticas de ordem superior e análise de componentes independentes (CARVALHO et al., 2014; ALMEIDA et al., 2017).

Estatística de ordem superior ou (*higher-order statistics* - HOS) é definida por meio de cumulantes e momentos e permite extrair maiores informações dos dados. Por sua vez, a análise de componentes independentes (*independent component analysis* - ICA) visa separar os sinais em componentes que são estatisticamente independentes tendo-se apenas a informação dos sinais misturados.

Apesar dessas técnicas já serem utilizadas em trabalhos envolvendo processamento de sinais, e até certo nível na área de sinais elétricos, com o advento da aprendizagem profunda (*deep learning*) outras técnicas eficientes surgiram para lidar com esse problema. Essa técnica geralmente é mais vantajosa por não ser necessário conhecer profundamente a origem e comportamento dos dados. Ela visa extrair características do conjunto de dados independente da forma, distribuição, estatísticas ou como ele se comporta. Isso pode ser importante e valioso dependendo da aplicação.

Essa dissertação aborda o uso de duas técnicas chamadas *Rocket* e *MiniRocket* que fazem uso de conceitos de redes neurais convolucionais, mas com menor complexidade computacional, e permitem realizar a transformação de séries temporais independente da sua origem e da sua fonte. Essas técnicas transformam os sinais em *features* de forma rápida e simples sem necessidade de se ter um conhecimento prévio sobre o sinal. Após a transformação, os dados podem ser usados em conjunto com um classificador linear para realizar a classificação das amostras.

1.1 Objetivos

O objetivo geral dessa pesquisa é realizar um estudo comparativo utilizando técnicas clássicas e modernas no processamento de sinais e classificação de faltas em linhas de transmissão. Dessa forma, utilizou-se filtro *notch* e *random forest* que são amplamente utilizados na área de processamento de sinais e classificação. Além disso, *Rocket* e *MiniRocket* que são técnicas recentes de extração de características de séries temporais foram utilizadas junto com um classificador de regressão *ridge* para efeito comparativo. Os objetivos específicos estão listados a seguir:

- a) analisar e processar os sinais disponíveis utilizando filtro *notch*, *MiniRocket* e *Rocket*;
- b) verificar o impacto da informação do sinal pós-falta na classificação das faltas;
- c) investigar se ocorre diferença no resultado final a partir da variação das técnicas utilizadas no pré-processamento;
- d) avaliar o desempenho da *random forest* e regressão *ridge* na identificação de faltas em linhas de transmissão;

Como contribuição desse trabalho destaca-se o uso de métodos de aprendizado profundo para tratamento e classificação de séries temporais. Esses métodos quando devidamente usados permitem uma classificação rápida do sinal sem que seja necessário ter conhecimento a priori sobre sua fonte, forma, características ou estatísticas.

1.2 Organização do texto

Para facilitar a leitura e o entendimento do texto como um todo, essa dissertação foi subdividida em capítulos. No Capítulo 2 são apresentados variados trabalhos que usaram técnicas diversas no campo de estudo desse projeto. Nele são abordadas, em sua maioria, trabalhos recentes na área mostrando o que os pesquisadores têm estudado e como esse assunto tem sido abordado. No Capítulo 3 são apresentadas algumas teorias, algoritmos e técnicas que foram importantes para a realização desse trabalho. Os materiais, métodos e metodologias utilizados na realização desse trabalho podem ser encontrados no capítulo 4. Por sua vez, no Capítulo 5 são mostrados alguns resultados obtidos durante essa pesquisa. No Capítulo 6 são apresentadas as conclusões e pontos de desenvolvimento para trabalhos futuros. E no Capítulo 7 é apresentada a equipe que contribuiu no desenvolvimento do projeto.

2 ESTADO DA ARTE

Nesse capítulo é apresentada uma revisão bibliográfica mostrando as diversas técnicas e metodologias utilizadas na detecção, classificação e localização de faltas em linhas de transmissão. Esse problema já foi tratado por diversos autores e abordagens distintas foram utilizadas para sua solução.

Nos últimos anos, técnicas de inteligência computacional têm sido utilizadas nessa e outras áreas, devido a capacidade de generalização, modelagem de fenômenos desconhecidos e imprecisos, exploração de sistemas complexos e difíceis de serem modelados etc. Dentre essas técnicas, cabe destacar a rede neural artificial (*artificial neural network* - ANN) (HAYKIN, 2001), que é um método, no qual a partir de um conjunto de entradas e saídas desejadas o sistema ajusta determinados parâmetros chamados de pesos e *bias*. O objetivo é que o modelo seja posteriormente capaz de generalizar e encontrar as saídas correspondentes para entradas desconhecidas, mas que possuam a mesma natureza daquelas utilizadas durante o treinamento.

Em geral, as pesquisas que envolvem detecção, classificação ou localização da falta possuem anteriormente uma etapa de pré-processamento. Existem inúmeras maneiras de fazer isso, mas a transformada de Fourier (*Fourier transform* - FT) e *wavelet* (*wavelet transform* - WT), são as técnicas mais utilizadas nessa situação. Alguns estudos comparativos entre o uso dessas técnicas já foram realizados com a finalidade de verificar aquela que apresenta os melhores resultados para o conjunto de dados e modelos utilizados. Abdollahi e Seyedtabaii (2010), por exemplo, compararam o uso dessas duas técnicas supracitadas como etapa de pré-processamento combinadas com redes neurais artificiais para estimar a localização de faltas em linhas de transmissão. Os algoritmos propostos são baseados na análise da transformada de Fourier da frequência fundamental dos sinais de corrente e tensão na ocorrência de um curto-circuito na linha. Uma análise similar também é realizada nos sinais transientes de corrente e tensão utilizando a transformada *wavelet*. Os pesquisadores inicialmente utilizaram o algoritmo com FT para detecção e classificação de faltas como etapa preliminar à estimativa da localização da falta. O modelo de redes neurais proposto opera usando dois grupos independentes de entradas, o tipo da falta é fornecido pelo classificador, enquanto as outras entradas são fornecidas pelo cálculo da transformada discreta de Fourier (*discrete Fourier transform* - DFT) das correntes e tensões das três fases. As quantidades de amplitudes de DFT em frequência fundamental de acordo com o tipo de falta e os locais são usadas para obter entradas. Esses sinais são então utilizados na rede neural que irá fornecer uma estimativa para o local no qual a falta

ocorreu. O segundo modelo implementado faz uso da transformada discreta *wavelet* (*discrete wavelet transform* - DWT) para pré-processamento dos sinais. Nesse caso, o localizador também irá ter dois grupos de entradas. O primeiro é o tipo da falta obtido pelo classificador e o segundo é composto pelos coeficientes de detalhe no nível três para baixa frequência obtidos pela DWT. Esses dados são usados em uma rede neural que por sua vez realiza uma estimativa para a localização da falta. Como conclusão para o conjunto de dados utilizado, a DWT teve melhores resultados no caso de faltas monofásicas com terra, enquanto a DFT apresentou melhores resultados para outros tipos de faltas, sendo que o erro para ambos modelos ficou abaixo de 6% em todas as situações testadas (resistência de 50 Ω e 200 Ω , distância de 0 a 200 km, faltas monofásicas-terra, bifásicas, bifásicas-terra, trifásicas e trifásicas-terra).

Alves Da Silva, Lima e Souza (2012) fizeram uso de redes neurais de domínio complexo para mapear as relações entre sinais elétricos e a localização das faltas. Esse método permite representar a tensão ou corrente sem desacoplar arbitrariamente a amplitude e a fase. Dessa maneira, foi mostrado como a localização de faltas pode ser melhorada se a combinação de processamento de sinal e seleção de modelo também considerarem o domínio apropriado para as variáveis de entrada. Eles compararam o desempenho entre redes neurais de domínio real e domínio complexo considerando várias representações do espaço de entrada. As redes neurais de domínio complexo são extensões das redes *feedforward* de domínio real. A diferença é que elas usam como variáveis de entrada e saída valores complexos, além de seus pesos e *biases* também serem números complexos e conseqüentemente a função de ativação ter o seu domínio e contra-domínio definido no eixo complexo. Nesse trabalho foram utilizados dois conjuntos de dados, o primeiro é baseado no estado estacionário e como entrada foram utilizados os fasores das tensões e correntes pré e pós falta, totalizando 12 entradas. O segundo, armazena sinais transientes eletromagnéticos e passa por um série de processamentos que incluem *downsampling* e o uso de filtro Butterworth. A partir desses sinais transientes processados são criados dois novos conjuntos de dados um a partir do uso da Transformada Discreta de Fourier e o outro com a Transformada *wavelet* estacionária. Assim, foram criados 6 modelos distintos e utilizou-se seis redes neurais com parâmetros (número de entradas, camadas escondidas, neurônios) diferentes para efeito de comparação.

Saravanan e Rathinam (2012) apresentaram um trabalho que também utiliza redes neurais, no entanto, eles não abordaram a detecção e se limitaram somente à classificação e localização de faltas. Além disso, o projeto realizado por eles leva em consideração linhas de trans-

missão de circuito duplo. Essa é uma configuração na qual as linhas são dispostas em paralelo e geralmente apresenta uma complexidade maior na modelagem de sistemas de proteção tendo em vista que ocorre a interferência de um circuito em outro. Nesse trabalho foram utilizadas três variações de redes neurais artificiais (RNA) *feedforward*: com algoritmo *backpropagation*, função de base radial (*radial basis function* - RBF) e rede de cascata correlacionada (*cascaded correlation feed forward network* - CFBPN). O método proposto utiliza os valores das correntes de linha para aprender relações escondidas nos padrões de entrada. Algoritmos baseados em redes neurais tem excelentes características como capacidade de generalização, imunidade a ruídos e robustez. Um aumento na magnitude da corrente ou diminuição na tensão podem ser consideradas como métricas para detectar uma falta. Além disso, quando uma falta ocorre a linha de transmissão passa por um período transiente, e pode não ser fácil determinar a corrente e tensão após a ocorrência da falta. Três arquiteturas de redes neurais *feedforward* foram usadas para a resolução desse problema. E concluiu-se que a rede RBF leva menos tempo para fornecer o resultado, mas a CFBPN fornece um erro médio quadrático menor quando comparada com as demais arquiteturas usadas no trabalho.

Algumas pesquisas já foram realizadas visando detectar, classificar e localizar faltas em tempo real. Costa, Souza e Brito (2012), por exemplo, usaram as energias do coeficiente *wavelet* da Transformada *Wavelet* de Sobreposição Máxima (*Maximal Overlap Discrete Wavelet Transform* - MODWT) em relação aos transientes induzidos por falta para a classificação de faltas em tempo real nas linhas de transmissão. A MODWT é uma variação da DWT e usa filtros passa baixa e passa alta para dividir a banda de frequência do sinal de entrada em componentes de baixa e alta frequência em várias escalas. No entanto, ao contrário da DWT, a MODWT não realiza o processo de *downsampling*. Dessa forma, os coeficientes *wavelet* podem ser calculados logo após cada processo de amostragem, essa é uma característica interessante para aplicações em tempo real. Nesse trabalho, foi utilizada a versão normalizada desses coeficientes (que variavam entre 0 e 1 para as fases e era maior ou igual a 0 para o terra) e fizeram uso de visualização gráfica para verificar a classificação das faltas. Assim, os valores dos coeficientes para cada fase geravam um plano triangular no espaço tridimensional, sendo que a cor era um indicativo da presença ou não do terra, verde quando o coeficiente do terra fosse maior ou igual a 0,1 ($G \geq 0,1$) indicando uma falta monofásica ou bifásica com terra, ou azul ($G < 0,1$) para indicar faltas sem presença de terra, ou seja, bifásicas ou trifásicas. A abordagem proposta nesse trabalho é composta por dois módulos. O primeiro tem a função de detectar distúrbios

em tempo real, assim a MODWT é aplicada nas tensões e correntes procedendo com o cálculo da energia do coeficiente *wavelet* na primeira escala. A detecção de distúrbio é realizada através da análise dessas energias. Esse método é capaz de detectar faltas, afundamento de tensão, transientes de comutação e outros tipos de distúrbio. O segundo módulo proposto, por sua vez, classifica a falta em tempo real quando ela é detectada no primeiro módulo. Somente as energias do coeficiente *wavelet* em relação às correntes de fase e neutro são usadas nesta etapa. Por fim, o desempenho do sistema proposto foi avaliado por meio do Simulador Digital em Tempo Real (*Real Time Digital Simulator - RTDS*), uma taxa de sucesso de 100% na classificação em tempo real foi obtida para faltas monofásicas com terra e bifásicas, apesar de algumas faltas bifásicas com terra e trifásicas terem sido classificadas erroneamente pela análise da energia do coeficiente *wavelet*, os coeficientes de aproximação podem ser usados para identificar esses tipos de faltas. A performance do método também foi avaliada de forma *off-line* em 59 registros oscilográficos reais de uma concessionária brasileira de sistemas de energia e somente duas faltas foram classificadas incorretamente.

Carvalho et al. (2014), propuseram o uso de estatística de ordem superior (*Higher-Order Statistics - HOS*), cumulantes e redes neurais artificiais para proteção de linhas de transmissão. Nesse caso, HOS é usada na extração de características no esquema de proteção. As ANNs recebem essas estatísticas como entradas e são responsáveis pelo funcionamento lógico do sistema decidindo se um desvio é necessário após detectar, classificar e localizar uma falta. O estágio de detecção se mostrou imune a alta presença de ruído aditivo e também ao desvio de frequência do sistema de potência. HOS possibilita a obtenção de maiores informações sobre o processo quando comparada com a média e variância. Assim, ela fornece uma maneira mais eficaz de diferenciar padrões em algumas aplicações. O sistema proposto faz uso também de um filtro *notch* usado para remover a frequência fundamental do sinal. Após a filtragem, a HOS é calculada fazendo uso dos cumulantes de segunda e quarta ordens, os sinais provenientes são então utilizados como entradas para as redes neurais. Cada subsistema possui, sua própria ANN do tipo MLP (*multilayer perceptron*). Como resultado, foi obtida acurácia de 100% para detecção, 99,15% para classificação e mostraram graficamente como a curva da localização predita pelo modelo se aproxima da curva real do local no qual as faltas ocorreram.

Kumar, Swamy.R e Venkatesh (2014) também exploraram o uso de redes neurais artificiais em sistemas de proteção para linhas de transmissão, eles propuseram uma configuração que utiliza o algoritmo *backpropagation* durante o treinamento da rede. Após detectar, clas-

sificar e localizar a falta, um relé envia um sinal de desvio para um *circuit breaker* a fim de desconectar (isolar) a linha (ou linhas) que apresenta (apresentam) defeito. Para isso, os autores implementaram um esquema completo para proteção à distância do sistema. Eles fizeram uso de duas ANNs uma foi usada para detectar e classificar a falta enquanto a outra foi utilizada para localizá-la em diferentes zonas. Nesse trabalho foram exploradas apenas as faltas monofásicas-terra (*single-phase to ground*), bifásicas (*double phase*) e bifásicas-terra (*double phase to ground*). Além disso, diferentemente de muitos trabalhos na área, os pesquisadores utilizaram como entrada os valores RMS (*root mean square*) da tensão e corrente. Assim, as arquiteturas de redes neurais utilizadas possuem seis entradas (tensões e correntes para três fases) e quatro saídas associadas com as quatro categorias de faltas. As saídas assumem valores binários correspondendo às três fases ou terra e pode ser generalizada para representar todas as permutações práticas de categorias de faltas que envolvem combinações de fases. Por sua vez, a localização é realizada a partir da divisão das linhas de transmissão em 3 zonas, e o modelo neural deve indicar em qual delas a falta ocorreu. Essa abordagem difere um pouco do que é encontrado na literatura, uma vez que, em geral os projetos nessa área visam estimar o local da ocorrência da falta na extensão total da linha.

Yadav e Swetapadma (2015) propuseram um esquema de proteção com transformada *wavelet* e análise de discriminante linear (*Linear Discriminant Analysis* - LDA) para detectar e classificar faltas em linhas de transmissão. Para isso, foram projetados 4 módulos que identificam a fase na qual a falta ocorreu e realiza a classificação. Foram utilizadas três correntes de linha e corrente de sequência zero, além disso, o número de amostras no treinamento corresponde a um quarto de ciclo após o pré-processamento com DWT. A LDA é usada para estudar a diferença entre dois ou mais grupos de objetos a respeito a várias variáveis simultaneamente, determinando se diferenças significativas existem entre eles e identificando o poder discriminativo de cada variável. No modelo proposto os valores das correntes de fase e sequência nula são obtidos e pré-processados com a DWT, e em seguida eles são testados na rede de detecção LDA treinada, que irá informar se teve ou não ocorrência de uma falta. Os coeficientes de aproximação são usados como entrada para cada um dos módulos LDA treinados que irão determinar qual fase está presente na falta e se houve ou não presença do terra. O modelo proposto apresentou excelentes resultados em todos os casos de falta que incluem testes em linhas de transmissão de circuito duplo e faltas que ocorrem simultaneamente em diferentes localizações da linha. A sua acurácia foi de 100% para a detecção e localização de faltas. Além disso, cabe

citar que os resultados da simulação mostraram que o esquema proposto é capaz de detectar e classificar faltas de até 99% da linha dentro de um tempo de ciclo, o que é uma vantagem do esquema proposto em relação a outros métodos utilizados.

Shaik e Pulipaka (2015) usaram a transformada *wavelet* para pré-processar os sinais de entrada em um sistema de detecção, classificação e localização de faltas em linhas de transmissão. Além disso, eles utilizaram um sistema de posicionamento global (*Global Positioning System* - GPS) para sincronizar a amostragem dos sinais de tensão e corrente nas duas extremidades da linha. Foi considerado meio ciclo do sinal para obtenção dos coeficientes de detalhe local que eram transmitidos por um cabo de fibra ótica até o terminal remoto, onde após algumas operações eram obtidos os índices de falta. Após isso, foi realizada uma comparação entre esses índices e os valores limiares pré determinados para detectar e classificar a falta. Em seguida, a estimativa do local de ocorrência da falta é feita utilizando-se ANNs. Duas redes foram empregadas nesse projeto, uma delas para localizar faltas que envolvem o terra enquanto a outra executa a mesma tarefa para faltas de fase. As decomposições A3 obtidas em um ciclo completo foram usadas como entradas para essas redes. O vetor de entrada da rede usada para localizar faltas envolvendo a fase e terra compreende as decomposições A3 das tensões e correntes das fases em falta além da sequência de corrente nula. Para faltas bifásicas-terra, a decomposição A3 de qualquer fase em falta pode ser usada para estimar a sua localização. A rede usada para localizar faltas de fase usa como vetor de entrada as decomposições A3 da diferença das tensões e correntes das fases em falta. Por sua vez, no caso de falta trifásica, o vetor de entrada é obtido a partir de quaisquer quantidades de duas fases.

Outra abordagem que já foi explorada na classificação e localização de faltas é a árvore de decisão. Swetapadma e Yadav (2017), usaram essa técnica para estimar a distância da ocorrência de faltas em linhas de transmissão de circuito duplo. A localização da falta é estimada usando a informação obtida dos dados dos eventos de falta. As vantagens da árvore de decisão em relação a outros métodos são requerer um menor tempo de treinamento, oferecer maior acurácia com um conjunto de dados grande, além da sua robustez. Esse método é capaz de estimar a localização a partir de estimativas aproximadas e não é afetado por variações de parâmetros como tipo da falta, resistência, ângulo de incidência etc. Nesse projeto, também foi realizado um comparativo entre a Transformada Discreta de Fourier e *Wavelet*, dessa maneira, inicialmente os valores de corrente e tensão são obtidos e em seguida processados utilizando-se a DFT e DWT. Após o processamento, os sinais são usados como entradas para quatro árvores

de decisão, a depender do tipo da falta: monofásica, bifásica com e sem terra ou trifásica. Assim, são geradas regras de decisão que a depender dos valores de corrente e tensão a localização é estimada. Os módulos de treinamento para localizar as faltas são projetados de acordo com os diferentes tipos. Testes foram realizados com a variação dos mais diversos parâmetros para assim se avaliar o desempenho dos modelos. Ao final, verificou-se que o erro percentual ao estimar a localização da falta é de $\pm 1\%$ na maioria dos casos. No entanto, vale destacar que a estimativa baseada na DWT teve a acurácia um pouco superior do que a baseada na DFT.

Dashtdar, Dashti e Shaker (2018) utilizaram duas redes neurais artificiais, uma para a classificação e outra para a localização de faltas em linhas de transmissão. Os sinais de corrente foram pré-processados com a transformada *wavelet* e a entropia foi calculada. A primeira rede implementada teve como objetivo identificar qual o tipo de falta ocorreu, ou seja, diferenciar se trata de uma falta monofásica, bifásica com ou sem presença de terra, ou trifásica. Para essa rede, além da entropia das fases, as sequências de energias positivas e nulas também foram consideradas como entradas. Cada saída, por sua vez, possui valores binários indicando a fase em falta e se existe ou não presença de terra. A segunda RNA, foi utilizada para estimar a distância na qual a falta ocorreu e também para obter a resistência da falta. Os resultados desse estudo mostraram que a entropia calculada depende fortemente do ângulo e da resistência da falta, dessa maneira, a acurácia do algoritmo proposto poderia ser melhorada caso os dois parâmetros fossem considerados como saída da rede neural. Por fim, os algoritmos apresentaram um bom resultado sendo que o erro máximo de estimativa obtido foi de 1,5%.

Maheshwari, Agarwal e Sharma (2019) realizaram um estudo comparativo entre a localização de faltas utilizando a teoria das ondas viajantes e redes neurais artificiais. O treinamento é realizado usando diversos conjuntos de dados disponíveis de um modelo de sistema de potência no qual variados cenários são simulados (localização e tipo da falta). A teoria de ondas viajantes é um método tradicional adotado na localização de faltas em linhas de transmissão. Se denomina onda viajante as ondas de corrente e tensão que viajam de um terminal a outro na linha de transmissão. Quando uma falta ocorre, uma sobrecorrente ou sobretensão surge no ponto da falta. Essa perturbação irá propagar como uma onda viajante em ambas direções com velocidade próxima a da luz, no caso de linhas aéreas. Para esse modelo, inicialmente foi realizada uma transformação modal matricial para desacoplar os sinais de fase em suas equações independentes para cada modo, eles utilizaram a transformação de Clarke para isso. Em seguida, os sinais são processado com a transformada *wavelet* e é realizado um cálculo para verificar se a

falta ocorreu na primeira ou segunda metade da linha. Esse fato é crucial para determinar qual equação será utilizada para estimar a localização da falta. Além disso, conforme citado anteriormente foi realizado um comparativo com um localizador baseado em redes neurais. Para essa finalidade, foram utilizadas duas configurações uma com 3 entradas e a outra com 6, sendo que ambas possuíam uma saída para a estimativa da localização da falta. Como entrada, eles utilizaram os sinais de corrente anteriormente pré-processados com a transformada *Wavelet*. Deve-se destacar que para a primeira rede se utilizou os dados de apenas um terminal, enquanto para a outra configuração os sinais de ambos terminais foram usados. Em todas as situações testadas o erro foi menor do que 0,5%, mas observou-se que a rede neural 2 (com 6 entradas) obteve o menor erro médio, seguida pela rede neural 1 (com 3 entradas) e depois o método baseado em ondas viajantes. Nesse trabalho tratou-se apenas de localização de faltas monofásicas com terra, mas foram avaliados que os localizadores implementados também podem ser utilizados em outros tipos de faltas com a mesma acurácia.

Chaitanya e Yadav (2019) apresentaram um esquema de identificação e localização de faltas com base em ondas viajantes (*Traveling Wave - TW*) auxiliadas por árvores de decisão. Como mostrado anteriormente, ao utilizar TW é necessário saber em qual seção da linha a falta ocorreu, uma vez que isso irá alterar a equação a ser utilizada durante a estimativa da localização. Para determinar isso foi implementada uma árvore de decisão que recebe como entrada o tempo de viagem da onda do terminal local e com duas regras simples é possível determinar se a falta ocorreu na primeira ou na segunda seção da linha, no caso de um sistema com apenas dois terminais. No entanto, essa abordagem foi ampliada para que sistemas com múltiplos terminais também fossem incluídos, o que resultou em uma árvore de decisão maior e mais complexa, uma vez que são consideradas um número maior de seções e regras na árvore. O uso dessa metodologia foi testada em variados tipos de linha de transmissão, distintos tipos de falta e com variações de parâmetros como localização, resistência e ângulo de incidência da falta. Para todos os casos considerados o percentual de erro da localização estimada da falta foi abaixo de 1%.

Outra técnica utilizada em processamento de sinais e que já foi utilizada em trabalhos de detecção, classificação e localização de faltas é a análise de componentes independentes (*independent component analysis - ICA*). Essa é uma técnica estatística para separação cega de fontes e já foi extensivamente utilizada na área médica, especialmente na separação, classificação e análise de sinais de eletroencefalograma. Nessa dissertação não serão explicados a fundo

tais projetos, uma vez que eles fogem do escopo do trabalho proposto, contudo é possível verificar essa aplicação nos trabalhos de Zhou (2002), Leichter (2006), Abe e Nakayama (2007), James, Abasolo e Gupta (2007), Zhu, Tong e Chen (2008), Guo e Wu (2010), Kaur e Singh (2016), Rajaguru e Prabhakar (2017), Michelmann et al. (2018), Shimamoto et al. (2018).

Além disso, é possível encontrar na literatura o uso dessa técnica na área de qualidade da energia elétrica, especificamente na extração de harmônicos, detecção e classificação de distúrbios elétricos. Ran e Zhou (2011), por exemplo, propuseram um método que utiliza o algoritmo FastICA e entropia negativa para detectar harmônicos, obter sinais de onda base e cada sinal harmônico. Os resultados da simulação mostraram que esse método é capaz de detectar com uma relativa precisão cada composição harmônica de múltiplos harmônicos, além disso o sistema proposto não se mostrou ser sensível a ruídos. Dessa maneira, concluiu-se que ele é adequado para detectar harmônicos de qualidade de energia. Cabe destacar que os erros apresentados durante a simulação ficaram abaixo de 0,6%.

Por sua vez, Ferreira et al. (2015) propuseram um método baseado na análise de componentes independentes de canal único (*single channel independent component analysis* - SCICA), que é uma modificação da ICA usada no monitoramento de apenas uma linha, para analisar e classificar eventos de qualidade de energia com múltiplos distúrbios. Nesse trabalho, foi utilizado um filtro *notch* para pré-processamento, além da SCICA e um classificador neural para classificar os distúrbios. O modelo proposto possuía duas limitações, a primeira é que ele é capaz de identificar no máximo 3 diferentes tipos de distúrbio por forma de onda processada e a outra é que ele somente identifica um tipo de transiente. O uso da ICA de canal único em qualidade da energia também pode encontrado nos trabalhos de Coury, Lima e Cerqueira (2012), Lima et al. (2012), Ferreira, Seixas e Cerqueira (2015), Nagata et al. (2018).

No trabalho de Almeida et al. (2017) a ICA foi utilizada em conjunto com a teoria de ondas viajantes e máquinas de vetor de suporte (*support vector machines* - SVM) visando localizar e classificar faltas em linhas de transmissão de alta tensão em sinais com presença de ruído. Como citado nesse capítulo, a teoria de ondas viajantes é uma técnica amplamente utilizada em sistemas de proteção para a localização de faltas, no entanto, o uso desse método é ineficaz em sinais ruidosos. Nesse projeto, os dados com e sem ruído são usados como entrada para treinamento e teste no algoritmo SVM. Foram testadas 3 funções *kernel* distintas: linear, sigmoide e RBF, sobre diferentes condições de ruído cujo *signal-to-noise ratio* (SNR) varia entre 30 dB a 75 dB. Dentre as funções utilizadas, observou-se que a linear não tem êxito ao lidar

com extração de características na classificação de faltas. Por sua vez, a função sigmoide falha ao classificar faltas com ruído. Por fim, a função RBF mostrou melhores resultados obtendo uma taxa de classificação de 100% para sinais com e sem ruído. Quanto a ICA o algoritmo utilizado foi o FastICA, e sua função foi dividir o espaço dos dados em dois novos subespaços contendo dados com presença e ausência de ruído. Dessa maneira, o ponto de incidência da falta na linha de transmissão pode ser corretamente determinado. Apesar dos dados terem sido separados do ruído, o conjunto de dados original (sinal com ruído) foi considerado para validar a efetividade do classificador utilizado. Em resumo, nesse trabalho foi utilizada a análise de componentes independentes como filtro para separar o ruído presente nos sinais de tensão na linha de transmissão. Dessa maneira, a localização da falta é realizada utilizando a teoria das ondas viajantes nos dados sem ruído, enquanto para classificação utilizou-se SVM em conjunto com uma metodologia chamada *multidimensional scaling* (MDS) para permitir a visualização gráfica dos resultados. Por fim, os erros obtidos nesse projeto foram menores do que 1% para a localização de faltas e a acurácia foi de 100% na classificação.

Muitos projetos utilizaram técnicas de inteligência computacional com uma etapa de pré-processamento. Apesar da extensa utilização da transformada de Fourier e *wavelet* nessa fase, algumas outras metodologias como estatística de ordem superior e análise de componentes independentes já foram usadas com esse intuito. Por ser um tema muito abordado na literatura torna-se difícil a tarefa de abranger todos os trabalhos em uma revisão de literatura. No entanto, alguns artigos foram publicados com objetivo de contemplar um número mais variado de projetos que abordam sistemas de proteção em linhas de transmissão. Cabe destacar os artigos de Prasad, Edward e Ravi (2017a) e Prasad, Belwin Edward e Ravi (2017b), nos quais são apresentadas diferentes metodologias usadas na classificação de faltas em linhas de transmissão. Foram incluídas quase todas as técnicas e filosofias relatadas na literatura. Os métodos são divididos nesse trabalho em 3 tipos: proeminentes, híbridos e modernos. O primeiro, compreende o uso de técnicas bem conhecidas e comumente utilizadas para classificação de faltas em linhas de transmissão. O segundo surge a partir da integração das técnicas proeminentes e visa superar alguma desvantagem existente em uma abordagem a partir da integração dela com outra metodologia. Por sua vez, as técnicas modernas compreendem métodos avançados para a classificação de faltas.

Outro estudo que realizou uma revisão bibliográfica para verificar as técnicas mais utilizadas nessa área foi conduzido por Chen, Huang e He (2016a). Nesse trabalho foi apresentada

uma revisão abrangente de diversas técnicas usadas na detecção, classificação e localização de faltas em linhas de transmissão. Além disso, outras etapas do processo foram abordadas como a extração de características, e foram apresentadas as técnicas mais utilizadas nesse estágio como a transformada S , de Fourier, *wavelet*, transformações modais e técnicas de redução de dimensionalidade, como a análise de componentes principais. Em relação a metodologias para classificação de faltas foi citado o uso de fluxo lógico que leva em consideração a comparação de determinados valores extraídos dos dados de tensão ou corrente com certos limiares para decidir qual a fase presente na falta. E além desse, foram apresentados projetos que fizeram uso de outros métodos como redes neurais artificiais, máquinas de vetor de suporte, sistemas de inferência *fuzzy* (*Fuzzy Inference Systems* - FIS) e árvore de decisão. Em relação à localização de faltas, foram abordados os algoritmos para falhas de área ampla, linha de transmissão híbrida e linhas com compensação em série, além dos algoritmos baseados em ANN, FIS e SVM.

Analisando-se os artigos pesquisados na área observou-se que uma das técnicas de inteligência computacional que tem sido amplamente utilizada para a classificação e localização de faltas é a rede neural artificial. No entanto, ela possui algumas desvantagens quando comparada com outros métodos. Em primeiro lugar, geralmente o tempo para realizar o treinamento da rede é elevado dependendo do número de entradas, neurônios, e variáveis do problema. Isso pode vir a gerar modelos consideravelmente complexos. Além disso, quando comparadas com outras técnicas as ANNs possuem muitos hiperparâmetros a serem ajustados antes da realização do treinamento: quantidade de neurônios e camadas escondidas, funções de ativação, algoritmo de otimização etc. Por esse motivo, pretende-se estudar outros métodos de classificação que são mais simples de serem utilizadas como as florestas aleatórias (*random forest*), essa técnica tem o treinamento em geral mais rápido, exige menor processamento dos dados de entrada e possuem menos hiperparâmetros a serem ajustados antes do treinamento. Outra técnica usada para classificação nesse trabalho foi a regressão *ridge* que possui ainda menos hiperparâmetros a serem ajustados, e quando usado com alguma técnica de extração de características pode obter ótimos resultados.

Nos trabalhos analisados, quantidade de sinal pós falta utilizada na classificação das faltas foi pouco explorada ou não informada. Dentre eles é possível citar Yadav e Swetapadma (2015) que utilizaram um ciclo, além dele, Shaik e Pulipaka (2015) usaram um e meio ciclo de sinal. Dessa maneira, nessa dissertação pretende-se variar a quantidade de informação pós-falta entre: um, meio, um quarto, um oitavo, um dezesseis avos, um trinta e dois avos, um

sessenta e quatro avos e um cento e vinte e oito avos de ciclo. Isso permite observar se o método implementado é capaz de realizar corretamente a classificação de faltas quando menos informações do sinal estão disponíveis.

Por fim, deve-se destacar que as duas principais técnicas utilizadas em pré-processamento de sinais que são a transformada de Fourier e *wavelet* já foram extensamente utilizadas em trabalhos dessa natureza. Nesse trabalho buscou-se utilizar técnicas mais simples como o filtro *notch* como etapa de pré-processamento e florestas aleatórias para classificação de faltas. É possível notar também que trabalhos que envolvem conceitos de *deep learning* ainda não foram muito utilizados em projetos dessa área, apesar da grande utilização em trabalhos que envolvem a classificação de séries temporais. Cabe destacar que técnicas de aprendizagem profunda em geral adicionam um complexidade extra desnecessária em projetos que podem ser solucionados de maneira mais simples. No entanto, nesse projeto utilizou-se duas técnicas para transformações de sinais e extração de *features* conhecidas como *MiniRocket* e *Rocket*. Elas permitem transformar os sinais não adicionando tanta complexidade computacional quanto as redes de aprendizagem profundas clássicas.

3 REFERENCIAL TEÓRICO

Nesse capítulo são apresentados resumidamente alguns conceitos teóricos de técnicas e metodologias utilizadas no embasamento pro projeto.

3.1 Filtro *Notch*

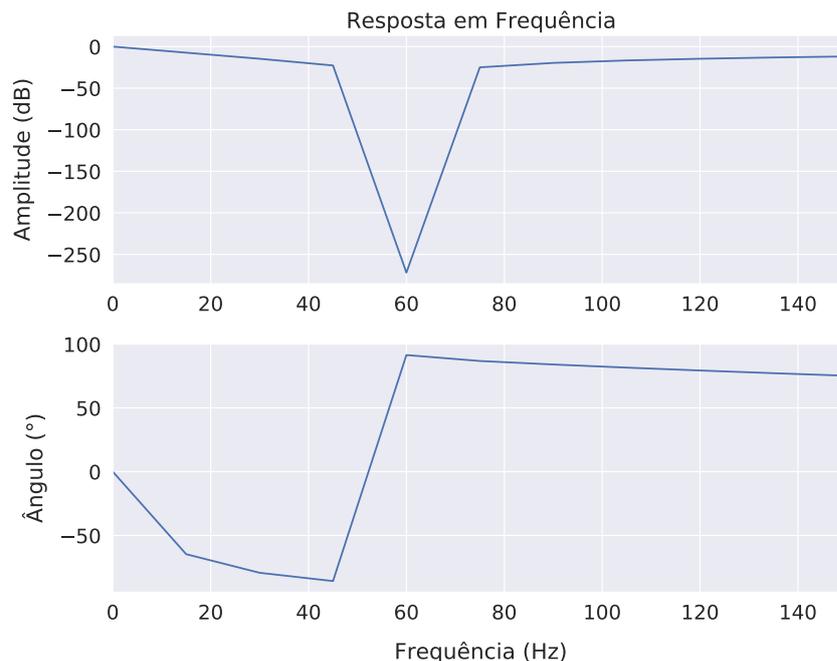
O filtro *notch* é geralmente usado devido ao seu baixo custo computacional e a maior seletividade na frequência de interesse, neste trabalho foi considerado um filtro IIR (*Infinite Impulse Response*) de segunda ordem, sua transformada z é dada pela equação a seguir (FERREIRA, 2010).

$$H(s) = \frac{1 + a_0z^{-1} + z^{-2}}{1 + \rho_0a_0z^{-1} + \rho_0^2z^{-2}} \quad (3.1)$$

onde, $a_0 = -2 \cos \omega_0$ e ρ_0 é o fator *notch* com $0 \ll \rho_0 < 1$.

O filtro *notch* remove apenas uma estreita faixa de frequência, por isso é geralmente usado para remover ruídos de frequência de linha (50 ou 60 Hz). Além disso, também é usado para remover ressonância do sistema, o que faz com que seja amplamente usado em projetos de controladores já que faz isso enquanto deixa passar as altas frequências e cria menos atraso na malha de controle.

Figura 3.1 – Resposta em frequência de um filtro *notch*



Fonte: Do autor (2020).

3.2 *Rocket*

A maioria dos métodos usados em classificação de séries temporais tem alta complexidade computacional ou focam em um único tipo de característica como forma, frequência ou variância. A partir do sucesso das redes neurais convolucionais outros métodos nasceram, um deles é o *Rocket (RandOm Convolutional KErnel Transform)* que conforme apresentam Dempster, Petitjean e Webb (2020) é capaz de obter resultados do estado da arte em acurácia na classificação de séries temporais. Esse método transforma a entrada em um conjunto *features* usando *kernels* (filtros) convolucionais aleatórios e realiza a classificação usando simples classificadores lineares.

Modelos de aprendizagem profunda baseados em *kernels* convolucionais como *ResNet* e *InceptionTime* já se provaram eficientes na classificação de séries temporais. Por basearem-se em um sistema que envolve treinamento as redes convolucionais possuem um custo computacional maior, em contraste *Rocket* se baseia na geração de um alto número de *kernels* convolucionais aleatórios, que quando combinados, capturam características relevantes para a classificação de séries temporais. Dempster, Petitjean e Webb (2020) mostraram que o método proposto é capaz de obter acurácia de estado da arte mas com uma fração do tempo quando comparado com os métodos atualmente usados.

Rocket é mais escalável para grandes conjuntos de dados, com complexidade de treino linear independente do tamanho da série temporal ou da quantidade de amostras no conjunto de treinamento. Além disso, esse método é por natureza paralelizável, e pode ser executado ainda mais rapidamente usando múltiplos núcleos da CPU.

Alguns autores chegaram a conclusão de que métodos baseados em *kernels* convolucionais seriam efetivos na classificação de séries temporais devido ao sucesso que eles apresentaram anteriormente na classificação de imagens. Esses métodos possuem uma premissa diferente quando comparados aos métodos clássicos, em vez de usarem uma representação pré concebida das séries temporais, *kernels* convolucionais detectam padrões nas entradas. Ao aprender os pesos dos *kernels*, uma rede neural convolucional aprende as características na série temporal associadas com diferentes classes.

Segundo Fawaz et al. (2019) a convolução pode ser vista como a aplicação de um filtro (*kernel*) deslizando sobre uma série temporal. O filtro pode ser entendido ainda como uma transformação genérica não linear da série temporal, e pode ter apenas uma dimensão tratando-se de séries univariadas ou n dimensões para multivariadas. Para exemplificar, considerando

um filtro de comprimento 3 com uma série temporal univariada, ajustando os valores do filtro para serem iguais a $[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ a convolução resultará em aplicar uma média móvel com janela deslizante de tamanho 3.

O resultado de uma convolução em uma série temporal X pode ser considerada como outra série temporal univariada C que passou por um processo de filtragem. Assim, aplicando vários filtros em uma série temporal resultará em uma série temporal multivariada cujas dimensões são iguais ao número de filtros usados. A aplicação de múltiplos filtros pode ser importante para o aprendizado de múltiplas características discriminativas em uma tarefa de classificação.

Na rede neural convolucional (*convolutional neural network* - CNN) os pesos e biases são usados para encontrar o resultado em todo intervalo de tempo $t \in [1, T]$ isso permite as CNNs aprenderem filtros que sejam invariantes em toda a dimensão do tempo. Esses pesos e biases compartilhados formam o que é chamado de mapa de características (*feature map*), que faz exatamente o mapeamento entre a entrada e a camada oculta.

Os parâmetros básicos de um *kernel* são seu tamanho (comprimento), pesos e biases, dilatação e *padding*, ele tem em geral a mesma estrutura que a entrada porém menor. Tratando-se de séries temporais, um *kernel* é um vetor de pesos, no qual o termo bias é adicionado ao resultado da operação de convolução entre a série temporal de entrada e os pesos do *kernel*. A dilatação 'propaga' um *kernel* sobre a entrada de tal forma que considerando, por exemplo, a dilatação sendo 2, os pesos do *kernel* são convolucionados com cada segundo elemento da entrada. *Padding* é o processo de adicionar valores (geralmente zero) no início e no fim da entrada, para que o peso central do kernel se alinhe com o primeiro elemento da entrada no início da operação de convolução (DEMPSTER; PETITJEAN; WEBB, 2020).

Rocket transforma as séries temporais usando um alto número de *kernels* convolucionais aleatórios, ou seja, com comprimento, pesos, bias, dilatação e *padding* aleatórios. As *features* transformadas são usadas para treinar um classificador linear. Segundo Dempster, Petitjean e Webb (2020), para conjuntos de dados que não sejam grandes o classificador de regressão *ridge* é indicado já que tem a vantagem de validação cruzada rápida para o hiperparâmetro de regularização (e nenhum outro hiperparâmetro). Por sua vez, regressão logística treinada usando gradiente descendente estocástico é mais escalável para conjuntos de dados grandes.

Algumas coisas diferenciam *Rocket* de camadas convolucionais tipicamente usadas em redes neurais convolucionais e métodos semelhantes anteriormente criados. Maiores detalhes sobre essas diferenças podem ser verificados no artigo original desse método (DEMPSTER;

PETITJEAN; WEBB, 2020), mas resumidamente: *Rocket* usa um alto número de *kernels*; os pesos não são aprendidos, mas sim aleatórios; ele usa uma grande variedade de *kernels*; ele também faz uso da dilatação de forma aleatória o que acaba sendo responsável por capturar padrões em diferentes frequências e escalas; esse método usa também uma *feature* adicional e original que é a proporção dos valores positivos (*proportion of positive values* - ppv), responsável por ponderar a prevalência de um dado parâmetro dentro de uma série temporal.

O único hiperparâmetro a ser ajustado no *Rocket* é o número de *kernels*, o que conduz a um *tradeoff* entre a acurácia e o tempo computacional. Em geral, um número alto de *kernels* resulta em uma alta acurácia de classificação, mas à custa de uma computação proporcionalmente mais longa. Os autores usaram o valor de 10000 *kernels* como padrão do método, e apesar de ser um número alto, *Rocket* se mostrou um método extremamente rápido.

Os parâmetros dos *kernels* são obtidos da seguinte maneira:

- **Comprimento:** selecionado aleatoriamente do conjunto $\{7, 9, 11\}$ com probabilidade igual;
- **Pesos:** amostrados de uma distribuição normal, $\forall k \in \mathbf{K}, k \sim \mathcal{N}(0, 1)$ e são centralizados na média depois de ajustados, $k = \mathbf{K} - \bar{\mathbf{K}}$;
- **Bias:** amostrados de uma distribuição uniforme $b \sim \mathcal{U}(-1, 1)$, somente valores positivos no mapa de características são usados;
- **Dilatação:** é amostrada em uma escala exponencial $d = \lceil 2^x \rceil$, $x \sim \mathcal{U}(0, A)$ onde $A = \log_2 \frac{l_{input} - 1}{l_{kernel} - 1}$, que assegura o comprimento efetivo do *kernel*, incluindo a dilatação, até o comprimento da série temporal de entrada l_{input} ;
- **Padding:** quando cada *kernel* é gerado, o modelo decide (aleatoriamente) se *padding* será usado ou não. Caso seja usado, o valor de zero *padding* é anexado no início e no final de cada série temporal quando o *kernel* é aplicado, de tal forma que o elemento central do *kernel* é centralizado em cada ponto da série temporal, ou seja, $((l_{kernel} - 1) \times 2) / 2$. Sem *padding*, os *kernels* não são centralizados no primeiro e último $\lceil l_{kernel} / 2 \rceil$ pontos da série temporal, e "foca" em padrões nas regiões centrais da série temporal enquanto que com *padding*, *kernels* também correspondem a padrões no início ou no final da série temporal.

Stride (que é o passo usado durante o processo de convolução) é sempre um. Além disso, não são aplicadas não linearidades aos *feature maps*. Importante observar que os parâmetros

para pesos e biases são ajustados baseados na suposição que, como uma prática padrão, séries temporais de entrada foram normalizadas para ter média zero e desvio padrão igual a um.

Em resumo cada *kernel* é aplicado a cada entrada, produzindo um *feature map*. A operação de convolução envolve um produto interno entre um *kernel* e uma série temporal de entrada. Esse processo é descrito pela equação 3.2.

$$X_i * k = \sum_{j=0}^{l_{kernel}-1} X_{i+(j \times d)} \times k_j \quad (3.2)$$

onde k representa o *kernel*, d a dilatação, X a série temporal, i a posição na série.

A equação 3.2 gera vários mapas de características na qual são computadas 2 *features* para cada mapa, produzindo dois números com valor real como recursos por *kernel*: o valor máximo e a proporção de valores positivos (ppv). O ppv produz maior acurácia de classificação do que outras *features* como a média, por exemplo. Para k *kernels*, *Rocket* produz $2k$ *features* por série temporal.

Apesar do *Rocket* poder ser usado com qualquer classificador, os autores encontraram que ele é muito efetivo quando usado com classificadores lineares. Por esse motivo, no artigo original são propostos o uso da regressão logística e regressão *ridge*.

3.3 *MiniRocket*

MiniRocket (*MINImally RandOm Convolutional KErnel Transform*) é uma reformulação do método *Rocket* 75 vezes mais rápido em grandes conjuntos de dados sendo quase totalmente determinístico, enquanto mantém essencialmente a mesma acurácia. Para isso, *MiniRocket* usa um conjunto pequeno e fixo de *kernels*, ele mantém os dois mais importantes aspectos do *Rocket*: dilatação e proporção de valores positivos (ppv). E, além disso, explora várias propriedades dos *kernels* e da ppv, para efetivamente reduzir o tempo necessário para a transformação. Enquanto *kernels* convolucionais aleatórios são altamente efetivos, é possível essencialmente obter a mesma acurácia usando um procedimento principalmente determinístico e muito mais rápido (DEMPSTER; SCHMIDT; WEBB, 2021).

MiniRocket faz mudanças cruciais para remover quase toda aleatoriedade do *Rocket*, e explora essas modificações para aumentar a velocidade da transformação. A Tabela 3.1 resume as principais diferenças entre os dois métodos.

Tabela 3.1 – Comparativo entre métodos *Rocket* e *MiniRocket*

	<i>Rocket</i>	<i>MiniRocket</i>
comprimento	{7, 9, 11}	9
pesos	$\mathcal{N}(0, 1)$	{-1, 2}
biases	$\mathcal{U}(-1, 1)$	da saída da convolução
dilatação	aleatório	fixo (ajustado ao comprimento da entrada)
<i>padding</i>	aleatório	fixo
<i>features</i>	ppv + max	ppv
num. <i>features</i>	20.000	10.000

Fonte: Dempster, Schmidt e Webb (2021)

Visando maximizar eficiência, o conjunto de *kernels* deve ser o menor possível. Em relação ao comprimento, no *MiniRocket* existem $2^9 = 512$ possíveis *kernels*, no entanto, os autores (DEMPSTER; SCHMIDT; WEBB, 2021) sugerem utilizar um subconjuntos de 84 desses filtros de tal forma que é balanceada a acurácia com a baixa complexidade computacional.

Os pesos usados nos *kernels* foram setados de forma arbitrária para serem -1 ou 2, a escala não é importante já que o bias é extraído da saída da convolução e assim, por definição, coincide com a escala dos pesos e a escala da entrada. No entanto, é importante que a soma dos pesos seja zero, isso garante que a saída da convolução seja invariante à adição ou subtração de qualquer constante na entrada.

O conjunto de 84 *kernels* de tamanho 9, com três pesos com valor 2 e seis com o valor -1 tem as propriedades desejadas de ser relativamente pequeno e produzir alta acurácia. No entanto, subconjuntos de *kernels* de tamanho 9, ou outros tamanhos fornecerem uma acurácia similar. Os autores, a partir se seus testes, apenas decidiram utilizar por padrão esse subconjunto de 84 *kernels*.

Os valores de bias, como citado, são extraídos da saída da convolução e são usados para computar a ppv, conforme a equação 3.3.

$$ppv(X * k) = \frac{1}{n} \sum [X * k > b] \quad (3.3)$$

onde, X representa a série temporal, k o *kernel*, e b o *bias*.

Basicamente, um exemplo de treinamento é escolhido aleatoriamente (esse é o único elemento estocástico do *MiniRocket*), a convolução para essa amostra e uma dada combinação de *kernel*/dilatação é computada, e os valores dos quantis [0,25; 0,5; 0,75] são usados como bias para computar a ppv.

A dilatação é usada para "espalhar" um *kernel* sobre a entrada. Considerando o valor de dilatação como d um dado *kernel* é convolucionado com cada d -ésimo da entrada. Por exemplo, se $d = 2$ a convolução ocorre com cada segundo elemento da série temporal de entrada. O valor de d pode variar entre 1 e $\log_2(l_{input} - 1)/(l_{kernel} - 1)$, onde l_{input} é o comprimento da entrada e l_{kernel} o comprimento do *kernel*. Um maior número de *features* é calculado para menores dilatações, por padrão no *MiniRocket* utilizou-se $d = 32$ para garantir a eficiência do modelo.

Esse método usa *padding* zero para metade das combinações possíveis de *kernel* / dilatação enquanto a outra metade não usa *padding* algum. Esse anexo de vetores com zero garante que a convolução comece com o elemento central do *kernel* centrado no primeiro elemento da série temporal, e que termine com ele centrado no último elemento da série temporal.

No *MiniRocket*, somente a ppv é usada como *feature* para cada *kernel*. Além disso, utilizando 10000 *features* o método proposto rivaliza com o *Rocket* em acurácia, existindo pouco ou nenhum benefício em usar um número maior que esse.

A transformação da série temporal usando *MiniRocket* é otimizada devido alguns fatores. Dentre eles, cabe citar que a saída da operação de convolução é reusada para calcular múltiplas *features*, por exemplo para vários valores de *bias* diferentes. Isso reduz o custo computacional de computar a ppv para cada valor de *bias*.

Outro fator que otimiza a transformação é a restrição dos pesos dos *kernels* a dois valores apenas. Isso permite com que a operação de convolução seja executada usando apenas adição. Para uma série temporal $X = [x_0, x_1, \dots, x_{n-1}]$ e *kernel* $k = [k_0, k_1, \dots, k_{m-1}]$, com dilatação d , a convolução é calculada usando a equação 3.4.

$$X * k = \sum_{j=0}^{m-1} x_{i - (\frac{m}{2} \cdot d) + (j \cdot d)} \cdot k_j, \forall i \in \{0, 1, \dots, n-1\} \quad (3.4)$$

De forma equivalente, a operação de convolução pode ser considerada como a soma das colunas de uma matriz \hat{C} na qual cada linha corresponde a multiplicação da série temporal de entrada pelo peso apropriado do *kernel*, e o alinhamento das linhas corresponde à dilatação, como no exemplo a seguir.

$$\hat{\mathbf{C}} = \begin{pmatrix} 0 & 0 & 0 & 0 & k_0x_0 & \cdots \\ 0 & 0 & 0 & k_1x_0 & k_1x_1 & \cdots \\ 0 & 0 & k_2x_0 & k_2x_1 & k_2x_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ k_{m-1}x_4 & k_{m-1}x_5 & k_{m-1}x_6 & k_{m-1}x_7 & k_{m-1}x_8 & \cdots \end{pmatrix} \quad (3.5)$$

O resultado da operação de convolução é dado pela soma das colunas de $\hat{\mathbf{C}}$: $C = X * K = \mathbf{1}^\top \hat{\mathbf{C}}$, onde $\mathbf{1}^\top$ é um vetor $[1, 1, \dots, 1]^\top$ de tamanho n . Os pesos dos *kernels* são restritos a dois valores, α e β , e é possível "fatorar" as multiplicações pré-computando $A = \alpha X$ e $B = \beta X$, então para um dado *kernel*, por exemplo, $K = [\alpha, \beta, \alpha, \dots, \alpha]$, completando a operação de convolução usando um somatório $A = [a_0, a_1, \dots, a_{n-1}]$ e $B = [b_0, b_1, \dots, b_{n-1}]$:

$$\hat{\mathbf{C}} = \begin{pmatrix} 0 & 0 & 0 & 0 & a_0 & \cdots & a_{n-5} \\ 0 & 0 & 0 & b_0 & b_1 & \cdots & b_{n-4} \\ 0 & 0 & a_0 & a_1 & a_2 & \cdots & b_{n-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_4 & a_5 & a_6 & a_7 & a_8 & \cdots & 0 \end{pmatrix} \quad (3.6)$$

Ou seja, é necessário somente computar αX e βX uma única vez para cada série temporal de entrada e então reusar os resultados para completar a operação de convolução para cada *kernel* por adição.

O que foi mostrado nessa seção contribui para que o *MiniRocket* produza uma acurácia semelhante ao *Rocket* mas com menor custo computacional. Isso faz com que ele seja atualmente o estado da arte em classificação de séries temporais. Mais detalhes sobre seu funcionamento e para verificar os testes de performance que foram executados basta consulta Dempster, Schmidt e Webb (2021).

3.4 Classificadores usados com *MiniRocket* e *Rocket*

Como citado anteriormente os artigos originais tanto o *Rocket* (DEMPSTER; PETIT-JEAN; WEBB, 2020) quanto o *MiniRocket* (DEMPSTER; SCHMIDT; WEBB, 2021) usam ou regressão logística ou regressão *ridge* como métodos de classificação das séries temporais após as transformações.

A regressão logística é particularmente adequada em *datasets* muito grandes porque ela fornece rápido treinamento com custo de memória fixo pelo tamanho de cada mini lote (*mini-batch*). A transformação pode ser executada em cada *minibatch*, ou em parcelas maiores do conjunto de dados que são então divididas em *minibatches* para treinamento. Por sua vez, para conjuntos de dados pequenos, a regressão *ridge* é significativamente rápida enquanto obtém a mesma acurácia da regressão logística, mas para *dataset* maiores esse método não é escalável, além de exigir a transformação completa e não trabalhar incrementalmente (DEMPSTER; PETITJEAN; WEBB, 2020).

Basicamente, é recomendado usar a a regressão *ridge* na maioria dos casos alterando para regressão logística apenas quando há mais amostras de treinamento do que *features*, ou seja, quando há mais do que aproximadamente 20000 amostras para a transformação *Rocket* ou 10000 para a *MiniRocket*. Assim, nesse projeto decidiu-se utilizar a regressão *ridge*, mais detalhes sobre o seu funcionamento podem ser vistos a seguir.

3.4.1 Regressão *ridge*

A regressão *ridge* é em geral usada como um método de penalização ou regularização quando se é necessário prever uma variável contínua. O seu uso se deve ao fato de que a regressão simples cujo objetivo é minimizar a soma dos erros quadráticos possui uma forte tendência de *overfitting* quando há muitas variáveis principalmente quando existe multicolinearidade entre elas.

Basicamente, a regressão *ridge* endereça alguns problemas dos Mínimos Quadrados Ordinários impondo uma penalidade no tamanho dos coeficientes da função a ser minimizada. Dessa maneira, é adicionada a função um termo quadrático, como pode ser visto na equação 3.7.

$$\min_w \|Xw - y\|_2^2 + \alpha \|w\|_2^2 \quad (3.7)$$

onde o primeiro termo da equação representa a soma dos erros quadráticos. Xw são os valores preditos e y representa os valores reais. O segundo termo $\alpha \|w\|_2^2$ representa a regularização, que no caso da regressão *ridge* é um termo quadrático. O parâmetro α controla o quanto os coeficientes da equação serão encolhidos. Quanto maior o seu valor, maior o encolhimento, assim os coeficientes se tornam mais robustos a colinearidade (SCIKIT-LEARN, 2021b). No entanto, se os valores de todos os coeficientes se tornarem muito pequenos, é possível que o modelo torne-se subajustado (ocorra o *underfitting*).

Regularização é criticamente importante quando o número de *features* é significativamente maior do que o número de amostras de treinamento, permitindo a otimização de modelos lineares, e prevenindo o comportamento patológico na otimização iterativa. A regressão *ridge* pode explorar validação cruzada generalizada para determinar rapidamente o valor apropriado para o parâmetro de regularização (DEMPSTER; PETITJEAN; WEBB, 2020).

Quando α é muito grande, o efeito da regularização domina a função de perda quadrática e os coeficientes tendem a zero. Como α tende a zero e a solução tende para os mínimos quadrados, os coeficientes exibem grandes oscilações. Na prática, é necessário sintonizar α de tal forma que o equilíbrio entre os dois seja mantido (SCIKIT-LEARN, 2021a).

A regressão *ridge* pode ser levemente alterada para ser usada em problemas de classificação. Caso se trate de um problema de classificação binária, os valores alvos são convertidos para $\{-1, 1\}$ e então o problema é tratado como uma tarefa de regressão, otimizando a função objetiva apresentada na equação 3.7. Nesse projeto, utilizou-se a implementação de regressão *ridge* disponível na biblioteca *scikit learn* do *Python*, na qual, os problemas de classificação multi-classe são tratados como regressão multi-saída, e a classe predita corresponde à saída com maior valor. Esse modelo pode ser significativamente mais rápido do que a regressão logística com um alto número de classes porque ele é capaz de calcular a matriz de projeção $(X^T X)^{-1} X^T$ uma única vez (SCIKIT-LEARN, 2021b).

A implementação do classificador *ridge* no *scikit learn* usa a estratégia um contra todos (*one-vs-all*) na classificação (SINGH; PRAKASH; CHANDRASEKARAN, 2016). Para isso é realizado o treinamento de c classificadores *ridge*, onde c é o número de classes do problema. Ao aprender o c -ésimo classificador todos os pontos que não estão na classe c são tratados como uma única classe "não- c ". Assim, para cada classe c é temporariamente atribuído aos pontos de treinamento o rótulo $+1$, enquanto para todos os outros pontos das outras classes é atribuído o valor -1 . Com esses rótulos temporários o classificador é treinado para distinguir os pontos na classe c de todas as outras classes. Para rotular cada ponto do espaço de características é usada uma regra de fusão na qual as amostras são rotuladas de acordo com qualquer classificador que produza a maior avaliação (WATT REZA BORHANI, 2016).

3.5 Florestas Aleatórias

Florestas Aleatórias é um método de inteligência computacional que opera a partir da construção de múltiplas Árvores de Decisão durante a etapa de treinamento. Ele foi proposto

por Breiman (2001), que definiu esse método como sendo um classificador consistindo em uma coleção de árvores estruturadas $\{h(\mathbf{x}, \Theta_k), k = 1, \dots\}$ onde o $\{\Theta_k\}$ são vetores aleatórios independentemente distribuídos e cada árvore realiza um voto unitário para a classe mais popular da entrada x .

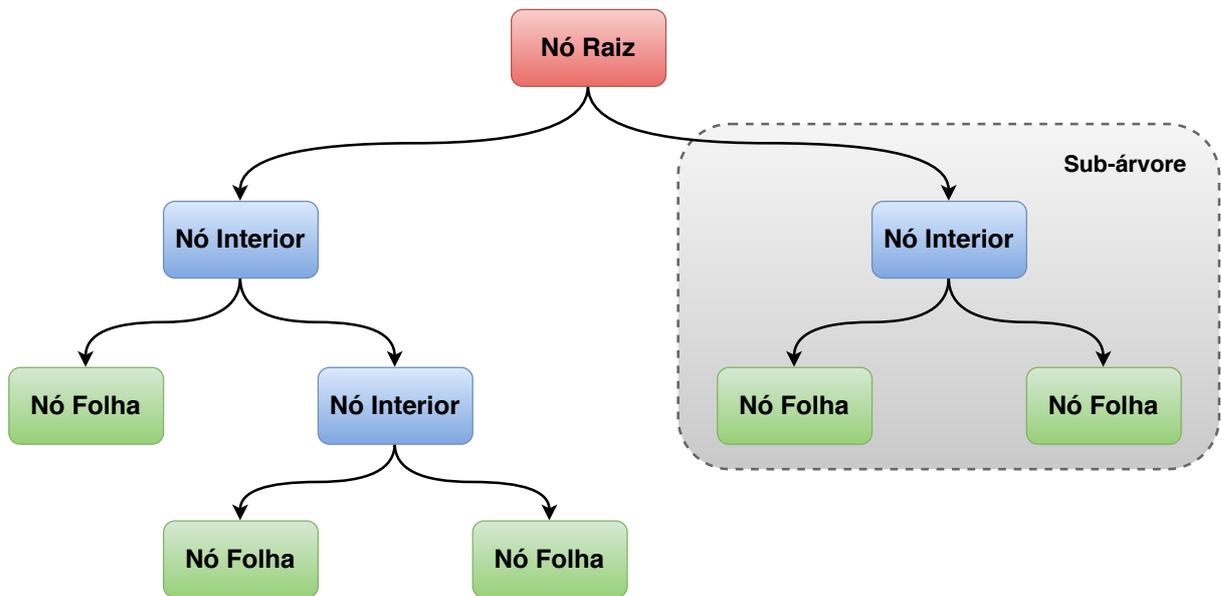
Apesar da definição anterior, Florestas Aleatórias podem ser usadas tanto em problemas de classificação quanto regressão. No primeiro caso o resultado final é obtido a partir da maioria de votos dentre todas as árvores enquanto no segundo ele resulta do cálculo da média entre os valores finais obtidos pelas árvores. Algumas vantagens do uso de Florestas Aleatórias em relação a Árvores de Decisão e outros métodos é que em geral esse método reduz o risco de sobre-ajuste (*overfitting*) dos dados, executa eficientemente em grandes conjuntos de dados produzindo alta acurácia e pode ser usado quando existem dados faltantes garantindo a precisão do modelo. Para se entender Florestas Aleatórias é importante saber alguns conceitos sobre Árvores de Decisão, por esse motivo uma breve explicação será fornecida na subseção a seguir.

3.5.1 Árvores de Decisão

Árvores de Decisão são diagramas nos quais suas ramificações representam possíveis soluções para determinado problema. O modelo criado visa prever o valor da variável de saída aprendendo regras de decisão inferidas a partir das características (*features*) dos dados (SCIKIT-LEARN, 2019b). Essa técnica se tornou comum em problemas de inteligência computacional por poder ser aplicada tanto em problemas de regressão quanto classificação, devido a sua simplicidade e fácil interpretabilidade.

Uma Árvore de Decisão é composta por diversos nós: nó raiz, nós interiores e nós terminais (ou folhas). Os dois primeiros são conectados a estágios de decisão; por sua vez os nós terminais representam as classes ou valores finais preditos pelo modelo. Os nós contidos em um mesmo nível representam uma camada. Cada nó consiste em um conjunto de classes a serem discriminadas, o conjunto de características (*features*) a serem utilizados e a regra de decisão para executar a classificação (Swain; Hauska, 1977). A Figura 3.2 mostra a estrutura de uma Árvore de Decisão simples.

Figura 3.2 – Diagrama de Árvore de Decisão.



Fonte: Do autor (2020).

O algoritmo de Árvore de Decisão é baseado nas etapas básicas a seguir (NAVLANI, 2018):

- a) Selecionar a melhor característica (*feature*) usando alguma métrica de seleção de atributo (ganho de informação, razão de ganho, índice de Gini) para dividir as observações;
- b) usar essa característica com um nó de decisão e quebrar o conjunto de dados em subconjuntos menores;
- c) iniciar a construção da árvore repetindo esse processo recursivamente para cada nó filho até uma das condições ser satisfeita:
 - não existirem características remanescentes;
 - não existirem mais observações no conjunto de dados;
 - o nó terminal ser puro, o que indica que a métrica usada para seleção de atributo é a melhor possível.

Existem algumas métricas usadas na seleção do critério de divisão que possibilite particionar os dados da melhor maneira possível. As mais utilizadas são o ganho de informação, taxa de ganho e índice de Gini.

3.5.1.1 Ganho de Informação

O ganho de informação (*information gain*) é baseado no conceito de entropia introduzido por Shannon (1948), ele mede a impureza de um grupo de amostras. O ganho de informação computa a diferença entre a entropia antes da divisão e calcula a sua média após a divisão do conjunto de dados baseado nos valores de determinadas características (ou atributo). O algoritmo ID3 ou *Iterative Dichotomizer* sugerido por Quinlan (1986) utiliza essas métricas como critério de divisão. Segundo o autor, a entropia pode ser calculada pela equação 3.8.

$$Info(D) = - \sum_{i=1}^m p_i \cdot \log_2(p_i) \quad (3.8)$$

onde, p_i representa a probabilidade de uma amostra arbitrária em D pertencer à classe i .

Assim, a informação esperada necessária para classificar uma amostra de D com base no particionamento A é dada pela equação 3.9.

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \cdot Info(D_j) \quad (3.9)$$

onde, $|D_j|/|D|$ age como o peso da j -ésima partição.

Dessa maneira, a informação ganha pode ser definida pela equação 3.10.

$$Gain(A) = Info(D) - Info_A(D) \quad (3.10)$$

Por fim, o algoritmo escolhe o atributo com ganho ($Gain(A)$) máximo como nó raiz, o que significa escolher os atributos com $Info_A(D)$ mínimo (SHAO et al., 2001).

3.5.1.2 Taxa de ganho

O ganho de informação possui uma desvantagem que é a tendência de selecionar atributos que possuam muitos valores, por essa razão, alguns autores criaram a taxa de ganho definida pela equação 3.11 (DAI; JI, 2014; SINGH; GUPTA, 2014).

$$GainRatio(A) = \frac{Gain(A)}{Info(D)} \quad (3.11)$$

onde, $Gain(A)$ é o ganho de informação definido na equação 3.10 e $Info(D)$ a entropia definida na equação 3.8.

O atributo com maior taxa de ganho é escolhido para a divisão. Essa métrica é usada no algoritmo C4.5 que representa uma evolução do ID3, e gera uma árvore de decisão para os dados a partir da divisão recursiva dos dados.

3.5.1.3 Índice de Gini

Outro algoritmo famoso para construção de Árvores de Decisão é o CART (*Classification and Regression Trees*) introduzido por Breiman et al. (1984). Ele utiliza o índice de Gini como método para criação das divisões e seleção de atributos na árvore, para calculá-lo é utilizada a equação 3.12.

$$Gini(t) = \sum_{i=1}^m p_i(1 - p_i) \quad (3.12)$$

onde p_i é a frequência relativa (que é determinada dividindo o número total de observações da classe pelo número total de observações) da classe i no nó t , e o nó t representa qualquer nó (pai ou filho) no qual uma dada divisão do dado é executada. O atributo com valor mínimo para o índice de Gini é escolhido como atributo da divisão.

3.5.2 Bagging

Bagging (abreviação de *bootstrap aggregating*) é um método de agregação proposto por Breiman (1996) e usado para agregar as Árvores de Decisão criando as Florestas Aleatórias. No entanto, antes de explicar esse método é importante conhecer o *bootstrap*.

O *bootstrap* é um método estatístico poderoso para estimar a quantidade de uma amostra de dado. Considerando um conjunto de dados maior, essa técnica gera amostras menores sorteadas aleatoriamente com reposição. Dessa maneira, o novo conjunto gerado terá boas propriedades estatísticas, sendo composto por amostras representativas e independentes da distribuição real dos dados.

Para entender melhor esse processo considere uma amostra de 100 valores (x) da qual se gostaria de estimar a sua média. Isso pode ser calculado utilizando-se a equação 3.13 (BROWNLEE, 2016).

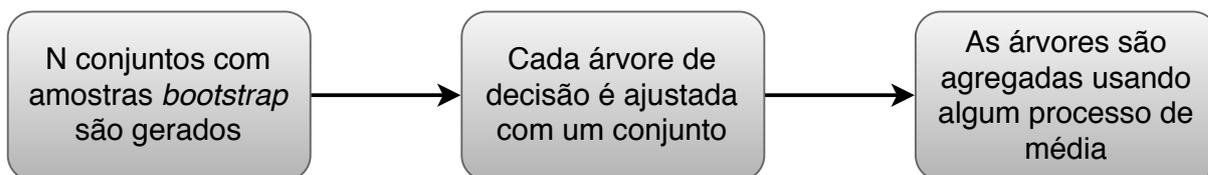
$$mean(x) = \frac{1}{100} \times \sum_{i=1}^{100} x_i \quad (3.13)$$

Como a amostra é pequena pode-se melhorar a estimativa da média usando o procedimento bootstrap:

- a) Criar muitas subamostras aleatórias do conjunto de dados com reposição;
- b) calcular a média de cada subamostra;
- c) calcular a média de todas as médias coletadas e usar isso como a média estimada para os dados.

Portanto, no *bagging*, o objetivo é ajustar vários modelos independentes e calcular a médias das previsões fornecidas por eles. Contudo, na prática não é viável realizar esse ajuste dos modelos de maneira totalmente independente pois exigiria uma grande quantidade de dados, por esse motivo são utilizadas amostras *bootstrap*. O resumo desse processo pode ser verificado na Figura 3.3.

Figura 3.3 – Fluxograma do processo de *bagging*.



Fonte: ??).

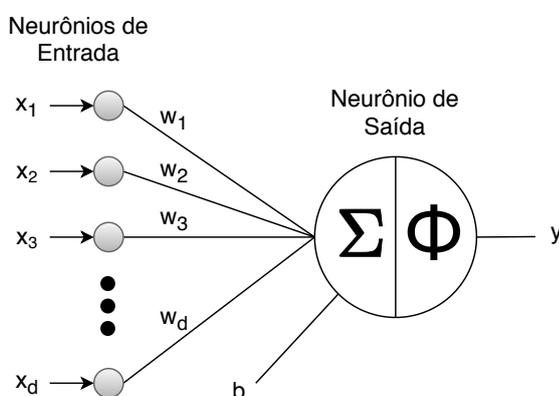
Mesmo utilizando o método *bagging* as árvores de decisão podem ter muitas semelhanças estruturais e fornecerem resultados com alta correlação em suas previsões. O método de florestas aleatórias altera o algoritmo de tal maneira que as subárvores são aprendidas, para que as previsões resultantes de todas as subárvores tenham menos correlação. No CART, ao selecionar o ponto de divisão, o algoritmo de aprendizado é permitido olhar através de todas as variáveis e seus valores para selecionar o ponto de divisão ideal. O algoritmo de Florestas Aleatórias altera esse procedimento de tal forma que ele tem sua busca limitada por uma amostra aleatória de características. O número de característica que podem ser procuradas em cada ponto de divisão deve ser especificado como um parâmetro do algoritmo. Diferentes valores podem ser ajustados usando validação cruzada (BROWNLEE, 2016).

3.6 Redes neurais artificiais

A rede neural artificial (*artificial neural network* - ANN) é um técnica de inteligência computacional muito popular inspirada em organismos biológicos. O seu objetivo é simular o funcionamento biológico de redes neurais de forma artificial em um determinado algoritmo, dessa maneira, as ANNs são compostas por unidades computacionais conectadas uma as outras através de pesos e *bias*. Uma rede neural artificial computa uma função das entradas propagando os valores computados dos neurônios de entrada para os neurônios de saída e usando os pesos como parâmetros intermediários. O estímulo externo em ANNs é fornecido pelos dados de treinamento contendo exemplos de pares entrada-saída da função a ser aprendida.

Por exemplo, suponto que as entradas contenham representações em pixel de imagens (entrada) e seus respectivos rótulos (classes) de saída. Esses pares de treinamento são alimentados na rede neural usando as representações da entrada para fazer previsões sobre as classes de saída. Sabendo-se os valores corretos e aqueles preditos pelo modelo é possível saber quão boa foi a previsão fornecida pelo modelo gerando um valor de erro. Assim, uma técnica de otimização é utilizada para ajustar os pesos da rede visando modificar a função computada para fazer as previsões serem as mais corretas possíveis em futuras iterações. A partir desse processo de sucessivo ajuste de pesos sobre um conjunto suficiente de pares de entrada-saída a rede irá fornecer previsões mais precisas. Essa capacidade de calcular com precisão as funções de entradas não vistas treinando sobre um conjunto finito de pares de entrada-saída é chamada de generalização de modelo (AGGARWAL, 2018). A Figura 3.4 apresenta uma unidade computacional de rede neural, que também pode ser chamada de neurônio ou perceptron.

Figura 3.4 – Arquitetura de um único neurônio.



Fonte: Adaptado de Aggarwal (2018).

Pela Figura 3.4 é possível observar que um neurônio é conectado a entradas (ou os neurônios da camada anterior caso o neurônio em questão esteja em uma camada intermediária) por meio de pesos ($w_1, w_2, w_3, \dots, w_d$), a unidade faz uma soma ponderada desses valores soma com o valor do bias (b) e passa por uma função de ativação (representada por $\Phi(\cdot)$) para fornecer a saída y . Isso é resumido pela equação 3.14.

$$y = \Phi \left(\sum_{i=1}^d w_i x_i + b \right) \quad (3.14)$$

onde, y é a saída gerada pelo neurônio, $\Phi(\cdot)$ representa uma função de ativação usada para saturar a saída em um determinado valor e gerar não linearidades, w_i o vetor de pesos, x_i o vetor de dados de entrada, b o *bias*.

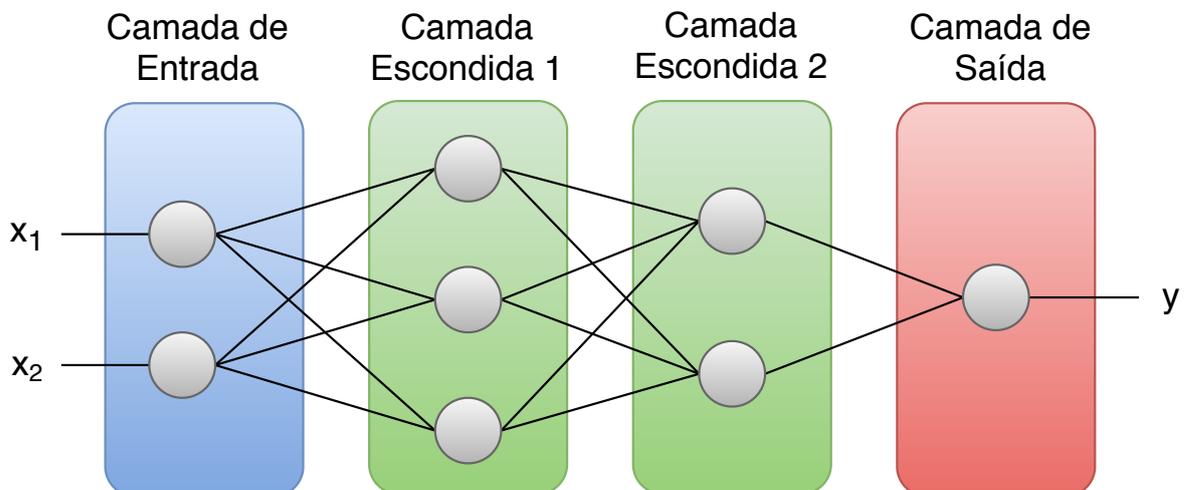
Cabe destacar que uma função de ativação comumente utilizada, especialmente na camada de saída, é a função sigmoide que é descrita pela equação 3.15.

$$y_j = \frac{1}{1 + \exp^{-x_j}} \quad (3.15)$$

onde, y_j representa a saída do neurônio, x_j o vetor de entrada e j um índice indicativo do neurônio.

A combinação de diversos neurônios como aquele apresentado na Figura 3.4 gera uma rede neural ou rede MLP (*multilayer perceptron*) cuja arquitetura pode ser vista na Figura 3.5.

Figura 3.5 – Arquitetura de uma Rede Neural Artificial.



Fonte: Do autor (2020).

Na Figura 3.5 os *biases* foram omitidos propositalmente para evitar com que a figura ficasse poluída. Nela é possível observar que uma ANN possui diversas camadas, a camada de entrada recebe conjunto de dados que possuem os exemplos para treinamento da rede. Essa informações são multiplicadas pelos pesos (representadas pelas linhas entre neurônios) e passada à primeira camada oculta na qual cada neurônio vai realizar o somatório e passar pela função de transferência conforme a equação 3.14 e passadas para a camada seguinte. As mesmas operações são realizadas pelas camadas subsequentes até a camada de saída na qual o resultado predito pelo modelo é comparado com o valor verdadeiro daquela amostra gerando o erro. Essa etapa é chamada de *forward* pois as informações são passadas na direção da entrada para a saída. Cabe destacar que em geral os neurônios de uma mesma camada possuem a mesma função de transferência mas isso pode variar de uma camada para outra, especialmente tratando-se da saída quando comparada com as camadas escondidas. Por sua vez na camada de entrada nenhuma operação é realizada, ela apenas indica a entrada de dados das camadas subsequentes. Durante o treinamento, após a etapa *forward* é necessário que o erro seja retro propagado para que ele seja minimizado, essa etapa é conhecida como *backward*.

A fase *backward* tem a finalidade de ajuste dos valores dos pesos da rede visando a minimização do erro na saída. Pra isso um procedimento de aprendizado é realizado utilizando-se um algoritmo de *backpropagation*. O objetivo é encontrar um conjunto de pesos que assegure que para cada vetor de entrada o vetor de saída produzido pela rede seja o mesmo (ou suficientemente próximo) ao vetor de saída desejado. Se há um conjunto fixo e finito de casos entrada-saída, o erro total de desempenho da rede com um particular conjunto de pesos pode ser calculado comparando os vetores de saídas reais e desejadas para cada caso. Dessa maneira, o erro total pode ser definido pela equação 3.16 (RUMELHART; HINTON; WILLIAMS, 1986).

$$E = \frac{1}{2} \sum_c \sum_j (y_{c,j} - d_{c,j})^2 \quad (3.16)$$

onde, c é um índice sobre os casos (pares entrada saídas), j é um índice sobre as unidades de saída, y é o estado real de uma unidade de saída e d é o estado desejado.

Diversos algoritmos podem ser usados como algoritmo de *backpropagation* para minimizar a equação 3.16. Quando esse modelo foi proposto, o algoritmo proposto foi o gradiente descendente. Para isso é necessário calcular a derivada parcial de E em relação a cada peso na rede. Isso é simplesmente a soma das derivadas parciais ara um dos casos entrada-saída.

A fase *backward* inicia computando $\partial E/\partial y$ para cada uma das unidades de saída. Derivar a equação 3.16 para um caso particular c e suprimindo o índice c fornece o que pode ser visto na equação]3.17.

$$\frac{\partial E}{\partial y_j} = y_j - d_j \quad (3.17)$$

Aplicando-se a regra da cadeia obtêm-se o que pode ser visto na equação 3.18.

$$\frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{dx_j} \quad (3.18)$$

Derivando a equação 3.15 para encontrar o valor de dy_j/dx_j e substituindo-o, obtêm-se o que pode se ver na equação 3.19.

$$\frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j} \cdot y_j(1 - y_j) \quad (3.19)$$

Isso quer dizer que é possível saber como uma mudança na entrada total x para uma unidade de saída irá afetar o erro. No entanto, essa entrada total é somente uma função linear dos estados das unidades de níveis inferiores e ela também é uma função linear dos pesos nas conexões, então é fácil de calcular como o erro será afetado pela mudança desses estados e pesos. Para um peso w_{ji} de i a j a derivada é dada pela equação 3.20.

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial x_j} \cdot \frac{\partial x_j}{\partial w_{ji}} = \frac{\partial E}{\partial x_j} \cdot y_i \quad (3.20)$$

e para a saída da i -ésima unidade a contribuição para $\partial E/\partial y_i$ resultando do efeito de i em j é simplesmente dada pela equação 3.21.

$$\frac{\partial E}{\partial x_j} \cdot \frac{\partial x_j}{\partial y_i} = \frac{\partial E}{\partial x_j} \cdot w_{ji} \quad (3.21)$$

Esse processo mostra como calcular dE/dy para qualquer unidade na penúltima camada quando dado dE/dy para todas unidades na última camada. Esse procedimento pode então ser repetido para calcular esse termo para camadas sucessivamente anteriores, realizando o cálculo de dE/dw para os pesos durante esse processo.

Uma maneira de usar dE/dw é mudar os pesos após cada caso de entrada-saída. Isso tem a vantagem de que não é necessária memória separada para as derivadas. Um esquema alternativo, que Rumelhart, Hinton e Williams (1986) usaram em seu artigo é acumular dE/dw

sobre todos os casos de entrada e saída antes de mudar os pesos. A versão mais simples do gradiente descendente é alterar cada peso por uma quantidade proporcional ao dE/dw acumulado como na equação 3.22.

$$\Delta w = -\varepsilon \frac{\partial E}{\partial w} \quad (3.22)$$

Esse método não converge tão rapidamente como métodos que usam as derivadas segunda, mas é muito mais simples e pode ser facilmente implementado por cálculos locais em hardware paralelo. Ele pode ser significativamente melhorado, sem sacrificar a simplicidade e localidade, usando um método de aceleração no qual o gradiente atual é usado para modificar a velocidade do ponto no espaço de pesos ao invés de sua posição, como mostra a equação 3.23.

$$\Delta w(t) = -\varepsilon \frac{\partial E}{\partial w(t)} + \alpha \Delta w(t-1) \quad (3.23)$$

onde, t é incrementado por 1 a cada varredura em todo o conjunto de casos de entrada-saída, e α representa um fator de decaimento exponencial entre 0 e 1 que determina a contribuição relativa do gradiente atual e gradientes anteriores à mudança do peso.

O processo de *backpropation* foi extraído do artigo original de Rumelhart, Hinton e Williams (1986) no qual maiores informações e exemplos podem ser encontrados. Cabe destacar que o método exposto possui algumas desvantagens:

- a) A função custo de erro a ser minimizada dada por 3.16 deve ser diferenciável;
- b) não é garantido que o gradiente descendente irá encontrar o mínimo global.

Durante os anos diversas outras técnicas e algoritmos de otimização foram pesquisados e utilizados de tal forma que as desvantagens citadas não representassem um empecilho. Decidiu-se por explicar sucintamente o processo usando o gradiente descendente por ele ter sido o primeiro usado nesse sentido e por ser mais simples de ser explicado.

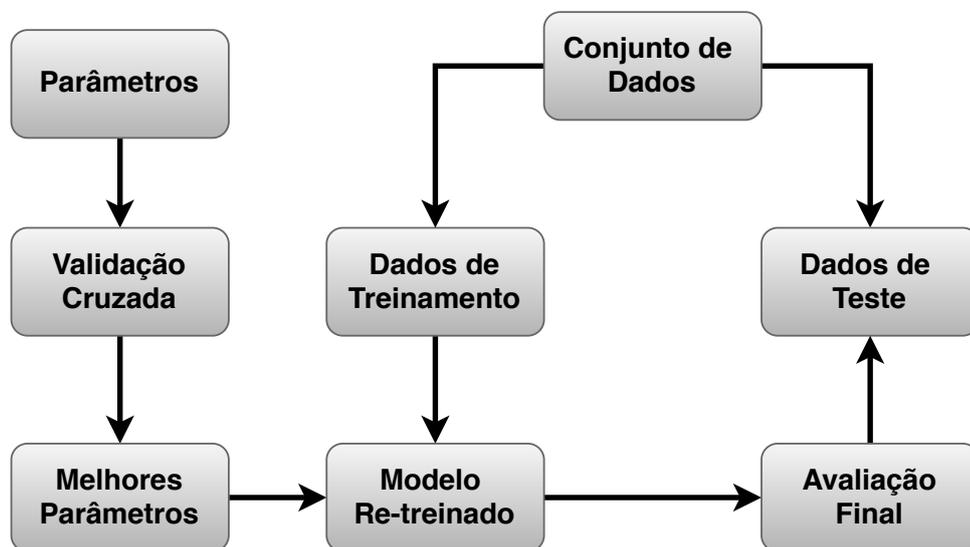
3.7 Validação Cruzada

Durantes o treinamento de modelos de inteligência computacional algumas técnicas podem ser utilizadas para verificar o comportamento e desempenho do modelo com dados de treino. A metodologia em geral mais utilizada para isso é a validação cruzada, ela permite avaliar a capacidade de generalização dos modelos preditivos a fim de evitar o sobre-ajuste (*overfitting*) dos dados (BERRAR, 2017).

Esse procedimento realiza a divisão sucessiva do conjunto de treinamento em duas partes, uma é usada para treinar o modelo e a outra para validá-lo. Essas duas porções são alternadas em rodadas sucessivas de tal maneira que todos dados tenham sido usados tanto para treinamento quanto para validação do modelo (REFAEILZADEH; TANG; LIU, 2009).

Ao lidar com problemas de aprendizado de máquina não se deve utilizar todos os dados disponíveis somente para treinamento, uma vez que isso não permitirá avaliar o desempenho do mesmo. Portanto, geralmente é adotada a divisão do conjunto disponível para treinamento e avaliação da rede. No entanto, muitos modelos lidam com aleatoriedades e fazer uma divisão simplória dos dados pode fazer com que o modelo tenha um bom ou mau desempenho a depender do acaso. Nessa situação, a validação cruzada se mostra extremamente eficaz uma vez que todos os dados são usados tanto para treinamento quando para avaliação e ao final é apresentada a média e desvio padrão da métrica de avaliação para todas essas rodadas sucessivas. Isso permite verificar de modo mais preciso qual o desempenho do modelo implementado. Em (SCIKIT-LEARN, 2019a) é apresentado um fluxograma de uma validação cruzada típica em treinamento. Ele foi traduzido e é apresentado na Figura 3.6.

Figura 3.6 – Fluxograma para validação cruzada.



Fonte: Adaptado de Scikit-learn (2019a).

Segundo Refaeilzadeh, Tang e Liu (2018) existem dois possíveis objetivos em validação cruzada:

- a) Realizar a estimativa do desempenho do modelo treinado com um algoritmo usando os dados disponíveis. Ou seja, avaliar a capacidade de generalização do modelo utilizado;

- b) realizar um comparativo entre dois ou mais algoritmos (modelos) visando encontrar aquele que possua melhor desempenho utilizando-se o conjunto de dados disponíveis.

A técnica de validação cruzada mais utilizada na prática chama-se *k-fold*, uma breve explicação de como ela funciona é apresentada da subseção a seguir.

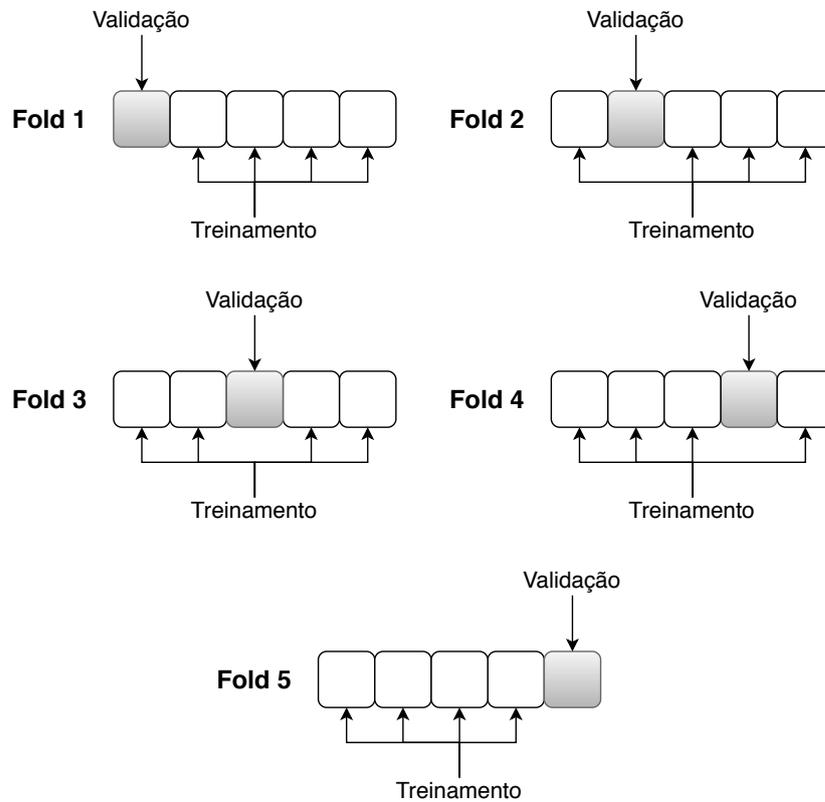
3.7.1 Validação *k-fold*

Ao utilizar a validação cruzada *k-fold* o conjunto de dados de treinamento é particionado em k subconjuntos de tamanhos aproximadamente iguais. Esse processo é realizado a partir da amostragem aleatória de pontos de dados do conjunto de treinamento sem reposição. O modelo é treinado com $k - 1$ subconjuntos, e portando, o subconjunto remanescente, chamado de conjunto de validação (ou teste, a depender do autor) é utilizado para avaliar o desempenho do modelo. Esse processo é repetido até que todos os subconjuntos tenham sido utilizados tanto para treinamento quanto para validação do modelo. Ao final, a média das k medidas de desempenho nos k conjuntos de validação representa a performance geral da validação cruzada do modelo em questão para o conjunto de dados utilizados (BERRAR, 2017).

A Figura 3.7 apresenta um exemplo de validação cruzada *5-fold*. nela é possível observar que o conjunto de dados é dividido em 5 partes iguais. No *fold 1*, o primeiro subconjunto é usado na validação enquanto os demais são utilizados no treinamento do modelo. No *fold 2*, o segundo subconjunto é usado na validação enquanto os outros são utilizados no treinamento. Esse processo é repetido até que todos os subconjuntos tenham sido utilizados tanto para treinamento quanto para validação. Considerando-se um modelo de classificação, por exemplo, a acurácia de validação cruzada nessa situação é a média de todas as cinco acurácias nos subconjuntos de validação.

Nesse método é importante também realizar o processo de estratificação dos dados após dividi-los nos *k-folds*. Isso consiste em rearranjar a divisão dos dados de tal maneira que se assegure que cada *fold* seja uma boa representação do todo, ou seja, garantir que as classes estejam balanceadas nos conjuntos de treino e teste. Por exemplo, em um problema de classificação binária onde cada classe compreende 50% dos dados, é melhor rearranjar os dados de tal maneira que em cada *fold* cada classe compreenda aproximadamente metade dos eventos (REFAEILZADEH; TANG; LIU, 2018)

Figura 3.7 – Validação cruzada 5-fold.



Fonte: Do autor (2020).

3.8 Matriz de Confusão

A matriz de confusão permite resumir a performance do modelo de classificação no conjunto de teste (ou validação a depender do autor). É uma matriz de duas dimensões, das quais uma representa as classes verdadeiras de um objeto e a outra as classes que o classificador atribuiu (TING, 2017). Dessa maneira, a diagonal principal representa todos os valores que o modelo acertou enquanto as demais células representam classes as quais foram erroneamente preditas.

Um exemplo de matriz de confusão pode ser visto no Quadro 3.1. Nele os dados estão balanceados contemplando 200 amostras distribuídas em 4 classes contendo 50 observações em cada. Assim, observando-se a diagonal principal é possível verificar que esse modelo hipotético é capaz de acertar 30 amostras da Classe 0, 25 da Classe 1, 27 da Classe 2 e 32 da classe 3. Além disso, é possível inferir também qual classe o modelo tem maior dificuldade para classificar. Para isso basta somar os valores das linhas excluindo-se a diagonal principal, assim verifica-se que o modelo errou 20 amostras da Classe 0, 25 da Classe 1, 23 da Classe 2 e 18 da Classe

3. Portanto, conclui-se que esse modelo hipotético apresenta pior resultado na classificação de amostras da Classe 1.

Quadro 3.1 – Matriz de confusão para 4 classes

<i>Classes reais</i>	<i>Classes atribuídas pelo classificador</i>			
	Classe 0	Classe 1	Classe 2	Classe 3
Classe 0	30	5	8	7
Classe 1	10	25	8	7
Classe 2	6	8	27	9
Classe 3	1	12	5	32

Fonte: Do autor (2020).

Algumas métricas de avaliação global do modelo podem ser extraídas da matriz de confusão, a principal delas e amplamente utilizada em problemas de classificação é a acurácia. Ela pode ser definida como a razão entre a quantidade de dados corretamente classificados pelo número total de itens (Junker; Hoch; Dengel, 1999). No exemplo apresentado no Quadro 3.1, por exemplo, ela poderia ser calculada utilizando a equação 3.24.

$$\text{acurácia} = \frac{\text{Predito pelo modelo corretamente}}{\text{Total de observações}} = \frac{30 + 25 + 27 + 32}{200} = 0,57 = 57\% \quad (3.24)$$

Outra métrica que pode ser extraída da matriz de confusão é a precisão (calculada pela equação 3.26.) Considerando-se por exemplo, uma classificação binária com duas classes: positivo ou negativo, ela calcula dentre todas as classificações de classe positivo que o modelo fez, quantas estão corretas. Uma métrica também utilizada em problemas de classificação é o *recall* ou revocação (calculada pela equação 3.27), ela calcula dentre todas as situações da classe positivo como valor esperado, quantas estão corretas. Ainda é possível calcular a média harmônica entre precisão e *recall* comumente chamada de f1-score e calculada pela equação 3.28. O Quadro 3.2 apresenta uma matriz de confusão mais simples, com apenas duas classes, para exemplificar como essas métricas são extraídas.

Quadro 3.2 – Matriz de confusão para 2 classes

Real	Predito	
	Positivo	Negativo
Positivo	Verdadeiro positivo (VP)	Falso Negativo (FN)
Negativo	False positivo (FP)	Verdadeiro negativo (VN)

Em resumo, as métricas citadas que podem ser calculadas a partir da matriz de confusão são:

$$acurácia = \frac{VP + VN}{VP + VN + FP + FN} \quad (3.25)$$

$$precisão = \frac{VP}{VP + FP} \quad (3.26)$$

$$recall = \frac{VP}{VP + FN} \quad (3.27)$$

$$f1-score = 2 \cdot \frac{precisão \cdot recall}{precisão + recall} \quad (3.28)$$

Matrizes de confusão são ferramentas especialmente interessantes para se averiguar o comportamento do modelo individualmente para cada classe, isso permite adotar estratégias distintas ou a adoção de outras métricas para verificação do desempenho do modelo.

3.9 t-Distributed Stochastic Neighbor Embedding

Existem variadas técnicas para visualização de dados com muitas dimensões, um método bem conhecido é o *t-Distributed Stochastic Neighbor Embedding* (t-SNE) criado por Maaten e Hinton (2008), ele permite visualizar dados de alta dimensão dando a cada ponto uma localização em um mapa de duas ou três dimensões. Essa técnica é uma variação da *Stochastic Neighbor Embedding* (SNE) (HINTON; ROWEIS, 2003) que é mais fácil de otimizar e produz melhores visualizações reduzindo a tendência de aglomerar pontos no centro do mapa. t-SNE é melhor do que as técnicas existentes na criação de um único mapa que revela a estrutura em muitas escalas diferentes.

Embora o SNE construa visualizações razoavelmente boas, ele é dificultado por uma função de custo que é difícil de otimizar e por um problema que foi chamado por Maaten e Hinton (2008) de "problema de aglomeração". O t-SNE usa uma função custo que se difere do método antecessor de duas maneiras: (1) usa uma versão simétrica da função custo do SNE com gradientes mais simples e (2) usa uma distribuição t de student em vez de uma Gaussiana para computar a similaridade entre os pontos no espaço de baixa dimensão. t-SNE emprega uma dis-

tribuição de cauda longa no espaço de baixa dimensão para aliviar o problema de aglomeração e os problemas de otimização do SNE.

O SNE inicia convertendo as distâncias Euclidianas de alta dimensão entre os pontos em probabilidades condicionais que representam similaridades. A similaridade do ponto x_j para o ponto x_i é a probabilidade condicional, $p_{j|i}$, que x_i escolheria x_j como seu vizinho se os vizinhos fossem escolhidos em proporção à sua densidade de probabilidade sob uma Gaussiana centrada em x_i . Para pontos próximos, $p_{j|i}$ é relativamente alto, considerando que para pontos de dados amplamente separados, $p_{j|i}$ será quase infinitesimal (para valores razoáveis da variância da Gaussiana, σ_i^2). Matematicamente, a probabilidade condicional é dada pela equação 3.29 (MAATEN; HINTON, 2008).

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)} \quad (3.29)$$

onde, σ_i^2 é a variância da Gaussiana que é centrada no ponto x_i . Como o interesse está em modelar semelhanças entre pares o valor de $p_{i|i}$ é setado em zero. Para as contraparte de baixa dimensão y_i e y_j dos pontos de alta dimensão x_i e x_j , é possível computar um probabilidade condicional similar que é denotada por $q_{j|i}$. Os autores (MAATEN; HINTON, 2008) setaram a variância da Gaussiana que é empregada no cálculo das probabilidades condicionais $q_{j|i}$ em $\frac{1}{\sqrt{2}}$. Portanto, o modelo de similaridade que mapeia o ponto y_j para o ponto y_i é dado por:

$$q_{j|i} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq i} \exp\left(-\|y_i - y_k\|^2\right)} \quad (3.30)$$

Novamente, já que o interesse é somente em modelar similaridades entre pares: $q_{i|i} = 0$.

Não há um único valor de σ_i^2 que seja ótimo para todos os pontos no conjunto de dados porque a densidade dos dados provavelmente varia. Em regiões densas um valor menor de σ_i^2 é mais apropriado do que em regiões esparsas. Qualquer particular valor de σ_i^2 leva a uma distribuição de probabilidade P_i sobre todos os outros pontos. SNE conduz uma busca binária para o valor de σ_i^2 que produz uma distribuição de probabilidade com uma perplexidade fixa que é especificada pelo usuário. A perplexidade é definida como:

$$\text{Perp}(P_i) = 2^{H(P_i)} \quad (3.31)$$

onde $H(P_i)$ é a entropia Shannon de P_i medida em bits dada por:

$$H(P_i) = - \sum_j P_{j|i} \log_2 P_{j|i} \quad (3.32)$$

A perplexidade pode ser interpretada com uma medida suave do número efetivo de vizinhos. O desempenho do SNE é bastante robusto para mudanças na perplexidade, e valores típicos variam entre 5 e 50.

No final, SNE visa minimizar a soma das divergências Kullback-Leibler sobre todos os pontos de dados usando um método gradiente descendente. A função custo é dada pela equação 3.33.

$$C = \sum_i KL(P_i || Q_i) = \sum_i \sum_j P_{j|i} \log \frac{P_{j|i}}{Q_{j|i}} \quad (3.33)$$

onde, P_i representa a distribuição de probabilidade condicional sobre todos outros pontos dado o ponto x_i , e Q_i representa a distribuição de probabilidade condicional sobre todos os outros pontos mapeados dado o ponto mapeado y_i .

Tendo em vista que as probabilidades condicionais definidas nas equações 3.29 e 3.30 não são simétricas, ou seja, $p_{j|i} \neq p_{i|j}$ e $q_{j|i} \neq q_{i|j}$, a função custo dada por 3.33 é difícil de ser otimizada. É possível utilizar distribuições conjuntas apresentadas a seguir em vez das probabilidades condicionais para resolver o problema de simetria da equação 3.33.

$$\begin{aligned} p_{ij} &= \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_k \sum_{l \neq k} \exp\left(-\|x_k - x_l\|^2 / 2\sigma_i^2\right)}, & p_{ii} &= 0 \\ q_{ij} &= \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_k \sum_{l \neq k} \exp\left(-\|y_k - y_l\|^2\right)}, & q_{ii} &= 0 \end{aligned} \quad (3.34)$$

No entanto, as equações apresentadas em 3.34 não resolvem o problema da complexidade da função custo 3.33 já que são sensíveis a *outliers*. Para contornar esse problema as probabilidades conjuntas p_{ij} são definidas no espaço de alta dimensão para serem probabilidades condicionais simetrizadas, isto é, $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$. Isso garante que $\sum_j p_{ij} > \frac{1}{2n}$ para todos pontos x_i , como resultado cada ponto x_i faz uma contribuição significativa à função custo. Apesar disso reduzir a complexidade computacional e acelerar o processo de aprendizado, ainda não resolve o "problema de aglomeração".

O problema de aglomeração surge porque não há espaço suficiente quando ocorre a migração de altas para baixas dimensões. À medida que as dimensões encolhem, torna-se cada

vez mais difícil manter os pontos moderadamente distantes uns dos outros e, ao mesmo tempo, garantir que residam perto de seus vizinhos mais próximos. Como uma consequência pontos em uma distância moderada de x_i são colocados muito longe, se isso ocorrer para cada ponto, então o sistema eventualmente irá colapsar no centro do espaço mapeado (ANDERSSON, 2021). O método t-SNE surgiu para sanar esses problemas, as probabilidades conjuntas nessa técnicas são dadas por:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}, \quad p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_k \sum_{l \neq k} \exp\left(-\|x_k - x_l\|^2 / 2\sigma_i^2\right)}, \quad p_{i|i} = 0$$

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_k \sum_{l \neq k} \left(1 + \|y_k - y_l\|^2\right)^{-1}}, \quad q_{ii} = 0 \quad (3.35)$$

Pelas equações em 3.35 é possível observar que p_{ij} é igual a forma calculada para SNE simétrica, mas a expressão para q_{ij} foi alterada, e embora semelhante ao SNE, as modificações são imperativas para a superioridade do t-SNE. A nova forma de q_{ii} também está relacionada a distribuição t de Student com um grau de liberdade também chamada de distribuição Cauchy, cuja a função densidade de probabilidade (*probability density function* - pdf) é dada pela equação 3.36.

$$p(x) = \frac{1}{\pi\gamma} \left[1 + \left(\frac{x - x_0}{\gamma} \right)^2 \right]^{-1}, \quad \gamma > 0 \quad (3.36)$$

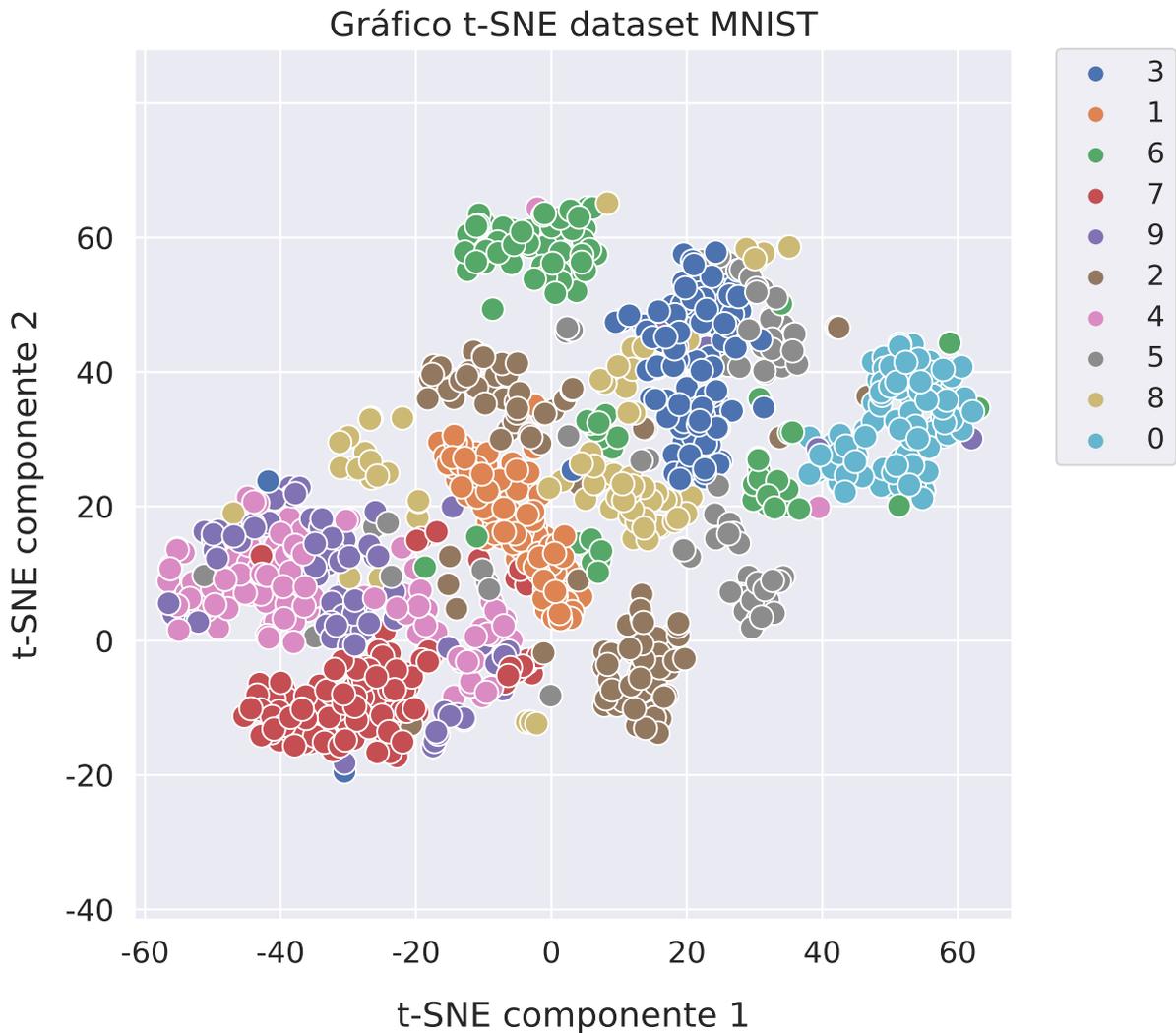
onde, x_0 é o parâmetro de localização que especifica a localização do pico da distribuição, e γ é o parâmetro que controla a escala da distribuição.

A distribuição de Cauchy tem caudas mais longas do que a distribuição gaussiana, que se acomoda melhor a valores atípicos; isso nos permite colocar pontos com distâncias moderadas no espaço de alta dimensão, muito longe, sem ter que nos preocupar com o colapso de nosso sistema. Convenientemente, também eliminamos as exponenciais na expressão q_{ij} , o que reduz o custo computacional (ANDERSSON, 2021).

Na Figura 3.8 é apresentado o resultado da aplicação do método t-SNE no conjunto de dados MNIST. Esse *dataset* é composto por 1000 imagens de dígitos escritos a mão nas mais variadas caligrafias. Cada classe representa um dígito variando entre 0 e 9. Na imagem podem

ser verificados os *clusters* formados e é possível identificar aqueles dígitos que um algoritmo de *machine learning* teria maior dificuldade em classificar por ter grande similaridade a outro.

Figura 3.8 – t-SNE aplicado ao *dataset* MNIST



Fonte: Do autor (2021).

Por fim, nesse capítulo, foram apresentadas algumas técnicas usadas durante a condução do projeto. O objetivo foi demonstrar as principais características delas e as formas de serem usadas na prática. Durante a condução do trabalho descobriu-se o efetivo uso do *Rocket* e *MiniRocket* na extração de características em séries temporais. Além disso, o t-SNE se mostrou muito eficaz em demonstrar o comportamento das variáveis e identificar as causas dos classificadores não performarem tão bem em algumas situações enquanto performam melhor em outras. Assim, cada uma das técnicas apresentadas foram importantes na condução desse

trabalho, seja para identificar as faltas, metrificar o desempenho dos modelos ou investigar as possíveis causas de uma baixa performance.

4 MATERIAIS E MÉTODOS

Nesse capítulo são apresentados os materiais, métodos e metodologias que foram utilizados na completa realização desse trabalho. São incluídas informações sobre os bancos de dados usados, ferramentas computacionais utilizadas, as etapas que foram conduzidas, os métodos implementados, entre outros tópicos.

4.1 Bases de dados

Nesse projeto foram utilizadas duas bases de dados distintas com variações de diversos parâmetros. A diferença básica entre as duas é que a primeira trata-se da simulação de uma linha de transmissão mais simples enquanto a segunda inclui um banco de capacitores em série para simular uma linha de transmissão com compensação.

4.1.1 Primeiro conjunto de dados

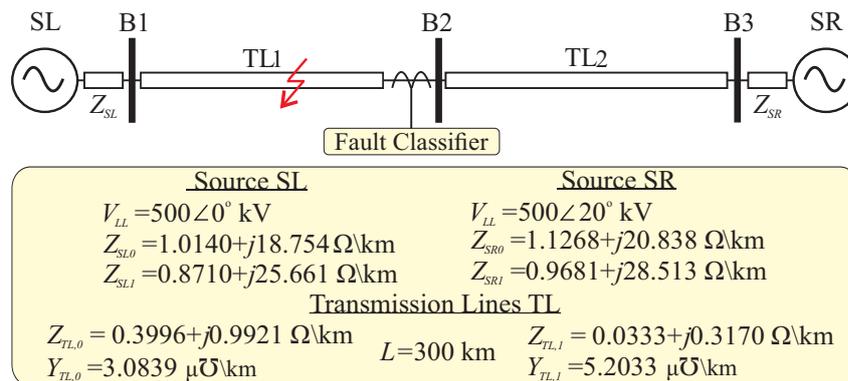
O primeiro banco de dados utilizado nesse projeto foi cedido pelo professor Flávio Bezerra Costa da Universidade Federal do Rio Grande do Norte (UFRN) e é baseado no trabalho de Costa, Souza e Brito (2010). Ele é composto por 940 arquivos que representam falhas ocorridas em linhas de transmissão com diversas variações de parâmetro. Os dados foram gerados a partir de simulações no *software* Matlab/Simulink, com as seguintes características:

- a) Tensão: 500 kV;
- b) extensão da linha: 300 km;
- c) tipo de falta: AT, BT, CT, AB, BC, CA, ABT, BCT, CAT e ABC;
- d) 94 amostras por tipo de falta;
- 3) distância na qual a falta ocorreu em relação ao barramento local: 20 km, 150 km e 280 km;
- f) ângulo de aplicação da falta: varia entre 0° e 180° com passo de 10° ;
- g) resistência da ocorrência da falta: 1Ω , 50Ω e 100Ω ;
- h) frequência fundamental: 60 Hz;
- i) amostras por ciclo: 256;

- j) frequência de amostragem: 15 360 Hz;
- k) os sinais originalmente tem tamanho de 3842 pontos amostrais.

O diagrama unifilar do sistema é apresentado na Figura 4.1 onde os parâmetros principais da linha de transmissão e as fontes são descritas. O comprimento total da linha é 300 km ($L_1 + L_2 = 300 \text{ km}$).

Figura 4.1 – Diagrama unifilar para o sistema de energia trifásico de 500 kV.

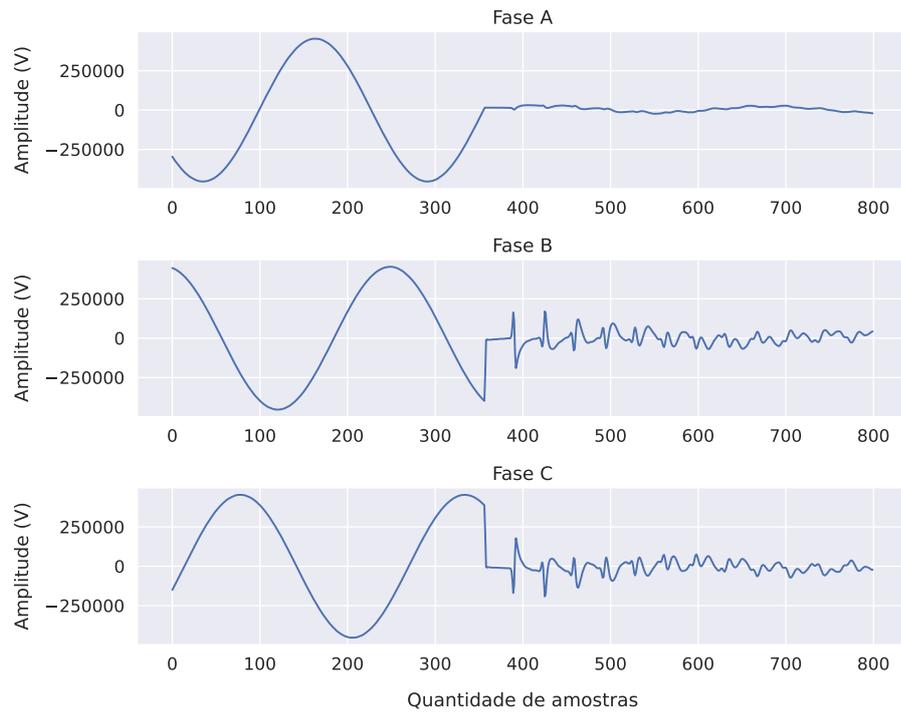


Fonte: Costa, Souza e Brito (2010).

Nesse trabalho focou-se exclusivamente na classificação das faltas, por esse motivo, utilizou-se os sinais após a detecção da falta não se dando enfoque na forma que a mesma foi realizada. Tendo em vista isso, foi incorporado um ruído branco gaussiano aos sinais de falta após a detecção. Muitos pesquisadores utilizam esse tipo de ruído em seus estudos e em vários artigos é possível verificar que o nível do ruído utilizado varia entre 75 e 20 dB (MUKHERJEE; KUNDU; DAS, 2021). Por fim, decidiu-se utilizar a relação sinal-ruído (*signal-noise-ratio* - SNR) de 60 dB nos sinais apresentados nessa dissertação.

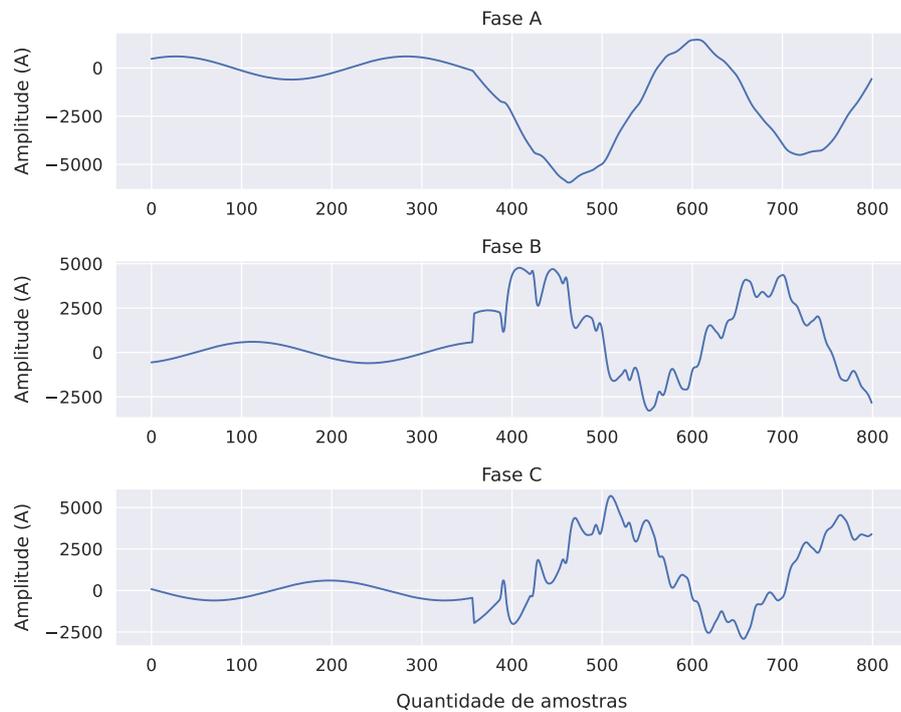
Em geral pode ser observada uma diminuição no nível de tensão ou aumento dos valores de corrente quando ocorre uma falha na linha de transmissão. No entanto, a depender das condições do sistema, esse comportamento pode variar. Contudo, isso pode ser visto para o primeiro conjunto de dados na Figura 4.2 e Figura 4.3 que apresentam respectivamente o comportamento da tensão e corrente para uma falha ABC que ocorreu a 20km do barramento local com resistência de falta de $1 \ \Omega$ e ângulo de incidência de 0° .

Figura 4.2 – Comportamento da tensão em uma falta ABC 20km 0° 1 Ω.



Fonte: Do autor (2021).

Figura 4.3 – Comportamento da corrente em uma falta ABC 20km 0° 1 Ω.



Fonte: Do autor (2021).

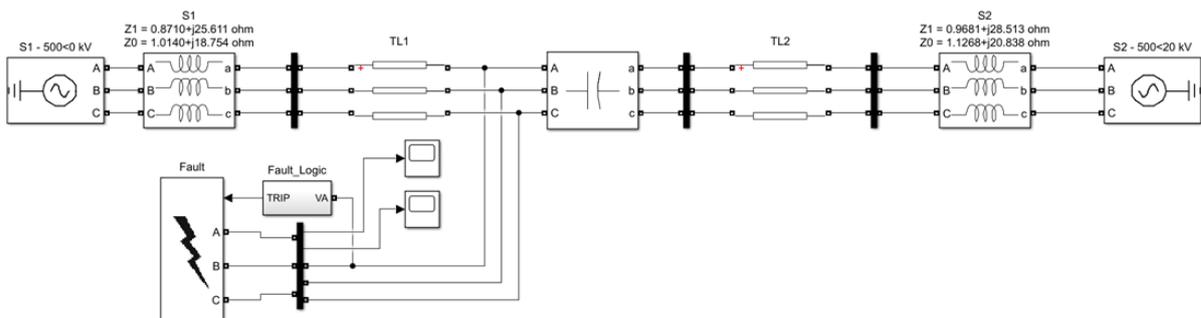
Em alguns momentos do texto da dissertação pode se fazer necessário referenciar especificamente esse conjunto de dados, portanto ele será tratado como *dataset 1*.

4.1.2 Segundo conjunto de dados

O segundo banco de dados utilizado apresenta as mesmas características mostradas no início dessa sessão para o primeiro *dataset*, ou seja, mesma tensão, extensão da linha, tipos de falta, distâncias da ocorrência, ângulo de incidência, resistência, frequência fundamental e de amostragem. A diferença dele para o primeiro reside no fato de ter sido incluído um banco de capacitores conectados em série na linha. Devido essa variação, para esse segundo *dataset* é gerada uma quantidade bem maior de dados totalizando 6840 amostras, ou seja, 684 para cada tipo de falta. Para efeitos de referência esse banco de dados será referenciado como *dataset 2*.

Esse conjunto de dados foi simulado também se utilizando o *software* Matlab/Simulink, e o diagrama do sistema pode ser verificado na Figura 4.4. As tensões das fontes S1 e S2 são $V_{S1} = 500\angle 0^\circ$ kV e $V_{S2} = 500\angle 20^\circ$ kV, respectivamente. Os parâmetros da linha de transmissão são: $L = 300$ km, $Z_1 = 0.0333 + j0.3170 \Omega/\text{km}$, $Y_1 = 5.2033 \mu\text{S}/\text{km}$, $Z_0 = 0.3996 + j0.9921 \Omega/\text{km}$ e $Y_0 = 3.0839 \mu\text{S}/\text{km}$.

Figura 4.4 – Sistema de linha de transmissão com compensação em série.



Fonte: Fonseca et al. (2021).

Segundo Circuit Globe (2021), por definição, a compensação em série é um método para melhorar os sistemas elétricos conectando um capacitor em série com a linha de transmissão. Ou seja, a potência reativa é inserida em série com a linha para melhorar a impedância do sistema. Isso melhora a capacidade de transferência de potência, sendo usada principalmente em linhas de tensão extra e ultra-alta.

A compensação em série tem várias vantagens, como aumentar a capacidade de transferência de potência, melhorar a estabilidade do sistema, controlar a regulação da tensão e garantir

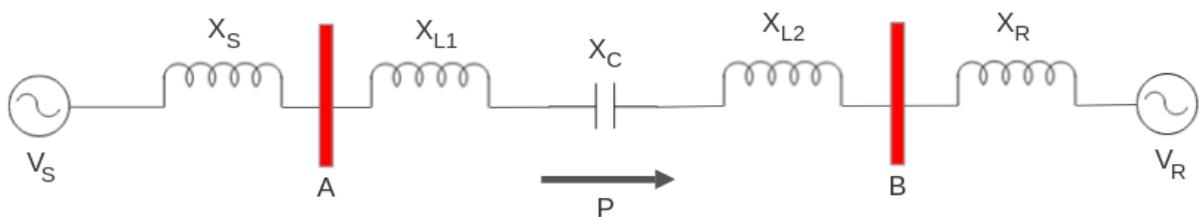
a divisão de carga adequada entre os alimentadores paralelos. A transferência de potência ativa sobre uma linha é dada pela equação 4.1.

$$P_1 = \frac{V_S V_R}{X_L} \sin \delta \quad (4.1)$$

onde P_1 é a potência ativa transferida por fase em Watts, V_S a tensão final de envio por fase (V), V_R a tensão final de recepção por fase, X_L a reatância indutiva em série da linha e δ o ângulo de fase entre V_S e V_R .

Se um capacitor com reatância de capacitância X_C for conectado em série com a linha, a reatância da linha é reduzida de X_L para $X_L - X_C$, sendo $X_L = X_{L1} + X_{L2}$. O sistema compensado em série é geralmente representado pela Figura 4.5 e a transferência de potência é dada pela equação 4.2.

Figura 4.5 – Diagrama de um sistema compensado em série.



Fonte: Circuit Globe (2021)

A potência ativa após a compensação da linha com banco de capacitores pode ser calculada utilizando-se a equação 4.2.

$$P_2 = \frac{V_S V_R}{X_L - X_C} \sin \delta \quad (4.2)$$

A relação entre as potências P_1 e P_2 é dada pela equação 4.3.

$$\frac{P_1}{P_2} = \frac{X_L}{X_L - X_C} = \frac{1}{1 - \frac{X_C}{X_L}} = \frac{1}{1 - k} \quad (4.3)$$

onde, $k = \frac{X_C}{X_L}$

O fator k é conhecido como grau de compensação ou fator de compensação. A compensação por unidade é dada em percentual como mostra a equação 4.4.

$$k = \frac{X_C}{X_L} pu \quad (4.4)$$

onde, X_L é a reatância indutiva total da linha por fase e X_C a reatância capacitiva do banco de capacitores por fase. Na prática o valor de k varia entre 0,4 e 0,7, esses portanto, foram os valores utilizados na geração do banco de dados. Usando como exemplo $k = 0,5$, a transferência de potência ativa é duplicada por uma compensação de 50%, como mostrado a seguir:

$$\frac{P_1}{P_2} = \frac{1}{1-k} = \frac{1}{1-0,5} = 2 \quad (4.5)$$

Além do aumento na capacidade de transferência de potência citado anteriormente as demais vantagens em se utilizar linhas com compensação são:

- a) Melhora da estabilidade do sistema: considerando a mesma transferência de potência e o mesmo valor de envio e recebimento de tensão, ocorre uma redução do valor do ângulo de fase δ para a linha com compensação em relação a sem compensação, isso confere uma maior estabilidade para o sistema;
- b) Divisão de carga entre linhas paralelas: quando uma nova linha com grande capacidade de transferência de energia está localizada em paralelo a uma linha já existente, o carregamento da nova linha pode levar ao sobrecarregamento da antiga. Utilizando a compensação em série, a reatância em série é reduzida e a divisão de carga entre o circuito paralelo pode ser feita facilmente de maneira adequada. Isso aumenta a capacidade de transferência de energia e reduz as perdas do sistema;
- c) Controle de Tensão: quando existe a variação repentina de carga no sistema, ocorrem quedas nos níveis de tensão. Isso é corrigido automaticamente em um sistema compensado pois ocorre uma mudança automática no V_{AR} (potência reativa) com a mudança na corrente de carga.

Por sua vez, a classificação de faltas em uma linha de transmissão com compensação em série é considerada mais difícil devido aos seguintes problemas (PARIKH; DAS; MAHESHWARI, 2010):

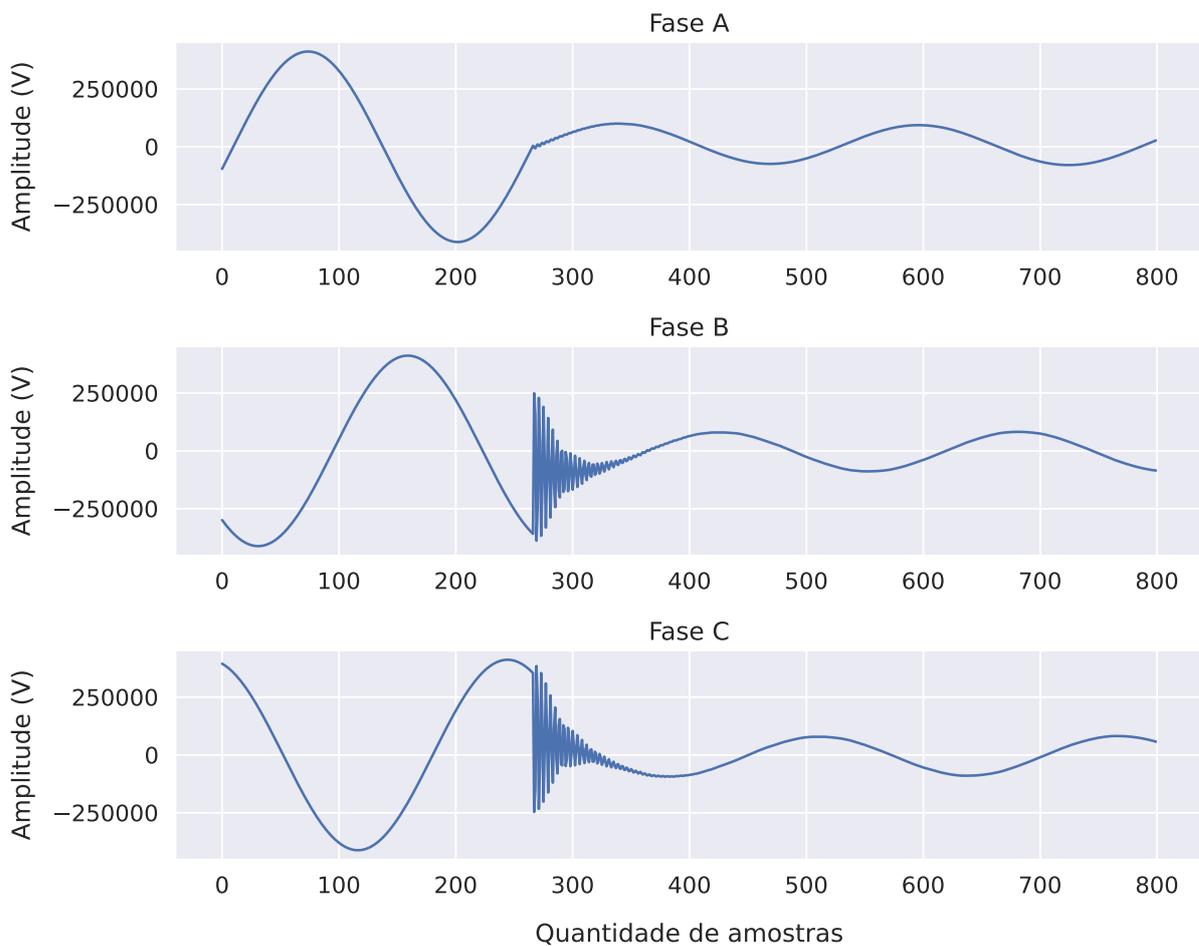
- a) A corrente de estado estacionário aumenta significativamente com a compensação em série e pode ser maior que a corrente de falta fase-terra em direção ao limite da linha;
- b) Em um arranjo típico de compensação em série, o varistor de óxido metálico (*metal oxide varistor* - MOV) é usado para proteger o capacitor de sobretensões durante uma

falta. No entanto, atua de forma não linear durante faltas e aumenta a complexidade do problema de proteção;

- c) Inversões de tensão e corrente;
- d) Os sinais de tensão e corrente produzidos na linha de transmissão contêm diferentes componentes de frequência. Portanto, esses sinais não são processados com muita precisão com métodos convencionais, como DFT de ciclo completo/DFT de meio ciclo (FCDFT/HCDFT) ou técnica de erro mínimo quadrado (*least square error* - LSE) e, portanto, grandes erros são frequentemente incorridos na estimativa dos fasores fundamentais.

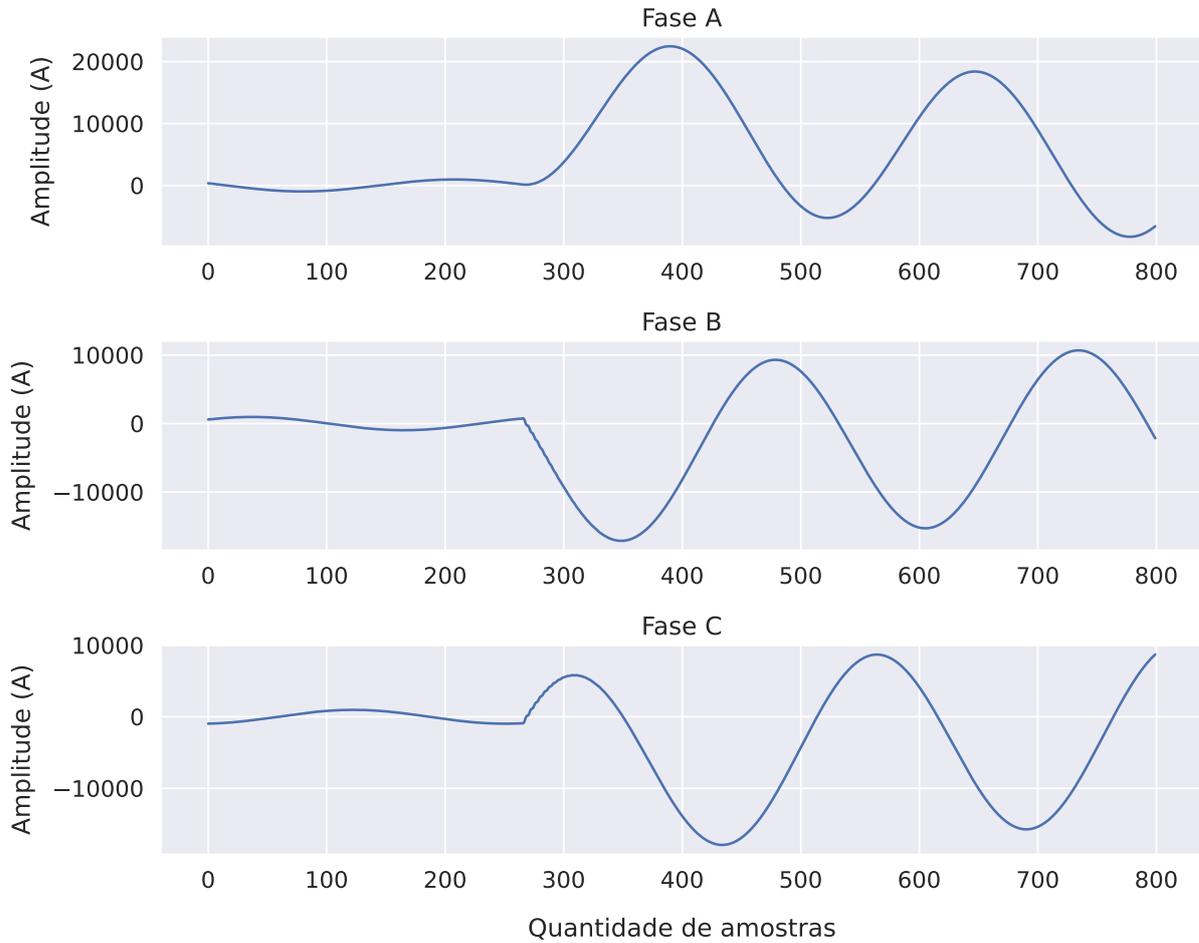
Os sinais de tensão e corrente gerados por uma falta ABC que ocorreu a 20 km do barramento, com resistência de falta de 1Ω , ângulo de incidência de 0° e compensação de 0,7 podem ser vistos respectivamente na Figura 4.6 e Figura 4.7.

Figura 4.6 – Comportamento da tensão em uma falta ABC 20 km 0° 1Ω 0,7 de compensação.



Fonte: Do autor (2021).

Figura 4.7 – Comportamento da corrente em uma falta ABC 20 km 0° 1Ω 0,7 de compensação.



Fonte: Do autor (2021).

4.2 Ferramentas utilizadas

Conforme citado na seção anterior, o *Matlab/Simulink* foi utilizado na geração dos sinais simulados. Além disso, *Python* e algumas de suas bibliotecas foram usadas no treinamento dos modelos de inteligência computacional e métricas de avaliação e validação. Cabe destacar principalmente as bibliotecas *Scikit-learn* (PEDREGOSA et al., 2011) e *sktime* (LÖNING et al., 2019). Além dessas bibliotecas, para implementação dos classificadores baseados em redes neurais utilizou-se o *Keras* (CHOLLET et al., 2015) que é uma API (*application programming interface*) de alto nível para implementação de redes neurais em *Python*. Os testes foram realizados em um computador com 8 Gb de memória RAM, processador i5 de oitava geração e no sistema operacional Manjaro.

4.3 Metodologia

Nesse trabalho foram propostos diversos métodos para lidar com o problema de classificação de faltas em linhas de transmissão. O objetivo disso é comparar esses métodos em termos de desempenho e complexidade computacional. Observando-se os trabalhos desenvolvidos na área é possível verificar que alguns autores utilizam sinais de corrente enquanto outros utilizam de tensão e há trabalhos que usam a combinação dos dois. Para efeitos de comparação entre o desempenho da utilização de um tipo de sinal em relação a outro, decidiu-se por aplicar os sinais de tensão como entrada para alguns métodos, enquanto os sinais de corrente foram utilizados em outros. Isso além de permitir a avaliação individual de cada modelo com o tipo de sinal também possibilita ter uma menor complexidade computacional caso os dois sinais fossem usados conjuntamente.

Antes de aprofundar em cada um dos métodos implementados é importante comentar alguns pontos relacionados aos conjuntos de dados usados. Em projetos de aprendizado de máquina os *datasets* utilizados influenciam diretamente na qualidade do modelo a ser treinado. É possível que modelos treinados para um conjunto não performem bem quando avaliados em outro conjunto que foi produzido de maneira distinta e que possua características diferentes.

Assim, os métodos A, B, D e F apresentados a seguir foram treinados e validados utilizando somente o *dataset 1* apresentado na subseção 4.1.1, enquanto os métodos C, E e G foram treinados com um conjunto de dados proveniente da combinação entre o *dataset 1* e o *dataset 2* apresentado na subseção 4.1.2. Isso gerou alguns resultados distintos que são apresentados de forma mais detalhada no Capítulo 5.

Ainda cabe destacar que para os métodos B, C, D, E, F e G os conjuntos de treinamento, validação e cada um dos *folds* foram salvos no disco rígido do computador. Isso garante que os exatos mesmos conjuntos sejam usados tanto no treinamento, quanto teste e validação dos modelos utilizados, permitindo que a comparação entre eles se dê de modo justo. Isso não foi realizado no método A, uma vez que ele foi implementado em um momento anterior do projeto, nesse caso visando evitar aleatoriedade a *loop* de validação cruzada foi repetida dez vezes.

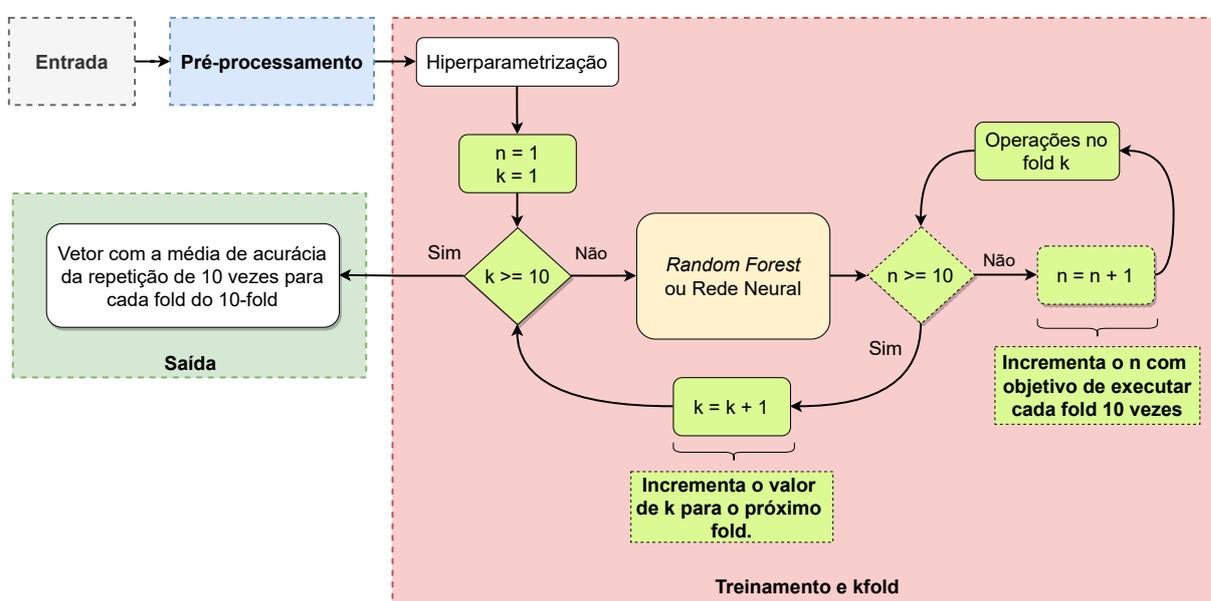
4.3.1 Avaliação dos modelos

Para se avaliar o desempenho dos classificadores, o banco de dados foi dividido da seguinte forma 80% treino, 10% teste e 10% validação. Foi utilizada nesse projeto a validação

cruzada k -fold estratificada a qual cria os *folders* de tal maneira que o percentual de amostras é preservado para cada classe.

Como o método A foi o primeiro a ser implementado uma metodologia de avaliação diferente foi adotada nele em relação aos demais. Maiores detalhes podem ser observados no artigo no qual ele foi publicado (FONSECA et al., 2021). Basicamente utilizou-se a validação cruzada k fold onde $k = 10$. Além disso, nesse caso cada um dos folds foi repetido 10 vezes visando eliminar possíveis aleatoriedades dos processos e modelos envolvidos. O fluxo de validação adotado nesse caso pode ser observado na Figura 4.8.

Figura 4.8 – Fluxograma da validação cruzada para o método A.



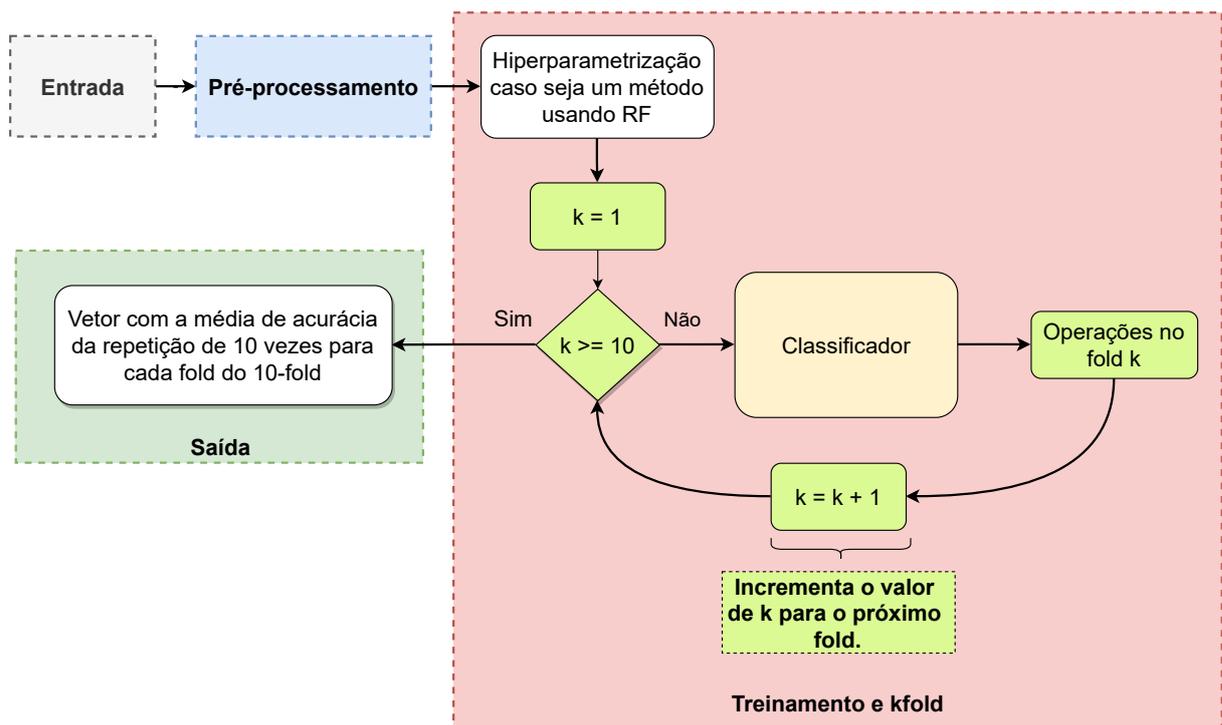
Fonte: Do autor (2021).

Conforme foi mencionado no método A, cada *fold* foi repetido dez vezes devido a complexidade computacional da rede neural implementada. Da mesma maneira decidiu-se por usar esse número de repetições também para a validação da *random forest* usada. Os resultados obtidos com o treinamento desse método podem ser verificados no Capítulo 5.

Para os demais métodos a abordagem na validação cruzada foi um pouco diferente. Nelas, a divisão entre dados de treinamento e validação foi realizada anteriormente na proporção de 80% e 20%, respectivamente. Além disso, o conjunto de treinamento foi previamente dividido utilizando o k -fold estratificado. Isso foi feito para garantir que todos os métodos utilizados fossem treinados e testados com os mesmos conjuntos de dados.

Como apresentado na seção 4.1 dois conjuntos de dados distintos foram utilizados nessa dissertação. Já na seção 4.3 foi apresentada a forma de utilização desses dados nos métodos implementados. Resumidamente, quando utilizado o *dataset 1* a quantidade de amostras no conjunto de treinamento foi de 752 enquanto no conjunto de validação foi 188 amostras. Por sua vez, quando utilizado também o *dataset 2*, é realizada uma reamostragem tendo em vista que esse conjunto possui um número muito superior de amostras, e nesse caso os dois *datasets* são usados em conjunto fornecendo 1504 amostras para treinamento e 376 amostras para validação dos modelos. Para esses métodos o fluxograma da validação cruzada é apresentado na Figura 4.9, deve-se destacar que a hiperparametrização foi usada somente quando o modelo de florestas aleatórias foi utilizado. Para os métodos que usam *MiniRocket* e *Rocket* como apresentado nas subseções 4.3.3.3 e 4.3.3.4 foram gerados diversos modelos a partir da variação da quantidade de *features* e *kernels* respectivamente.

Figura 4.9 – Fluxograma da validação cruzada para os métodos B, C, D, E, F e G.



Fonte: Do autor (2021).

Segundo a divisão dos dados adotada para esse projeto, a maior parcela foi utilizada durante o treinamento de cada uma das técnicas. Esse conjunto também foi utilizado na validação cruzada passando por divisões sucessivas onde cada divisão é usada no treino e no teste dos modelos como explicado na seção 3.7.1. Por sua vez, os dados de validação separados *a*

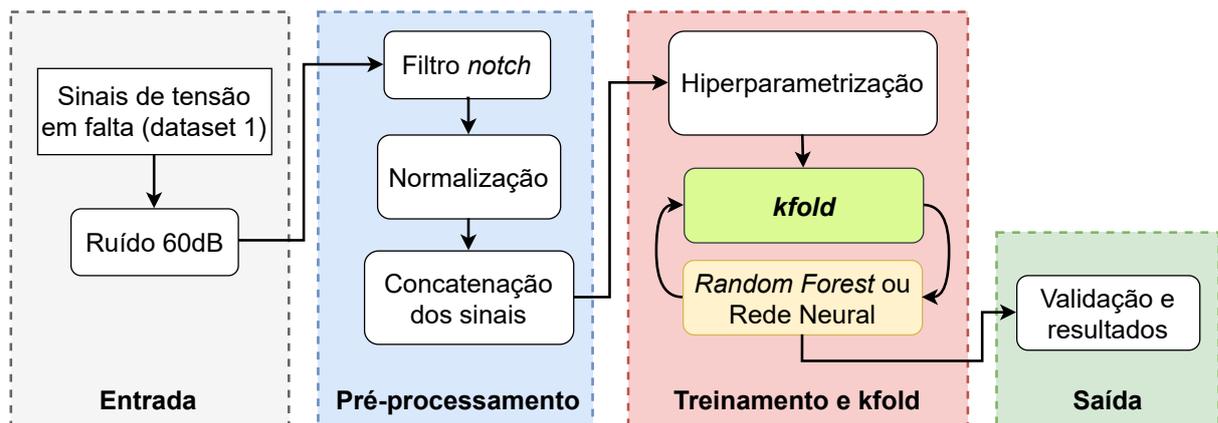
priori representam um conjunto não visto pelos modelos e foram utilizados para averiguar o desempenho dos métodos quando usado em um conjunto de dados desconhecido.

4.3.2 Métodos treinados usando sinais de tensão

4.3.2.1 Método A

Inicialmente um ruído de 60 dB foi aplicado aos sinais de tensão do *dataset 1*, e o filtro *notch* foi utilizado para filtrar a componente fundamental do sinal. Dessa maneira, esse filtro foi sintonizado na frequência de 60 Hz com objetivo de eliminar informação redundante, uma vez que, essa frequência está presente nas três fases do sinal e não representa muito do ponto de vista de caracterização da falta. As etapas que compõem esse método são mostradas no fluxograma da Figura 4.10.

Figura 4.10 – Fluxograma das etapas do método A.



Fonte: Do autor (2021).

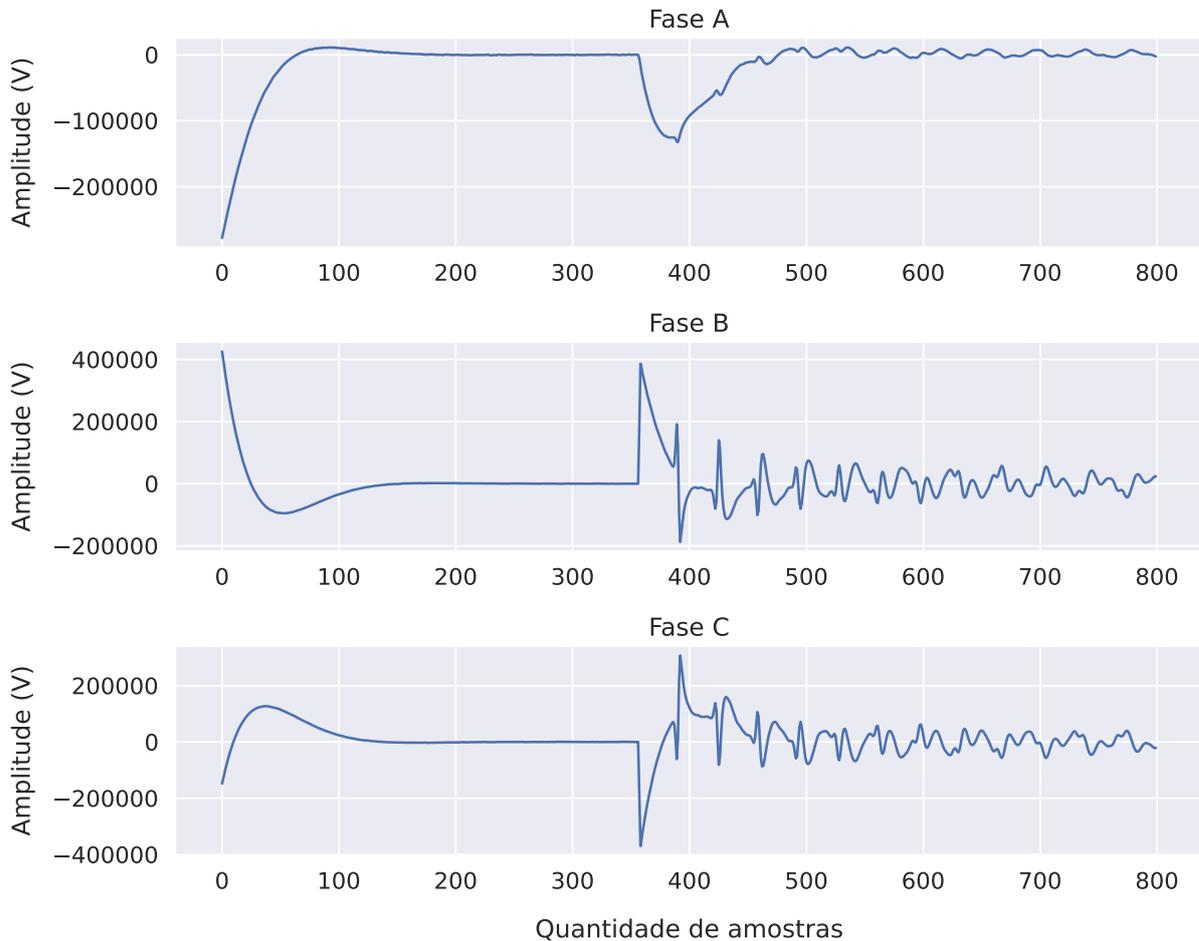
Considerando a equação do filtro *notch* mostrada em 3.1, utilizou-se o valor de $\rho_0 = 0,97$ pois ele apresentou melhores resultados para detecção e classificação de distúrbios como mostrado por Ferreira (2010), e portanto nesse projeto decidiu-se por utilizar o mesmo valor. Além disso, ω_0 utilizado nessa equação 3.1 levou em consideração as frequências de amostragem e corte segundo:

$$\omega_0 = \frac{2\pi f_0}{f_s} \quad (4.6)$$

onde, f_0 é a frequência de corte em Hz e f_s a frequência de amostragem em Hz. Dessa maneira, nesse trabalho $f_s = 15360 \text{ Hz}$, $f_0 = 60 \text{ Hz}$, ou seja, $\omega_0 = 24,544 \times 10^{-3} \text{ rad/s}$.

Ao utilizar o filtro *notch* com as especificações citadas, a frequência de 60 Hz é completamente removida do sinal. A Figura 4.11 apresenta o sinal gerado após aplicação desse filtro aos dados de tensão apresentados anteriormente na Figura 4.2.

Figura 4.11 – Comportamento da tensão de uma falta ABC 20km 0° 1 Ω após aplicação do filtro *notch*.



Fonte: Do autor (2021).

Pela Figura 4.11 nota-se que os primeiros pontos amostrais dos sinais de saída do filtro possuem um transitório causado pelo filtro *notch* que pode ser prejudicial no processo de classificação, por esse motivo eles são removidas (64 amostras). Após isso, todo o *dataset* foi normalizado entre -1 e 1 utilizando-se a classe *MinMaxScaler* do *Python*. Além disso, utilizou-se meio ciclo de sinal pós falta e as três fases dos sinais de tensão foram concatenadas gerando um vetor único. Dessa forma, cada amostra era composta por 384 pontos gerando um conjunto de *features* a ser utilizado na entrada dos classificadores.

Nesse método dois classificadores foram implementados, o primeiro é baseado em florestas aleatórias (*random forest*) cujo os hiperparâmetros utilizados podem ser vistos na Tabela 4.1.

Tabela 4.1 – Parâmetros utilizados classificador *random forest* implementado no método A.

Parâmetro	Significado	Valor usado
<i>n_estimators</i>	Quantidade de árvores na floresta.	50
<i>max_depth</i>	Profundidade máxima das árvores.	20
<i>min_samples_split</i>	Número mínimo de amostras necessárias para dividir um nó interno.	7
<i>min_samples_leaf</i>	Número mínimo de amostras necessárias para estar em um nó folha.	1
<i>criterion</i>	Função que mede a qualidade da divisão.	'entropy'
<i>bootstrap</i>	Todo conjunto de dados é usado para construir cada árvore.	False
<i>max_features</i>	Número de características consideradas quando procurar pela melhor divisão.	11

Fonte: Do autor (2020).

Por sua vez, o segundo classificador implementado foi baseado em uma rede neural do tipo MLP (*multilayer perceptron*), a qual os parâmetros são apresentados na Tabela 4.2.

Tabela 4.2 – Parâmetros utilizados na rede neural implementada no método A.

Parâmetro	Valor usado
Quantidade de entradas	384
Quantidade de camadas ocultas	1
Quantidade de saídas	10
Quantidade de neurônios na camada oculta	384
Inicialização dos pesos e bias	Inicialização Glorot
Função de ativação da camada oculta	Relu
Função de ativação da camada de saída	Softmax
Função de perda usada durante o treinamento	Entropia cruzada categórica
Algoritmo de treinamento utilizado	Adagrad
Taxa de treinamento utilizado	0,01
Atualização dos pesos	<i>Off-line</i> em lotes de 60
Quantidade de épocas	5000

Fonte: Do autor (2020).

Como citado anteriormente, os dados de entrada consistiram nos sinais filtrados e normalizados das três fases com meio ciclo de sinal pós-falta, isso corresponde a 128 pontos de amostra do sinal (características) para cada fase. Dessa maneira, o total de características (*features*) de cada exemplo a ser apresentado para os classificadores é igual a 384 (3 fases multiplicado por 128 características).

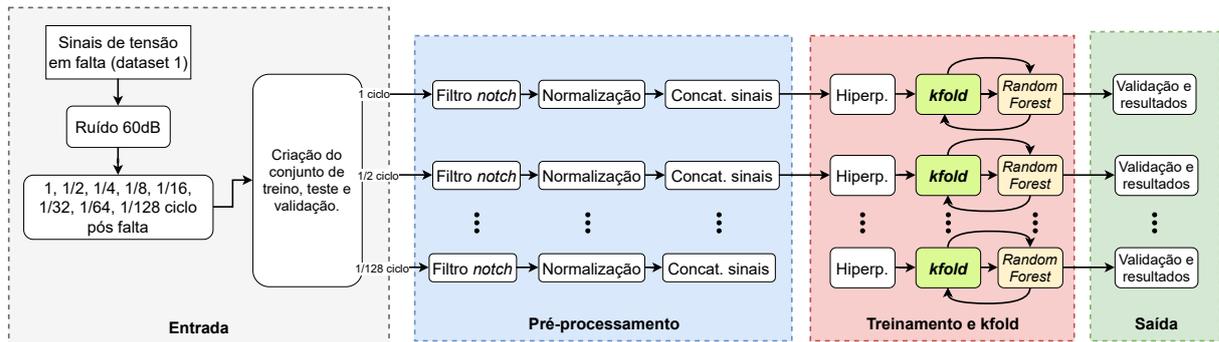
Cabe destacar que esse método foi apresentado no XXIII Congresso Brasileiro de Automação (CBA 2020), e compôs um artigo publicado na edição especial do *Journal of Control, Automation and Electrical Systems* (JCAE) que pode ser consultado em Fonseca et al. (2021). Portanto, maiores detalhes da metodologia e resultados obtidos podem ser encontrados no artigo em questão.

4.3.2.2 Método B

O segundo método implementado visou explorar o uso do modelo de *random forest* com variações da porção pós falta do sinal de tensão. Isso é importante para determinar a quantidade de dados mínima do sinal para gerar previsões satisfatórias. E ainda verificar qual é o comportamento dos modelos utilizados quando não existe tanta informação disponível em cada amostra para treinar e validar os modelos. Além disso, a variação do sinal não é um assunto muito abordado por autores que desenvolvem projetos nessa área. Isso é interessante pois, conforme apresentado por Chen, Huang e He (2016b), uma certa computação extra é necessária nos métodos para extrair características dos sinais originais de corrente e tensão, o que adiciona muita carga computacional aos dispositivos de monitoramento. Assim, alguns pesquisadores sugerem utilizar menos pontos amostrais dos sinais de corrente e tensão, ou seja, usar sinais com um quarto, um terço, meio ou um ciclo de sinal nos métodos de detecção, classificação ou localização de faltas.

Dessa forma, nesse e nos métodos que são apresentados a seguir nessa dissertação decidiu-se por utilizar os sinais de tensão com 1/4 de ciclo pré-falta e 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64 e 1/128 ciclo pós falta, gerando oito modelos distintos. A forma de treinamento foi bem similar ao que foi realizado anteriormente para o método A enquanto a validação foi executada conforme exposto na subseção 4.3.1. A Figura 4.12 mostra as etapas seguidas para o treinamento do modelo no método B.

Figura 4.12 – Fluxograma das etapas do método B.



Fonte: Do autor (2021).

Para esse método realizou-se a hiperparametrização com *random search* que amostra aleatoriamente o espaço de busca e avalia conjuntos de uma distribuição de probabilidade específica. Nesse caso, nem todas as possibilidades de combinação de hiperparâmetros são avaliadas o que pode levar a soluções sub-ótimas. Os hiperparâmetros obtidos para os modelos em cada uma dessas variações estão exibidos na Tabela 4.3.

Tabela 4.3 – Parâmetros utilizados nos classificadores *random forest* implementados no método B.

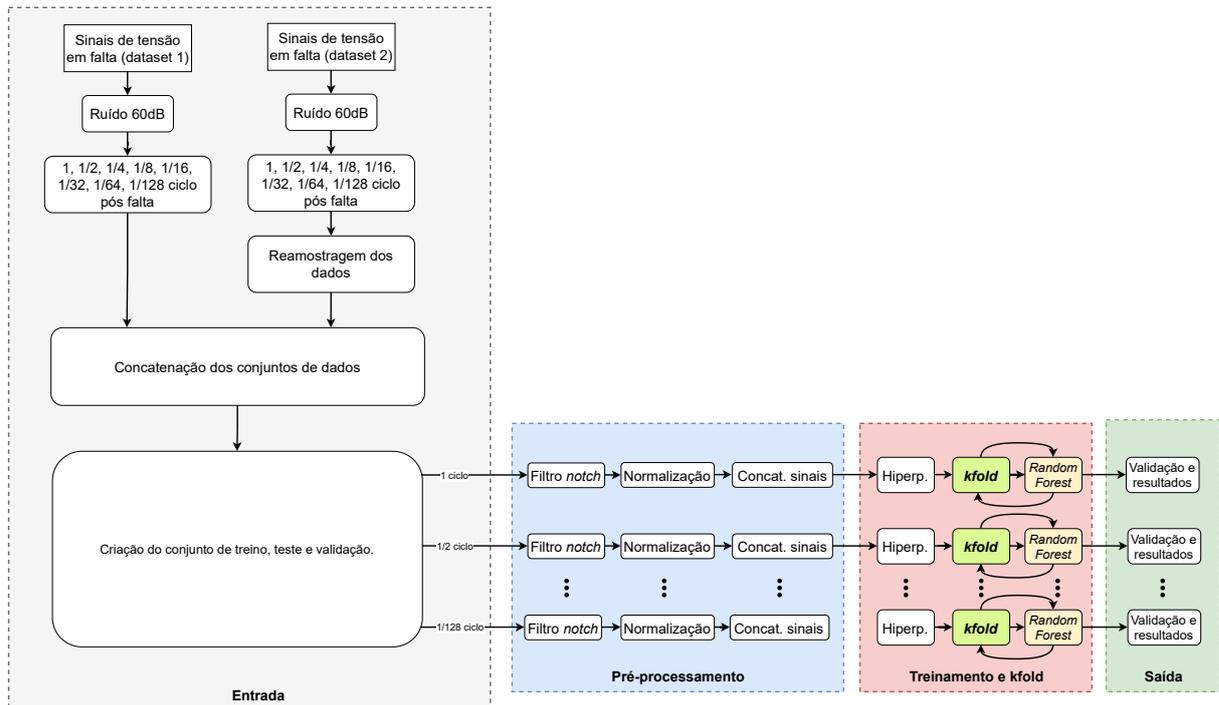
Varição	n_estimators	min_samples_split	min_samples_leaf	max_features	max_depth	criterion	bootstrap
1 ciclo	100	10	1	'auto'	30	'entropy'	True
1/2 ciclo	90	2	1	'sqrt'	30	'gini'	False
1/4 ciclo	60	5	1	'auto'	30	'entropy'	False
1/8 ciclo	100	5	1	'sqrt'	30	'gini'	False
1/16 ciclo	70	2	4	'auto'	10	'entropy'	False
1/32 ciclo	90	5	2	'auto'	10	'entropy'	False
1/64 ciclo	90	5	1	'sqrt'	30	'gini'	False
1/128 ciclo	50	2	1	'sqrt'	30	'entropy'	False

Fonte: Do autor (2021).

4.3.2.3 Método C

O terceiro método implementado tem muitas similaridades com o que foi apresentado anteriormente para o método B. As etapas de pré-processamento, treinamento e saída são exatamente iguais ao que foi apresentado na Figura 4.12. A diferença para o que foi realizado no método anterior foi a adição dos sinais provenientes do *dataset 2* para o treinamento dos modelos. Isso confere características e resultados diferentes que são melhor abordados no Capítulo 5. O fluxograma da Figura 4.13 mostram as etapas que foram seguidas para esse método que se diferem em relação ao método B somente na etapa de entrada.

Figura 4.13 – Fluxograma das etapas do método C.



Fonte: Do autor (2021).

Assim, como foi feito no método anterior, realizou-se a hiperparametrização para cada variante do conjunto de dados, os melhores hiperparâmetros gerados durante esse processo para o método C estão listados na Tabela 4.4.

Tabela 4.4 – Parâmetros utilizados nos classificadores *random forest* implementados no método C.

Variação	n_estimators	min_samples_split	min_samples_leaf	max_features	max_depth	criterion	bootstrap
1 ciclo	100	2	2	'sqrt'	50	'gini'	False
1/2 ciclo	80	2	2	'sqrt'	40	'entropy'	False
1/4 ciclo	50	10	1	'sqrt'	30	'entropy'	False
1/8 ciclo	70	5	1	'auto'	50	'entropy'	False
1/16 ciclo	60	10	1	'auto'	30	'entropy'	False
1/32 ciclo	70	5	2	'sqrt'	30	'entropy'	False
1/64 ciclo	60	5	1	'auto'	10	'entropy'	False
1/128 ciclo	100	2	1	'auto'	30	'entropy'	False

Fonte: Do autor (2021).

4.3.3 Métodos treinados usando sinais de corrente

Para efeitos de comparação, dois métodos foram implementados utilizando-se florestas aleatórias, sendo que utilizou-se os sinais de corrente para treiná-las. Além deles, esse tipo de sinal foi usado no treinamento de dois métodos usando *Rocket* e *MiniRocket* cuja as teorias foram melhor explicadas nas seções 3.2 e 3.3 respectivamente.

4.3.3.1 Método D

Esse método é muito similar ao método B apresentado em 4.3.2.2. Ao observar o fluxograma apresentado na Figura 4.12 a diferença entre o método D e o método B é a substituição dos dados de tensão pelos dados de corrente na etapa de entrada. As demais fases são exatamente iguais entre os dois métodos. No entanto, hiperparâmetros distintos são obtidos, isso pode ser verificado na Tabela 4.5.

Tabela 4.5 – Parâmetros utilizados nos classificadores *random forest* implementados no método D.

Varição	n_estimators	min_samples_split	min_samples_leaf	max_features	max_depth	criterion	bootstrap
1 ciclo	80	5	1	'auto'	20	'entropy'	False
1/2 ciclo	70	5	2	'auto'	10	'gini'	False
1/4 ciclo	30	2	1	'sqrt'	40	'entropy'	False
1/8 ciclo	50	2	2	'auto'	30	'entropy'	False
1/16 ciclo	70	2	2	'sqrt'	10	'entropy'	False
1/32 ciclo	40	10	1	'auto'	20	'entropy'	False
1/64 ciclo	90	15	1	'auto'	20	'entropy'	False
1/128 ciclo	90	5	1	'auto'	10	'gini'	False

Fonte: Do autor (2021).

4.3.3.2 Método E

O método E é muito semelhante ao que foi implementado para o método C apresentado em 4.3.2.3. Observando-se o fluxograma mostrado na Figura 4.13, a única diferença foi a utilização de dados de corrente em vez de dados de tensão. Para esse método os hiperparâmetros utilizados são listados na Tabela 4.6.

Tabela 4.6 – Parâmetros utilizados nos classificadores *random forest* implementados no método E.

Varição	n_estimators	min_samples_split	min_samples_leaf	max_features	max_depth	criterion	bootstrap
1 ciclo	70	5	1	'sqrt'	10	'gini'	False
1/2 ciclo	50	2	2	'auto'	10	'gini'	False
1/4 ciclo	50	2	1	'auto'	10	'gini'	False
1/8 ciclo	100	10	1	'sqrt'	50	'gini'	True
1/16 ciclo	70	5	2	'auto'	20	'entropy'	False
1/32 ciclo	100	2	1	'sqrt'	30	'entropy'	False
1/64 ciclo	100	2	1	'sqrt'	30	'entropy'	False
1/128 ciclo	90	2	4	'sqrt'	40	'entropy'	False

Fonte: Do autor (2021).

4.3.3.3 Método F

O método F utilizou os sinais de corrente do *dataset* 1 como entrada para os modelos tal qual o método D apresentado anteriormente. A diferença desse para os métodos apresentados anteriormente é a maneira como os dados foram pré-processados e o classificador utilizado.

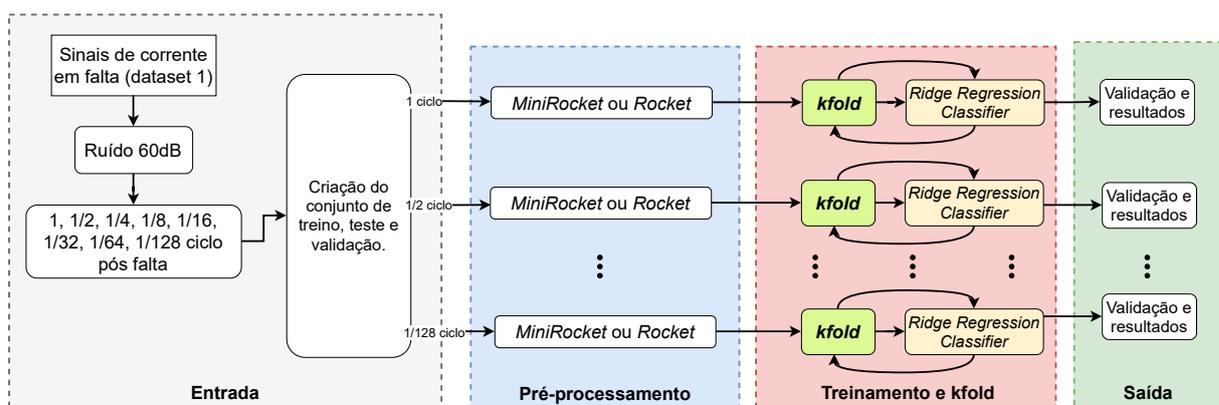
Nesse caso foram conduzidos dois experimentos distintos, o primeiro foi a realização da transformação dos dados utilizando o modelo *Rocket* cuja teoria foi apresentada na seção 3.2. O segundo foi transformar os dados usando *MiniRocket* como explicado na seção 3.3 e a classificação foi conduzida utilizando a regressão *ridge*.

Em ambos os casos, esses modelos foram utilizados para transformar os sinais de corrente em um grande conjunto de *features* que posteriormente foram utilizadas na classificação da falta usando o modelo de regressão *Ridge* apresentado na seção 3.4.1. Nesse método não foi realizada a hiperparametrização tendo em vista que os modelos utilizados possuem poucos hiperparâmetros. Para o *Rocket* deve-se ajustar somente o número de *kernels* que serão utilizados, e o modelo gera duas *features* por *kernel*. Já para o *MiniRocket* basta setar o número de *features* na qual se quer transformar a série temporal. Para, analisar o impacto desses parâmetros, variou-se seus valores de 100 em 100 registrando a acurácia e o tempo necessário para realizar o treinamento.

Por sua vez, a regressão *Ridge* tem como único hiperparâmetro necessário a ser ajustado o valor de alfa, que indica a força da regularização. Nesse caso, optou-se por usar os mesmos valores que são apresentados na demonstração de exemplo do método *Rocket* da biblioteca *sktime* (LÖNING et al., 2021; LÖNING et al., 2019). Ou seja, uma série logarítmica de dez valores que variam entre 0,001 e 1000.

O fluxograma das etapas seguidas para realização desse método é apresentado na Figura 4.14.

Figura 4.14 – Fluxograma das etapas do método F.

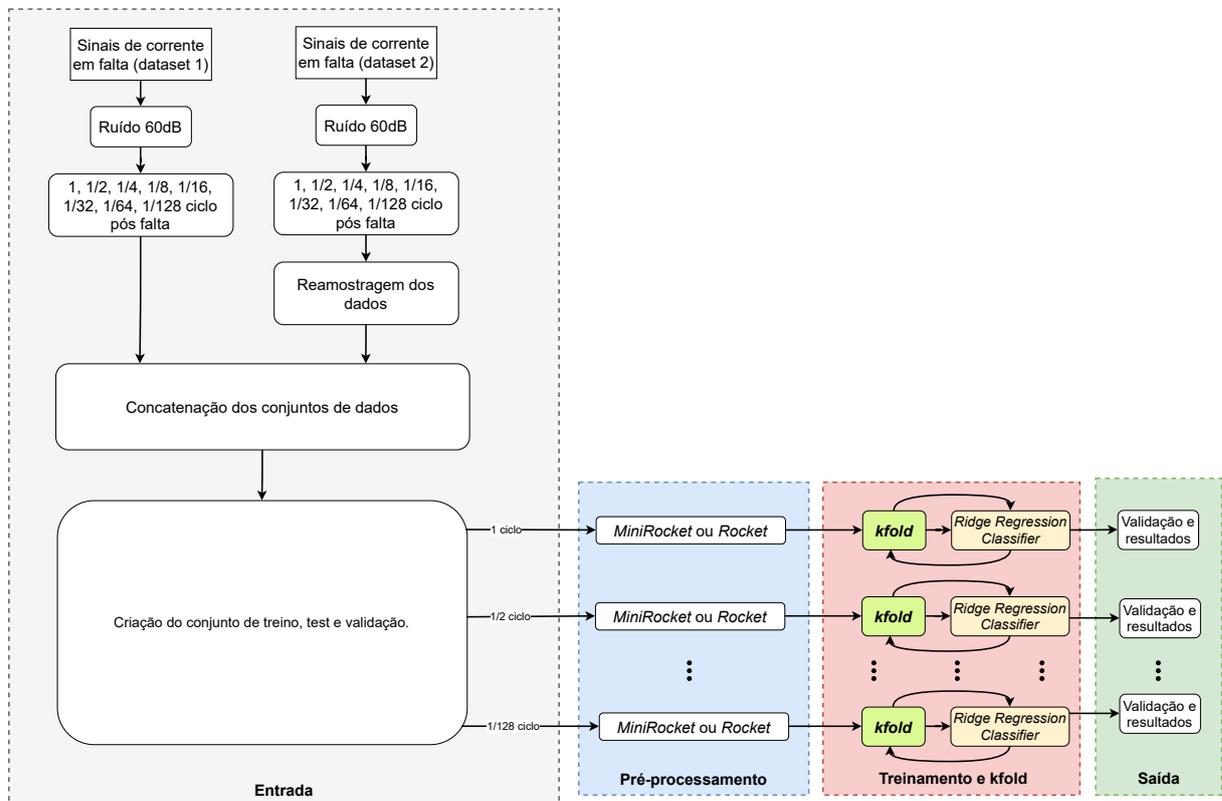


Fonte: Do autor (2021).

4.3.3.4 Método G

O último método implementado nessa dissertação tem semelhanças com outros dois métodos anteriormente explicados. Ele utiliza uma combinação dos dados de corrente do *dataset 1* e *dataset 2*, tal qual citado em 4.3.3.2 para o método E. Além disso utiliza *Rocket* ou *MiniRocket* na transformação dos sinais e geração de características como mostrado em 4.3.3.3 para o método F. Também realizou-se a variação da quantidade de *kernels* e *features* para analisar a acurácia obtida e o tempo necessário para treinamento. O fluxograma na Figura 4.15 resume as etapas citadas na construção desse método.

Figura 4.15 – Fluxograma das etapas do método G.



Fonte: Do autor (2021).

5 RESULTADOS E DISCUSSÕES

5.1 Comparação entre os bancos de dados

Na seção 4.1 foram apresentados os bancos de dados usados nessa dissertação. Apesar de terem muitos parâmetros iguais entre eles, apenas a inclusão do banco de capacitores no *dataset 2* é capaz de alterar o comportamento dos sinais.

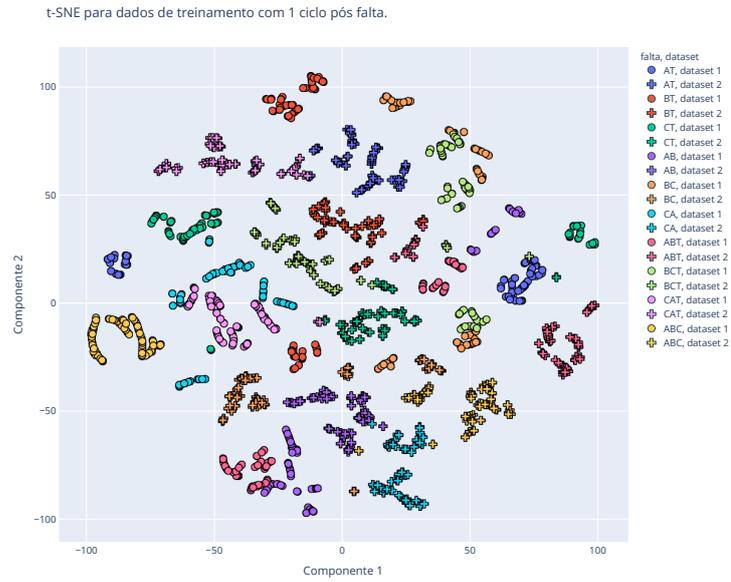
Durante os resultados apresentados nesse capítulo, é possível observar que com a inclusão do *dataset 2* os modelos treinados perdem a capacidade de generalização quando existem poucos pontos amostrais no sinal pós falta. Para efeito de comparação, utilizou-se a técnica t-SNE que foi melhor apresentada na seção 3.9. Ela permite visualizar como os dados usados se distribuem em um espaço bi ou tridimensional possibilitando observar a relação entre as classes e como as amostras estão separadas ou agrupadas nesse espaço.

A Figura 5.1 apresenta o t-SNE em duas dimensões para um ciclo de sinal pós falta. Na imagem é possível notar que existem *clusters* relativamente separados no espaço. Os dados originais tem cerca de cem *features* então deve-se destacar que existe uma perda considerável de informação ao realizar essa redução de dimensionalidade. O que é interessante de notar pelo gráfico é quais tipos de faltas estão próximas entre si, indicando que existe um grau de semelhança e que um possível classificador poderia se confundir entre elas. Mas novamente, apesar de duas dimensões permitir uma melhor visualização dos dados, quando as amostras estão em alta dimensão uma decisão diferente pode ser tomada pelos classificadores.

Pela Figura 5.1 nota-se que existe em geral uma diferença significativa entre as classes nos dois conjuntos de dados. Ou seja, as amostras de um tipo de falta para o *dataset 1* não estão próximas das amostras do mesmo tipo no *dataset 2*. Além disso, é possível notar que para esses dados com um ciclo de sinal pós falta, as amostras do *dataset 2* estão mais separadas, enquanto para o *dataset 1* algumas amostras com tipo de falta diferente formam alguns *clusters*, como é o caso das faltas ABT e AB, CAT e CA, BCT e BC. O que mostra que para essas amostras nesse conjunto pode ocorrer confusão entre as faltas bifásicas com terra e bifásicas.

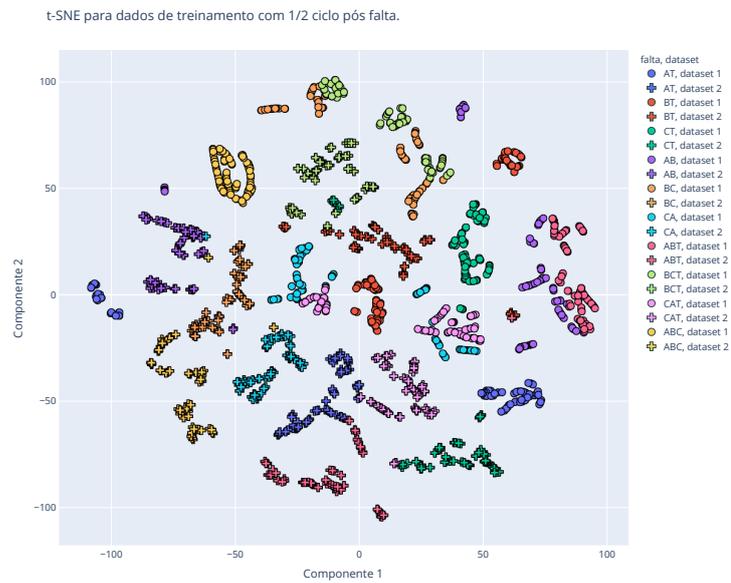
Essa análise foi conduzida também para as demais quantidades de ciclo pós falta. Na Figuras 5.2, Figuras 5.3, Figuras 5.4, Figuras 5.5 estão respectivamente os gráficos do t-SNE para dados com 1/2, 1/4, 1/8 e 1/16 ciclo pós falta. Nas imagens é possível observar que o comportamento das amostras no espaço se mantém de certa forma constante para o *dataset 1*, enquanto algumas mudanças começam a surgir para o *dataset 2*.

Figura 5.1 – t-SNE para dados de treinamento com 1 ciclo pós falta.



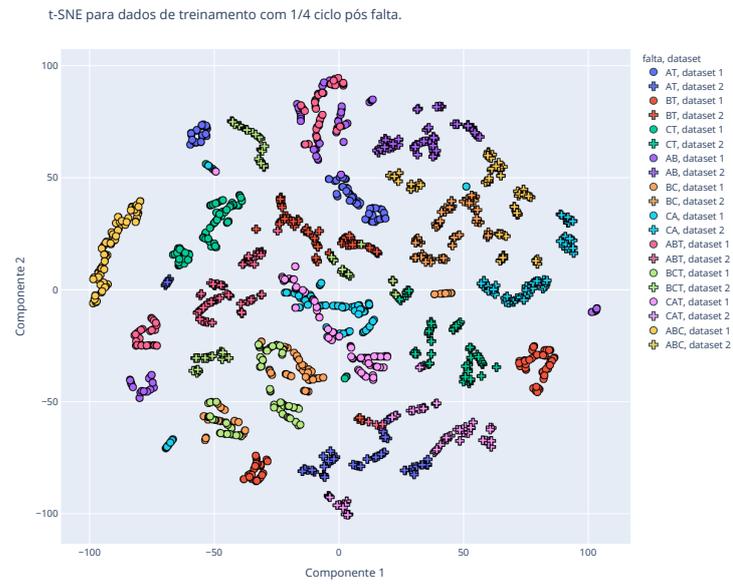
Fonte: Do autor (2021).

Figura 5.2 – t-SNE para dados de treinamento com 1/2 ciclo pós falta.



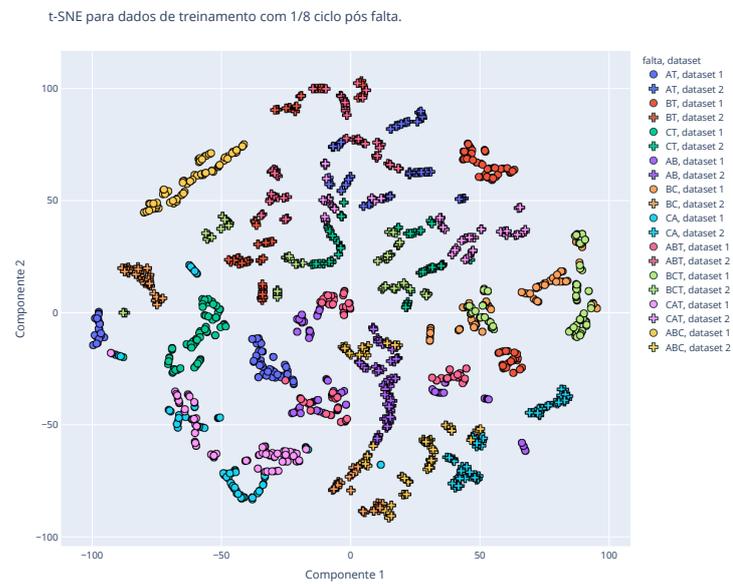
Fonte: Do autor (2021).

Figura 5.3 – t-SNE para dados de treinamento com 1/4 ciclo pós falta.



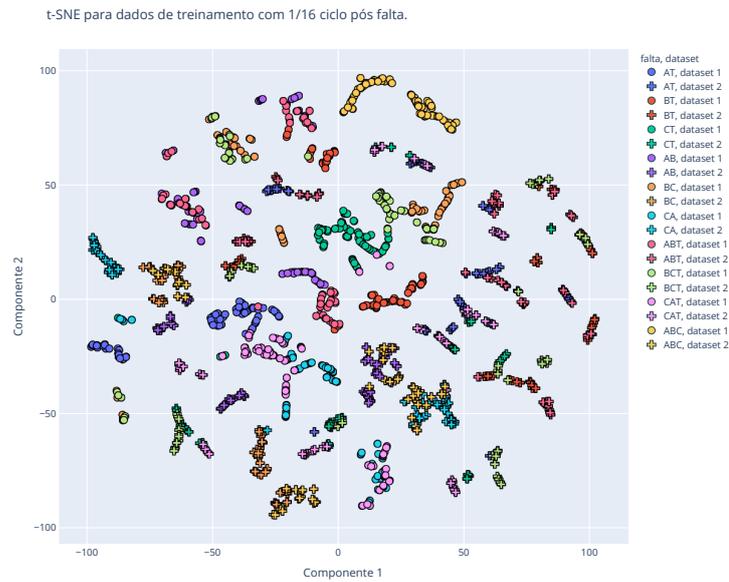
Fonte: Do autor (2021).

Figura 5.4 – t-SNE para dados de treinamento com 1/8 ciclo pós falta.



Fonte: Do autor (2021).

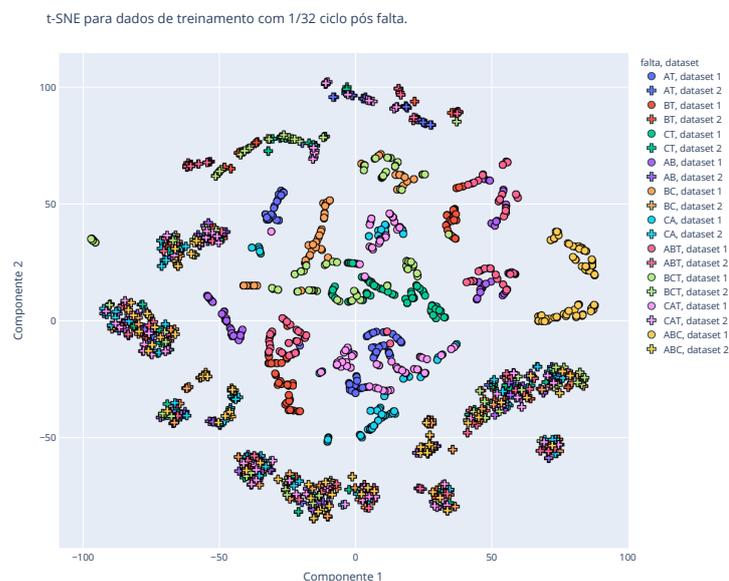
Figura 5.5 – t-SNE para dados de treinamento com 1/16 ciclo pós falta.



Fonte: Do autor (2021).

É possível observar que com a diminuição da quantidade de ciclo pós falta amostras com tipo de falta diferentes no *dataset 2* começam a se agrupar em *clusters*. Isso prejudica muito o treinamento dos modelos como será apresentado nesse capítulo. A Figura 5.6, Figura 5.7 e Figura 5.8 mostram respectivamente os dados de treinamento com 1/32, 1/64 e 1/128 ciclo pós falta. As imagens mostram claramente essa formação de *clusters* no *dataset 2* mesmo para tipos de falta completamente diferentes.

Figura 5.6 – t-SNE para dados de treinamento com 1/32 ciclo pós falta.



Fonte: Do autor (2021).

Figura 5.7 – t-SNE para dados de treinamento com 1/64 ciclo pós falta.

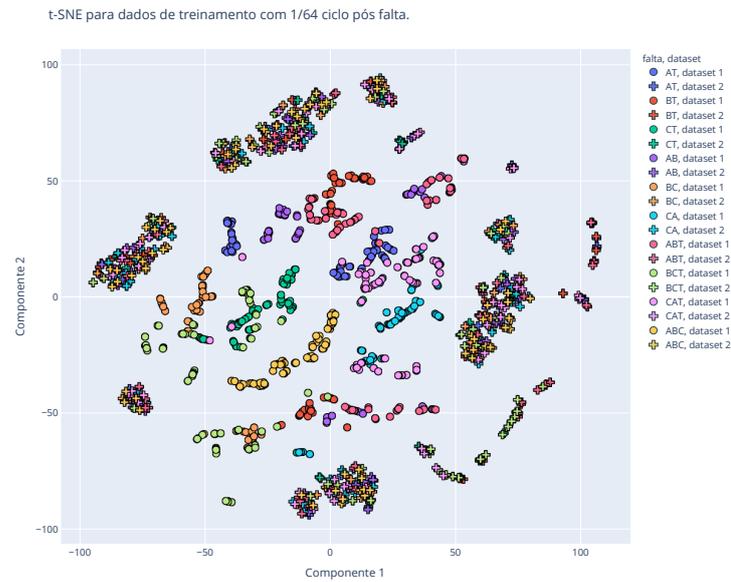
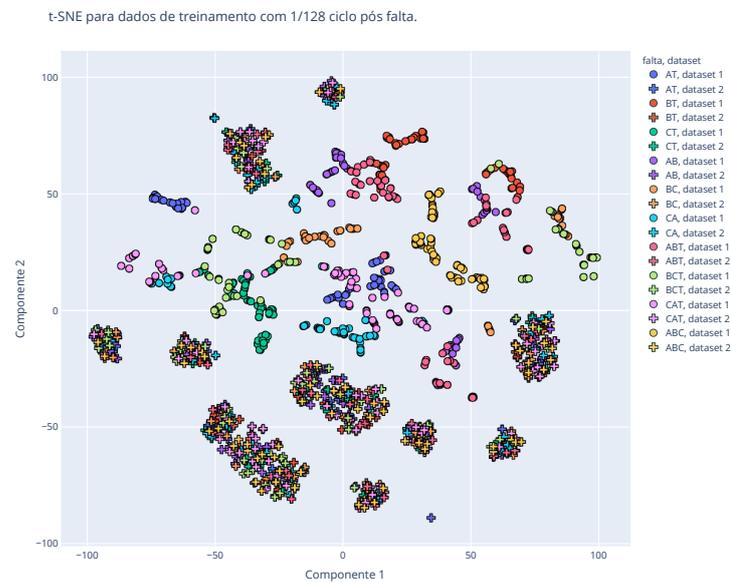


Figura 5.8 – t-SNE para dados de treinamento com 1/128 ciclo pós falta.



5.2 Métodos implementados

Antes de apresentar os resultados obtidos por cada método que foi implementado é importante recapitular quais as principais diferenças entre eles. Isso pode ser observado na Tabela 5.1.

Tabela 5.1 – Resumo das principais diferenças de cada modelo implementado.

Método	Tipo de sinal	Conjunto de dados	Ciclo pós-falta	Processamento	Classificador
A	Tensão	dataset 1	1/2	Filtro notch	Random forest e redes neurais
B	Tensão	dataset 1	1 a 1/128	Filtro notch	Random forest
C	Tensão	dataset 1 + dataset 2	1 a 1/128	Filtro notch	Random forest
D	Corrente	dataset 1	1 a 1/128	Filtro notch	Random forest
E	Corrente	dataset 1 + dataset 2	1 a 1/128	Filtro notch	Random forest
F	Corrente	dataset 1	1 a 1/128	MiniRocket e Rocket	Regressão ridge
G	Corrente	dataset 1 + dataset 2	1 a 1/128	MiniRocket e Rocket	Regressão ridge

Fonte: Do autor (2021).

5.3 Método A

Como foi apresentado na subseção 4.3.2.1, os resultados do método A podem ser melhor explorados no artigo publicado em Fonseca et al. (2021), mas um breve resumo encontra-se a seguir. Para o modelo de *random forest* nesse método, a acurácia média obtida foi de 91,96% com desvio padrão de 1,06%. Os valores obtidos durante a validação cruzada para cada um dos *folds* foram: [93,00; 89,28; 91,46; 92,46; 92,64; 93,22; 92,38; 92,05; 91,53; 91,62].

Após o *k-fold*, o conjunto de validação foi usado no modelo treinado para realizar a classificação em um conjunto de dados não visto. Para isso, o modelo de florestas aleatórias com maior acurácia no conjunto de teste foi usado na validação final. A matriz de confusão gerada é apresentada na Tabela 5.2, onde a acurácia global obtida foi de 91,49%. É possível observar também que o modelo possui pelo menos um erro em faltas que envolvem o terra, exceto para faltas do tipo CT. Isso mostra que há alguma dificuldade para o modelo de florestas aleatórias distinguir os tipos de falta quando o terra está envolvido (FONSECA et al., 2021).

Para o modelo de redes neurais, os seguintes valores de acurácia foram obtidos em cada um dos *folds*: [91,78; 86,97; 88,52; 89,82; 90,26; 90,68; 89,93; 89,88; 89,02; 89,05]. Assim, a acurácia média obtida pelo modelo foi de 89,59% com desvio padrão de 1,24%. O melhor modelo salvo durante o treinamento foi usado para gerar a matriz de confusão apresentada na Tabela 5.3. É possível verificar então que a acurácia de validação foi de 96,81%, mostrando que utilizar o modelo de redes neurais nesse caso permitiu obter uma melhor generalização para dados não vistos. Observa-se também que há uma dificuldade do modelo em predizer faltas ABC e AT.

Tabela 5.2 – Matriz de confusão para o modelo de florestas aleatórias no conjunto de validação.

		Valores preditos										
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT	
Valores reais	AB	100%										
		10/10										
	ABC		100%									
			12/12									
	ABT			85.7%				14.3%				
				6/7				1/7				
	AT				76.9%				15.4%	7.7%		
					10/13				2/13	1/13		
	BC					100%						
						5/5						
	BCT	8.3%					91.7%					
		1/12					11/12					
	BT		11.1%	11.1%				77.8%				
		1/9	1/9				7/9					
CA								100%				
								4/4				
CAT	7.7%								92.3%			
	1/13								12/13			
CT										100%		
										9/9		

Fonte: Fonseca et al. (2021).

Tabela 5.3 – Matriz de confusão para o modelo de redes neurais no conjunto de validação.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	100%									
		10/10									
	ABC		91.7%			8.3%					
			11/12			1/12					
	ABT			100%							
				7/7							
	AT				84.6%				15.4%		
					11/13				2/13		
	BC					100%					
						5/5					
	BCT						100%				
							12/12				
	BT							100%			
							9/9				
CA								100%			
								4/4			
CAT									100%		
									13/13		
CT										100%	
										9/9	

Fonte: Fonseca et al. (2021)

5.4 Método B

Esse método foi apresentado na subseção 4.3.2.2, mas resumidamente ele utilizou os dados de tensão do *dataset 1*, filtro *notch* e *random forest*. Os resultados obtidos são apresentados na Tabela 5.4, enquanto maiores detalhes sobre a acurácia em cada *fold* e a matriz de confusão se encontram na seção A.1 no Apêndice A.

Tabela 5.4 – Resultados obtidos pelo método B.

Pós falta	Acurácia			Tempo Operacional	
	Média trein. (%)	Desvio padrão trein. (%)	Validação (%)	Mín. (s)	Máx. (s)
1	97,21	1,62	97,87	0,066	0,134
1/2	97,48	1,50	98,94	0,042	0,043
1/4	97,34	1,68	95,21	0,038	0,038
1/8	97,61	1,00	97,87	0,025	0,026
1/16	97,74	1,04	96,81	0,028	0,029
1/32	97,08	1,54	94,15	0,024	0,024
1/64	96,94	0,86	96,28	0,024	0,024
1/128	96,15	1,50	95,74	0,026	0,026

Do autor (2021).

5.5 Método C

O método C foi apresentado na subseção 4.3.2.3 e resumidamente nele utilizou-se os dados de tensão provenientes tanto do *dataset 1* quanto do *dataset 2*. Além disso, utilizou-se o filtro *notch* para pré-processamento e o modelo de florestas aleatórias como classificador. Os resultados obtidos são apresentados na Tabela 5.5, enquanto maiores detalhes sobre a acurácia em cada *fold* e a matriz de confusão se encontram na seção A.2 no Apêndice A.

Tabela 5.5 – Resultados obtidos pelo método C.

Ciclo pós falta	Acurácia			Tempo Operacional	
	Média trein.(%)	Desvio padrão trein. (%)	Validação (%)	Mín. (s)	Máx. (s)
1	98,20	1,12	98,14	0,071	0,073
1/2	98,80	0,65	98,40	0,051	0,053
1/4	97,14	1,59	97,61	0,037	0,038
1/8	96,34	1,16	96,54	0,026	0,027
1/16	93,62	1,53	94,68	0,030	0,032
1/32	69,34	3,50	69,68	0,025	0,026
1/64	63,97	2,69	61,70	0,027	0,030
1/128	52,99	2,26	51,86	0,019	0,019

Do autor (2021).

5.6 Método D

O método D foi apresentado na subseção 4.3.3.1, ele é bem similar ao que foi apresentado no método B. A diferença reside no fato dele utilizar os dados de corrente do *dataset 1*, ele também utiliza filtro *notch* e florestas aleatórias como os outros métodos. Os resultados obtidos são apresentados na Tabela 5.6, enquanto maiores detalhes sobre a acurácia em cada *fold* e a matriz de confusão se encontram na seção A.3 no Apêndice A.

Tabela 5.6 – Resultados obtidos pelo método D.

Ciclo pós falta	Acurácia			Tempo Operacional	
	Média trein.(%)	Desvio padrão trein. (%)	Validação (%)	Mín. (s)	Máx. (s)
1	99,60	0,61	99,47	0,069	0,070
1/2	99,47	0,65	100,00	0,045	0,046
1/4	98,93	0,80	97,87	0,030	0,030
1/8	99,34	0,66	97,87	0,025	0,026
1/16	97,74	1,34	95,74	0,020	0,020
1/32	96,94	1,47	95,21	0,022	0,022
1/64	94,68	1,87	93,62	0,019	0,019
1/128	93,75	1,46	90,96	0,027	0,027

Do autor (2021).

5.7 Método E

O método E foi apresentado na subseção 4.3.3.2, ele possui muitas semelhanças ao que foi exposto para o método C, no entanto, esse método utiliza os sinais de corrente do *dataset 1* e do *dataset 2*. E assim como nos métodos citados anteriormente, nele foi usado o filtro *notch* e o modelo de florestas aleatórias. Os resultados obtidos são apresentados na Tabela 5.7, enquanto maiores detalhes sobre a acurácia em cada *fold* e a matriz de confusão se encontram na seção A.4 no Apêndice A.

Tabela 5.7 – Resultados obtidos pelo método E.

Ciclo pós falta	Acurácia			Tempo Operacional	
	Média trein.(%)	Desvio padrão trein. (%)	Validação (%)	Mín. (s)	Máx. (s)
1	99,60	0,44	99,73	0,071	0,072
1/2	99,93	0,20	99,73	0,046	0,047
1/4	99,53	0,52	99,73	0,037	0,037
1/8	99,20	0,78	98,67	0,029	0,029
1/16	94,14	1,77	92,82	0,024	0,024
1/32	68,28	2,23	67,82	0,030	0,030
1/64	57,31	1,98	57,98	0,024	0,025
1/128	51,19	1,80	50,80	0,017	0,018

Do autor (2021).

5.8 Método F

O método F foi apresentado na subseção 4.3.3.3, nele utilizou-se como entrada os sinais de corrente somente do *dataset 1*. Na etapa de pré-processamento duas técnicas novas foram utilizadas, primeiramente os sinais de entrada foram transformados utilizando-se a transformação *Rocket* e depois novos modelos foram treinados após pré-processamento com a técnica *MiniRocket*. Para esse método variou-se respectivamente a quantidade de *kernels* e *features*, nas subseções a seguir são apresentados os valores que obtiveram melhores resultados durante o treinamento e validação, mas no apêndice B são apresentadas tabelas com todos resultados obtidos. Como classificador foi utilizada a regressão ridge, ela não foi hiperparametrizada, mantendo-se como constante o valor de regularização que fornece valores em uma distribuição logarítmica entre 0,001 e 1000.

5.8.1 *Rocket*

Para utilizar o método *Rocket* variou-se a quantidade de *kernels* utilizados e registrou-se a acurácia de treinamento e validação. Importante lembrar que essa técnica gera dois *features* por *kernel*. A Tabela 5.8 apresenta os resultados obtidos, enquanto maiores detalhes sobre a acurácia em cada *fold* e as matrizes de confusão se encontram na seção A.5 no Apêndice A. Além disso, no apêndice B foi incluída a Tabela 65 que apresenta todos os resultados obtidos durante os testes.

Tabela 5.8 – Resultados obtidos pelo método F utilizando-se *Rocket*

Ciclo pós falta	<i>Kernels</i>	Acurácia			Tempo Operacional	
		Média trein.(%)	Desvio padrão trein. (%)	Validação (%)	Mín. (s)	Máx. (s)
1	300	100,00	0,00	100,00	0,014	0,015
1/2	400	99,87	0,40	100,00	0,014	0,014
1/4	400	99,73	0,53	100,00	0,013	0,015
1/8	500	99,07	1,04	100,00	0,014	0,030
1/16	900	97,87	1,60	98,40	0,018	0,019
1/32	1000	95,21	3,28	98,40	0,019	0,021
1/64	900	94,01	2,75	96,81	0,021	0,026
1/128	600	93,48	3,00	94,68	0,022	0,023

Do autor (2021).

5.8.2 *MiniRocket*

Para utilizar a técnica *MiniRocket* variou-se a quantidade *features* e foram registradas a acurácia de treinamento e validação, além dos tempos necessários para treinar e validar os modelos. A Tabela 5.9 resume os resultados obtidos, maiores detalhes sobre a acurácia em cada

fold e as matrizes de confusão se encontram na seção A.5 no Apêndice A. No apêndice B foi incluída a Tabela 66 que apresenta os resultados completos obtidos nos testes realizados.

Tabela 5.9 – Resultados obtidos pelo método F utilizando-se *MiniRocket*

Ciclo pós falta	Features	Acurácia			Tempo Operacional	
		Média trein.(%)	Desvio padrão trein. (%)	Validação (%)	Mín. (s)	Máx. (s)
1	100	100,00	0,00	100,00	0,012	0,014
1/2	300	100,00	0,00	100,00	0,012	0,012
1/4	400	100,00	0,00	100,00	0,012	0,012
1/8	400	99,87	0,40	100,00	0,012	0,012
1/16	300	99,47	0,60	99,47	0,011	0,011
1/32	500	99,07	1,34	98,94	0,012	0,012
1/64	300	97,07	1,96	97,87	0,011	0,011
1/128	800	97,34	1,89	97,87	0,014	0,014

Do autor (2021).

5.9 Método G

Por fim, o último método implementado que foi exposto na subseção 4.3.3.4 fez uso dos dados de corrente tanto do *dataset 1* quanto do *dataset 2*. Além disso, ele utilizou as mesmas técnicas do método F, ou seja, o pré-processamento foi realizado utilizando *Rocket* ou *MiniRocket* e a classificação foi conduzida pela regressão *ridge*. Nas subseções a seguir são apresentados os resultados que foram obtidos ao treinar e validar os dados utilizando-se esse método.

5.9.1 *Rocket*

Nesse método novamente foi utilizada a transformação *Rocket* variando-se a quantidade de *kernels*. Nesse processo registrou-se a acurácia de treinamento e validação e o tempo necessário para treinar e validar os modelos. Maiores detalhes sobre a acurácia em cada *fold* e as matrizes de confusão se encontram na seção A.6 no Apêndice A. Por sua vez, a Tabela 5.10 apresenta um resumo com os resultados obtidos. Na Tabela 67 no apêndice B é possível verificar os resultados de todos os testes realizados.

Tabela 5.10 – Resultados obtidos pelo método G utilizando-se *Rocket*

Ciclo pós falta	Kernels	Acurácia			Tempo Operacional	
		Média trein.(%)	Desvio padrão trein. (%)	Validação (%)	Mín. (s)	Máx. (s)
1	200	99,73	0,33	100,00	0,013	0,014
1/2	400	99,47	0,83	100,00	0,016	0,016
1/4	500	99,27	0,76	100,00	0,016	0,020
1/8	500	98,73	0,81	98,94	0,019	0,021
1/16	800	94,21	1,59	94,15	0,024	0,027
1/32	800	64,89	2,45	66,49	0,024	0,026
1/64	700	54,39	2,73	56,12	0,023	0,025
1/128	1000	48,21	2,65	50,80	0,027	0,029

Do autor (2021).

5.9.2 *MiniRocket*

Utilizando a transformação *MiniRocket* variou-se a quantidade de *features* utilizadas e foram obtidas as mais variadas acurácias de treinamento e validação, além de verificar o tempo necessário nessas etapas. Maiores detalhes sobre a acurácia em cada *fold* e as matrizes de confusão se encontram na seção A.6 no Apêndice A. Por sua vez, a Tabela 5.11 apresenta um resumo com os resultados obtidos. Na Tabela 67 no apêndice B é possível verificar os resultados de todos os testes realizados.

Tabela 5.11 – Resultados obtidos pelo método G utilizando-se *MiniRocket*

Ciclo pós falta	Features	Acurácia			Tempo Operacional	
		Média trein.(%)	Desvio padrão trein. (%)	Validação (%)	Mín. (s)	Máx. (s)
1	500	99,73	0,61	100,00	0,014	0,014
1/2	500	99,47	0,40	100,00	0,013	0,013
1/4	500	98,67	0,60	99,73	0,012	0,013
1/8	900	98,27	1,20	99,47	0,013	0,014
1/16	700	92,08	1,94	93,09	0,013	0,013
1/32	600	62,30	2,30	65,96	0,012	0,013
1/64	600	53,79	1,49	57,71	0,012	0,013
1/128	500	54,05	1,09	55,59	0,011	0,013

Do autor (2021).

5.10 Discussões

Nessa dissertação foram implementados diversos métodos que possuem algumas diferenças em relação aos dados usados ou as técnicas utilizadas no pré-processamento e classificação. O primeiro método testado difere-se um pouco dos demais pois foi implementado em um momento incipiente do projeto, por esse motivo as metodologias de execução foram conduzidas de uma forma um pouco diferente em relação aos demais. Ele foi nomeado de método A e gerou um artigo que foi publicado em Fonseca et al. (2021), o objetivo dele foi executar um

estudo comparativo utilizando os métodos de redes neurais e florestas aleatórias. Como etapa de pré-processamento os sinais foram normalizados para ficarem entre -1 e 1 e o filtro *notch* foi utilizado para remover a frequência de 60 Hz. Nesse método foi utilizado somente os sinais de tensão do *dataset 1* com meio ciclo pós falta. Observou-se que durante o treinamento o modelo de florestas aleatórias obteve um melhor resultado do que a rede neural, 91,96% e 89,59% respectivamente. No entanto, com o conjunto de dados que foi utilizado para validação o resultado foi um pouco diferente obtendo-se 91,49% para florestas aleatórias e 96,81% para a rede neural. Contudo *random forest* foi 8 vezes mais rápido durante o processo de treinamento o que pode ser um diferencial dependendo da aplicação. Além disso, esse modelo tem uma complexidade menor quando comparado com a rede neural então pode ser mais simples de ser implementado em *hardware* exigindo menor capacidade computacional.

Como foi citado, os demais métodos seguiram uma metodologia um pouco diferente. O objetivo deles foi estudar o impacto da quantidade de pontos amostrais do sinal na parte pós falta. Isso é interessante para estressar os modelos e verificar qual o limite de uso deles quando as informações provenientes para entrada estão faltantes ou incompletas. Além disso, decidiu-se por utilizar um segundo banco de dados para suprir algumas características não encontradas no primeiro. O *dataset 1* acabou gerando ótimos resultados independente da quantidade de ciclos pós falta utilizada no sinal, no entanto, com a inclusão do *dataset 2* verificou-se que alguns resultados ficaram aquém do esperado. Cabe destacar que um teste foi realizado treinando os modelos com o *dataset 1* e usando somente o *dataset 2* para validação. No entanto, os resultados ficaram muito abaixo do esperado com acurácias entre 40% e 50%. Por esse motivo, demais testes dessa natureza foram abandonados e os resultados não foram incluídos no Capítulo 5. Pelos gráficos t-SNE apresentados na seção 5.1 observa-se que os conjuntos de dados têm bastante distinção entre si, e os mesmos tipos de faltas ficam afastados no espaço 2D, o que é um indício do motivo de o treinamento com um conjunto não ter generalizado bem para o outro. Dessa maneira, decidiu-se por utilizar os dois *datasets* em conjunto, o que permitiu obter resultados bem interessantes.

No método B foi utilizado o filtro *notch* com dados de tensão do *dataset 1* e *random forest* como classificador. Para esse e, em geral, os demais métodos notou-se a ocorrência de uma diminuição da acurácia de acordo com a diminuição da quantidade de sinal pós falta. No entanto, como hiperparâmetros diferentes foram utilizados e também a depender do processamento do computador na hora da execução dos testes isso pode acabar se divergindo um pouco.

Para o método B, cujo os resultados foram apresentados na seção 5.4, obteve-se em geral acurácia no conjunto de validação variando entre 94,15% e 98,94%, enquanto o tempo necessário para realizar a predição no conjunto de validação variou entre 0,023s e 0,040s.

O método C foi o primeiro no qual se utilizou tanto dados do *dataset 1* quanto do *dataset 2*, em geral, o que se observou nele também refletiu em outros métodos. Nos testes realizados foi possível observar um aumento considerável do tempo de treinamento, o que é justificável tendo em vista que dobrou-se o número de amostras para treinamento dos modelos. Para esse método entre 1/16 e 1 ciclo pós falta bons resultados foram obtidos com a acurácia de validação variando entre 94,68% e 98,40%, e o tempo necessário para realizar a validação entre 0,023s e 0,053s. No entanto, resultados não tão satisfatórios foram obtidos com 1/32, 1/64 e 1/128 ciclo pós falta. Isso reflete a diferença entre os conjuntos principalmente nessas faixas de ciclo pós falta. Como mostrado na seção 5.1, para o *dataset 2* com a diminuição da quantidade de ciclos pós falta *clusters* com os diferentes tipos de faltas são formados, isso dificulta bastante qualquer modelo fazer uma classificação assertiva. Uma tentativa para sanar isso foi utilizar mais dados do *dataset 2* para treinar os modelos, mas ainda assim os resultados não foram satisfatórios. Dessa forma, para esse método obteve-se as seguintes acurácias de validação para sinais com 1/32, 1/64 e 1/128 ciclo pós falta: 69,68%; 61,70%; 51,86%; respectivamente.

Nos demais métodos decidiu-se por utilizar os sinais de corrente para realizar uma comparação com o desempenho dos modelos que utilizaram sinais de tensão. O método D possui as mesmas etapas e técnicas que foram utilizadas para o método B com a diferença de ter sido utilizado os sinais de corrente do *dataset 1*. Nesse caso, a acurácia de validação variou entre 90,96% e 100% com o tempo para validação variando entre 0,012s e 0,045s. Portanto, quando esse método é comparado com o método B é possível notar que a acurácia de validação varia mais com a variação do ciclo pós falta. É possível notar principalmente que para uma quantidade menor de ciclo pós falta (1/64 e 1/128) a acurácia obtida foi menor, enquanto utilizando-se mais sinal pós falta (1, 1/2, 1/4) obteve-se maior acurácia no método D do que no método B. A comparação entre os dois métodos é apresentada na Tabela 5.12.

Tabela 5.12 – Comparação entre os resultados dos métodos B e D.

Ciclo	Método B		Método D	
	Acurácia val.	Tempo val.	Acurácia val.	Tempo val.
1	97,87%	0,031s	99,74%	0,045s
1/2	98,94%	0,023s	100,00%	0,032s
1/4	95,21%	0,040s	97,87%	0,018s
1/8	97,87%	0,023s	97,87%	0,019s
1/16	96,81%	0,032s	95,74%	0,012s
1/32	94,15%	0,023s	95,21%	0,021s
1/64	96,28%	0,026s	93,62%	0,015s
1/128	95,74%	0,032s	90,96%	0,032s

Fonte: Do autor (2021).

O comportamento citado no parágrafo anterior também é observado ao se comparar os métodos C e E. Ambos utilizaram filtro *notch* e *random forest* com uma combinação entre o *dataset 1* e *dataset 2*, a única diferença é que no primeiro foram usados os sinais de tensão, enquanto no segundo utilizou-se os sinais de corrente. No método E a acurácia no conjunto de validação variou entre 92,82% e 99,3% considerando entre 1/16 e 1 ciclo pós falta. Para 1/32, 1/64 e 1/128 ciclo pós falta novamente não foram obtidos bons resultados, respectivamente a acurácia nesses casos foi: 50,80%; 57,98%; 67,82%. Assim, de maneira análoga ao que foi observado anteriormente, ao se usar os sinais de corrente como entrada obteve-se uma acurácia levemente superior para uma quantidade de sinal pós falta maior, e uma acurácia um pouco menor para uma quantidade menor de sinal pós falta. A comparação entre os resultados obtidos pelos métodos C e E é apresentada na Tabela 5.13.

Tabela 5.13 – Comparação entre os resultados dos métodos C e E.

Ciclo	Método C		Método E	
	Acurácia val.	Tempo val.	Acurácia val.	Tempo val.
1	98,14%	0,053s	99,73%	0,055s
1/2	98,40%	0,048s	99,73%	0,035s
1/4	97,61%	0,036s	99,73%	0,039s
1/8	96,54%	0,023s	98,67%	0,033s
1/16	94,68%	0,040s	92,82%	0,025s
1/32	69,68%	0,031s	67,82%	0,043s
1/64	61,70%	0,033s	57,98%	0,026s
1/128	51,86%	0,015s	50,80%	0,010s

Fonte: Do autor (2021).

Os dois últimos métodos implementados possuem maiores diferença em relação aos que foram relatados até o momento. A principal diferença reside no fato da utilização das técnicas

Rocket e *MiniRocket* para transformação dos dados e o classificador de regressão *ridge* para identificação das faltas. Por fazerem uso de conceitos de *deep learning* para extração de características e pré-processamento dos sinais, nesse caso não foi utilizado o filtro *notch* e somente realizou-se a normalização dos dados. Cabe destacar que para para esses métodos variou-se a quantidade de *kernels* quando *Rocket* foi utilizado, e a quantidade de *features* no caso do *MiniRocket*. Na seção 5.8 e 5.9, são apresentados os melhores resultados obtidos considerando-se a acurácia no treinamento e validação. No apêndice B encontram-se algumas tabelas que apresentam o resultados variando-se a quantidade de *features* e *kernels*.

O método F usou como entrada os sinais de corrente somente do *dataset 1*. Na literatura não notou-se uma justificativa clara do uso de corrente ou tensão na classificação de faltas. Alguns autores usam um deles ou até uma combinação dos dois. Dessa maneira, em alguns métodos foi usada a tensão e em outros a corrente para efeito de comparação. Para extração de *features* foram utilizadas as transformações *Rocket* e *MiniRocket*, dessa forma foi possível comparar essas duas técnicas e verificar qual teve melhor desempenho. Utilizando-se *Rocket* foi possível verificar que a acurácia de validação variou entre 94,68% e 100%, e o tempo necessário para classificar o conjunto de validação variou entre 0,040s e 0,106s. Comparativamente, utilizando-se o *MiniRocket* a acurácia variou entre 97,87% e 100% enquanto o tempo de validação ficou entre 0,011s e 0,054. Assim, observa-se um desempenho levemente superior ao se utilizar *MiniRocket*, já que essa técnica obteve uma acurácia levemente superior com uma quantidade menor de sinal pós falta, além de ser mais rápida no processamento dos dados de validação. Além disso, foi possível observar um desempenho levemente superior ao se classificar uma única amostra utilizando-se *MiniRocket*, como pode-se verificar no seção 5.8. Em geral, menores valores foram obtidos para o tempo operacional usando ela quando comparado ao uso da técnica *Rocket*.

Tendo em vista que tanto o método D quanto o método F utilizam os mesmo sinais de corrente, é possível realizar uma comparação entre eles para verificar qual modelo tem melhor desempenho. Pelo o que foi exposto é possível notar que foram obtidos resultados de acurácia levemente superiores utilizando tanto *Rocket* quanto *MiniRocket* com o classificador de regressão *ridge* no método F em comparação ao uso de filtro *notch* e florestas aleatórias do método D. A comparação entre esses métodos é mostrada na Tabela 5.14. Deve-se destacar que há um ganho considerável no tempo operacional ao se utilizar *MiniRocket* no método F em relação às técnicas usadas no método D, como mostrado na Tabela 5.15.

Tabela 5.14 – Comparação entre os resultados dos métodos F e D.

Ciclo	Método F				Método D	
	<i>Rocket</i>		<i>MiniRocket</i>		Acurácia val.	Tempo val.
	Acurácia val.	Tempo val.	Acurácia val.	Tempo val.		
1	100,00%	0,040s	100,00%	0,011s	99,74%	0,045s
1/2	100,00%	0,048s	100,00%	0,020s	100,00%	0,032s
1/4	100,00%	0,041s	100,00%	0,045s	97,87%	0,018s
1/8	100,00%	0,052s	100,00%	0,034s	97,87%	0,019s
1/16	98,40%	0,076s	99,47%	0,020s	95,74%	0,012s
1/32	98,40%	0,106s	98,94%	0,037s	95,21%	0,021s
1/64	96,81%	0,077s	97,87%	0,017s	93,62%	0,015s
1/128	94,68%	0,067s	97,87%	0,054s	90,96%	0,032s

Fonte: Do autor (2021).

Tabela 5.15 – Comparação entre os tempos operacionais (para classificar uma única amostra) mínimo e máximo para os métodos F e D.

Ciclo	Método F				Método D	
	<i>Rocket</i>		<i>MiniRocket</i>		Tempo oper. mín.	Tempo máx.
	Tempo mín.	Tempo máx.	Tempo mín.	Tempo máx.		
1	0,014s	0,015s	0,012s	0,014s	0,069s	0,070s
1/2	0,014s	0,014s	0,012s	0,012s	0,045s	0,046s
1/4	0,013s	0,015s	0,012s	0,012s	0,030s	0,030s
1/8	0,014s	0,030s	0,012s	0,012s	0,025s	0,026s
1/16	0,018s	0,019s	0,011s	0,011s	0,020s	0,020s
1/32	0,019s	0,021s	0,012s	0,012s	0,022s	0,022s
1/64	0,021s	0,026s	0,011s	0,011s	0,019s	0,019s
1/128	0,022s	0,023s	0,014s	0,014s	0,027s	0,027s

Fonte: Do autor (2021).

O método G foi o último a ser implementado, nele foram usados os sinais de corrente do *dataset 1* e *dataset 2* como entrada dos modelos. Para geração de *features* novamente utilizou-se *Rocket* e *MiniRocket*. Utilizando-se *Rocket* e considerando de 1/16 a 1 ciclo pós falta a acurácia obtida ficou entre 95,15% e 100%. Assim como nos métodos anteriores que usam os dois conjuntos de dados o desempenho dos modelos ficou abaixo do esperado, para 1/32 ciclo pós falta a acurácia de validação foi 66,49%, para 1/64 foi 56,12% e para 1/128 foi 50,80%. Por sua vez, ao utilizar *MiniRocket* a performance em acurácia ficou bem parecida variando entre 93,09% e 100% para 1/16 e 1 ciclo pós falta. A maior diferença entre as técnicas nesse caso ficou a cargo do tempo necessário para classificar o conjunto de validação com *Rocket* variando entre 0,023s e 0,105s, enquanto *MiniRocket* ficou entre 0,031s e 0,066s.

Tendo em vista que os métodos E e G utilizam dados de corrente tanto do *dataset 1* quanto do *dataset 2*, podemos fazer uma comparação entre eles, para observar se existe alguma

diferença ao utilizar alguma das técnicas. Pelo o que foi apresentado, observa-se que existe uma diferença mínima, mas não muito significativa, obtendo-se uma acurácia levemente superior no método G ao considerar maiores quantidade de ciclo pós falta (1/16, 1/8, 1/4, 1/2, 1), a comparação entre esses métodos encontra-se na Tabela 5.16. Além disso, como é possível notar pela Tabela 5.17, novamente se obtém um ganho no tempo operacional com a utilização da técnica *MiniRocket* e classificador de regressão *ridge* em comparação ao filtro *notch* e *random forest*.

Tabela 5.16 – Comparação entre os resultados dos métodos G e E.

Ciclo	Método G				Método E	
	<i>Rocket</i>		<i>MiniRocket</i>		Acurácia val.	Tempo val.
	Acurácia val.	Tempo val.	Acurácia val.	Tempo val.		
1	100,00%	0,023s	100,00%	0,044s	99,73%	0,055s
1/2	100,00%	0,051s	100,00%	0,031s	99,73%	0,035s
1/4	100,00%	0,072s	99,73%	0,048s	99,73%	0,039s
1/8	98,40%	0,072s	99,47%	0,066s	98,67%	0,033s
1/16	94,15%	0,102s	93,09%	0,060s	92,82%	0,025s
1/32	66,49%	0,105s	65,96%	0,051s	67,82%	0,043s
1/64	56,12%	0,092s	51,71%	0,057s	57,98%	0,026s
1/128	50,80%	0,103s	55,59%	0,043s	50,80%	0,010s

Fonte: Do autor (2021).

Tabela 5.17 – Comparação entre os tempos operacionais (para classificar uma única amostra) mínimo e máximo para os métodos G e E.

Ciclo	Método G				Método E	
	<i>Rocket</i>		<i>MiniRocket</i>		Tempo oper. mín.	Tempo máx.
	Tempo mín.	Tempo máx.	Tempo mín.	Tempo máx.		
1	0,013s	0,014s	0,014s	0,014s	0,071s	0,072s
1/2	0,016s	0,016s	0,013s	0,013s	0,046s	0,047s
1/4	0,016s	0,020s	0,012s	0,013s	0,037s	0,037s
1/8	0,019s	0,021s	0,013s	0,014s	0,029s	0,029s
1/16	0,024s	0,027s	0,013s	0,013s	0,024s	0,024s
1/32	0,024s	0,026s	0,012s	0,013s	0,030s	0,030s
1/64	0,023s	0,025s	0,012s	0,013s	0,024s	0,025s
1/128	0,027s	0,029s	0,011s	0,013s	0,017s	0,018s

Fonte: Do autor (2021).

Por fim, é possível notar que bons resultados foram obtidos ao se utilizar *Rocket* e *MiniRocket*, tanto em relação a acurácia de classificação quanto ao tempo operacional. Essas técnicas se destacam principalmente por não exigirem um conhecimento prévio em relação aos sinais, suas formas, frequências e estatísticas. Isso garante um bom funcionamento do sistema de classificação independente do sinal de entrada.

6 CONCLUSÃO

Esse trabalho investigou o uso de diversas técnicas de aprendizado de máquina na classificação de faltas em linhas de transmissão. O objetivo foi comparar o uso de modelos clássicos e de menor complexidade com técnicas mais recentes, além da variação dos conjuntos de dados utilizados. Por esse motivo, foram gerados diversos métodos distintos que foram nomeados de A a G.

É possível concluir com o trabalho realizado nessa dissertação que ao incluir os dados de linhas com compensação (*dataset 2*) existe uma queda de desempenho nos modelos (menor acurácia) quando se utiliza uma menor quantidade de ciclo pós falta, principalmente 1/32, 1/64 e 1/128. Cabe destacar ainda que dois testes foram realizados, mas como a princípio não se obteve uma melhora considerável não foi dada continuidade a eles. O teste inicial foi realizar somente o treinamento com o *dataset 1* e validar os modelos com o *dataset 2*, mas a acurácia nesse caso ficou entre 40% e 50%. Como foi destacado na seção 5.1, existem diferenças consideráveis entre os *datasets* no espaço bidimensional. Para o *dataset 2* a diminuição da quantidade de ciclo pós falta também induz a uma aglomeração em *clusters* de diferentes tipos de faltas o que dificulta o treinamento e generalização dos modelos. O segundo teste conduzido foi realizar o treinamento de modelos específicos somente com os dados do *dataset 2*, mas ainda assim não se obteve alguma melhoria significativa. Por esses motivos decidiu-se proceder com o treinamento dos métodos utilizando ambos conjuntos de dados, o que conferiu uma melhoria no desempenho para uma maior quantidade de ciclo pós falta, mas para 1/32, 1/64, 1/128 os modelos ainda não foram capazes de generalizar pelos motivos citados acima.

Observou-se ainda que a utilização dos sinais de corrente conferiram uma acurácia levemente superior na maioria dos testes realizados. Isso pode ser um indicativo de que a corrente é melhor para discriminar os tipos de falta. No entanto, a utilização conjunta de tensão e corrente pode ser capaz de trazer um ganho ainda melhor para os modelos. Nessa dissertação focou-se em utilizar os sinais individualmente para ter na entrada dos modelos uma quantidade menor de *features* visando possíveis aplicações práticas dos métodos implementados.

Outra análise realizada foi a comparação entre a utilização das técnicas *Rocket* e *MiniRocket*. Nesse caso, observou-se que o desempenho de acurácia entre elas foi bem similar, não se obtendo alguma diferença significativa. No entanto, ao verificar o tempo para treinamento e validação, *MiniRocket* é, em geral, mais rápida. Isso se deve ao fato da forma diferente de construção dos dois métodos como foi explicado na seção 3.2 e 3.3. Mas além disso, a quantidade

de *features* utilizada também interfere no tempo de processamento tendo em vista que *Rocket* extrai duas *features* por *kernel*, quanto *MiniRocket* extrai apenas uma.

Por fim, comparando-se *Rocket* e *MiniRocket* com *random forest* pouca diferença foi notada. Para o método F que utiliza somente o *dataset 1* houve uma leve diferença na acurácia, sendo que o modelo de florestas aleatórias foi levemente inferior. Em relação ao método G que utiliza os dois *datasets* praticamente não existe diferença no nível da acurácia obtida. No entanto, deve-se destacar que considerando o tempo operacional (tempo para classificar uma única amostra), em geral melhores resultados foram obtidos utilizando-se *MiniRocket*, isso pode ser uma vantagem considerando-se aplicações em tempo real.

6.1 Propostas de trabalhos futuros

Algumas propostas para melhoria do trabalho apresentado nessa dissertação e que podem ser implementadas no futuro encontram-se a seguir.

- Utilização de outros métodos de processamento de sinais. Por exemplo, a análise de componentes independentes (*independent component analysis* - ICA) já foi utilizada em problemas de localização de faltas e talvez possa ser usada também em problemas de classificação;
- retrainar os modelos utilizando conjuntamente dados de corrente e tensão para verificar como eles se comportam e se ocorre algum ganho;
- utilizar alguma metodologia que foque em linhas de transmissão com compensação, especialmente quando há uma menor quantidade de ciclo pós falta. Nesse trabalho observou-se que o comportamento desses sinais com 1/32, 1/64 e 1/128 ciclo pós falta é muito divergente dificultando o treinamento e generalização dos modelos;
- variar a faixa de ruído para verificar se há impacto no modelo se a relação sinal-ruído é muito baixa, nessa dissertação fixou-se o valor do ruído em 60 dB;
- realização do treinamento e validação dos modelos com dados reais;
- implementação dos modelos em tempo real.

6.2 Publicações

FERREIRA, D. D. et al. Classificação de faltas em linhas de transmissão utilizando florestas aleatórias e filtro notch. In: Anais do Congresso Brasileiro de Automática 2020. [S.l.]: sbabra, 2020.

FERREIRA, D. D. et al. Classificação de faltas em linhas de transmissão utilizando métodos de inteligência computacional. In: Anais do Simpósio Brasileiro de Sistemas Elétricos 2020.[S.l.]: sbabra, 2020.

FONSECA, G. A. et al. Fault classification in transmission lines using random forest and notch filter. Journal of Control, Automation and Electrical Systems, Nov 2021. Disponível em: <<https://doi.org/10.1007/s40313-021-00844-4>>.

7 EQUIPE TÉCNICA

As informações sobre a equipe técnica que está envolvida diretamente no projeto é descrita a seguir:

Aluno: Gabriel Aparecido Fonseca

Titulação: Bacharel em Engenharia Mecatrônica no Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)

Resumo curricular: Mestrando em Engenharia de Sistemas e Automação na Universidade Federal de Lavras (UFLA) e bacharel em Engenharia Mecatrônica pelo CEFET-MG.

Orientador: Danton Diego Ferreira

Titulação: Doutor pela Universidade Federal do Rio de Janeiro (UFRJ)

Resumo curricular: Possui graduação em Engenharia Industrial Elétrica pela Universidade Federal de São João del-Rei (UFSJ) (2005), Mestrado em Engenharia Elétrica pela Universidade Federal de Juiz de Fora (UFJF) (2007), Doutorado na COPPE/UFRJ (2010) e Pós-doutorado na UFJF (2014). Atualmente é Professor adjunto em regime de dedicação exclusiva na UFLA.

Coorientador: Flávio Bezerra Costa

Titulação: Doutor pela Universidade Federal de Campina Grande (UFCG).

Resumo curricular: Possui graduação em Engenharia Elétrica pela UFCG (2004), Mestrado em Engenharia Elétrica pela UFCG (2006), Doutorado na UFCG (2010), Pós-doutorado na UFCG (2010) e RWTH Aachen, Alemanha (2015). Pesquisador visitante em K.U.Leuven (2012) e INESC Porto (2014). Professor visitante em TU Berlin (2019). Atualmente é Professor de Engenharia Elétrica na Escola de Ciência e Tecnologia (ECT) na Universidade Federal do Rio Grande do Norte (UFRN).

Coorientador: Aryfrance Rocha Almeida

Titulação: Doutor pela Universidade Federal do Ceará (UFC).

Resumo curricular: Possui graduação em Engenharia Elétrica pela Universidade Federal do Maranhão (UFMA) (2005), Mestrado em Engenharia Elétrica pela UFMA (2007) e Doutorado pela UFC (2017). Atualmente é Professor Adjunto II no curso de Engenharia Elétrica da Universidade Federal do Piauí (UFPI), na área de Sistemas Elétricos.

APÊNDICE A – Resultados dos métodos implementado

A.1 Método B

A.1.1 1 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 97,21%.
- O desvio padrão da acurácia nos 10 *fold*s foi 1,62%.
- O tempo total necessário para treinamento foi 16,745 segundos.
- A acurácia no conjunto de validação foi 97,87%.
- O tempo necessário para classificação do conjunto de validação foi 0,031 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 94,74%
 - *Fold* 2: 100,00%
 - *Fold* 3: 97,33%
 - *Fold* 4: 94,67%
 - *Fold* 5: 98,67%
 - *Fold* 6: 97,33%
 - *Fold* 7: 98,67%
 - *Fold* 8: 96,00%
 - *Fold* 9: 97,33%
 - *Fold* 10: 97,33%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta AB com 0,134s.
 - O tipo de falta com menor tempo operacional foi a falta BC com 0,066s.

A Tabela 1 apresenta a matriz de confusão gerada pelo método B treinado com sinais com 1 ciclo pós falta.

Tabela 1 – Matriz de confusão do método B com 1 ciclo pós falta.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	100%									
		19/19									
	ABC		100%								
			18/18								
	ABT			94,74%	5,26%						
				18/19	1/19						
	AT				5,26%	94,74%					
				1/19	18/19						
	BC	5,26%				94,74%					
		1/19				18/19					
BCT						100%					
						19/19					
BT							100%				
							19/19				
CA								100%			
								19/19			
CAT									100%		
									18/18		
CT							5,26%			94,74%	
							1/19			18/19	

Do autor (2021).

A.1.2 1/2 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 97,48%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,50%.
- O tempo total necessário para treinamento foi 10,374 segundos.
- A acurácia no conjunto de validação foi 98,94%.
- O tempo necessário para classificação do conjunto de validação foi 0,023 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 94,74%
 - *Fold* 2: 98,68%

- *Fold 3*: 97,33%
 - *Fold 4*: 97,33%
 - *Fold 5*: 98,67%
 - *Fold 6*: 94,67%
 - *Fold 7*: 98,67%
 - *Fold 8*: 97,33%
 - *Fold 9*: 98,67%
 - *Fold 10*: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta BCT com 0,043s.
 - O tipo de falta com menor tempo operacional é a falta CAT com 0,042s.

A Tabela 2 apresenta a matriz de confusão gerada pelo método B treinado com sinais com 1/2 ciclo pós falta.

Tabela 2 – Matriz de confusão do método B com 1/2 ciclo pós falta.

		Valores preditos										
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT	
Valores reais	AB	100%										
		19/19										
	ABC		100%									
			18/18									
	ABT			94,74%	5,26%							
				18/19	1/19							
	AT				100%							
					19/19							
	BC					100%						
						19/19						
BCT						100%						
						19/19						
BT			5,26%				94,74%					
			1/19				18/19					
CA								100%				
								19/19				
CAT									100%			
									18/18			
CT										100%		
										19/19		

Do autor (2021).

A.1.3 1/4 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 97,34%.
- O desvio padrão da acurácia nos 10 *fold*s foi 1,68%.
- O tempo total necessário para treinamento foi 12,772 segundos.
- A acurácia no conjunto de validação foi 95,21%.
- O tempo necessário para classificação do conjunto de validação foi 0,04 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 94,74%
 - *Fold* 2: 97,37%
 - *Fold* 3: 98,67%

- *Fold 4*: 98,67%
 - *Fold 5*: 94,67%
 - *Fold 6*: 100,00%
 - *Fold 7*: 97,33%
 - *Fold 8*: 96,00%
 - *Fold 9*: 97,33%
 - *Fold 10*: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CAT com 0,038s.
 - O tipo de falta com menor tempo operacional foi a falta ABT com 0,038s.

A Tabela 3 apresenta a matriz de confusão gerada pelo método B treinado com sinais com 1/4 ciclo pós falta.

Tabela 3 – Matriz de confusão do método B com 1/4 ciclo pós falta.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	78,94%							21,05%		
		15/19							4/19		
	ABC		100%								
			18/18								
	ABT			100%							
				19/19							
	AT			5,26%	89,48%					5,26%	
				1/19	17/19					1/19	
	BC					84,21%			15,79%		
						16/19			3/19		
	BCT						100%				
						19/19					
BT							100%				
							19/19				
CA								100%			
								19/19			
CAT									100%		
									18/18		
CT										100%	
										19/19	

Do autor (2021).

A.1.4 1/8 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 97,61%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,00%.
- O tempo total necessário para treinamento foi 6,931 segundos.
- A acurácia no conjunto de validação foi 97,87%.
- O tempo necessário para classificação do conjunto de validação foi 0,023 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 97,37%
 - *Fold* 2: 98,68%
 - *Fold* 3: 98,67%
 - *Fold* 4: 98,67%
 - *Fold* 5: 97,33%
 - *Fold* 6: 96,00%
 - *Fold* 7: 97,33%
 - *Fold* 8: 96,00%
 - *Fold* 9: 97,33%
 - *Fold* 10: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta AB com 0,026s.
 - O tipo de falta com menor tempo operacional foi a falta CT com 0,025s.

A Tabela 4 apresenta a matriz de confusão gerada pelo método B treinado com sinais com 1/8 ciclo pós falta.

Tabela 4 – Matriz de confusão do método B com 1/8 ciclo pós falta.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	89,47%							10,53%		
		17/19							2/19		
	ABC		100%								
			18/18								
	ABT			100%							
				19/19							
	AT			5,26%	94,74%						
				1/19	18/19						
	BC					100%					
						19/19					
	BCT						94,74%	5,26%			
						18/19	1/19				
BT							100%				
							19/19				
CA								100%			
								19/19			
CAT									100%		
									18/18		
CT										100%	
										19/19	

Do autor (2021).

A.1.5 1/16 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 97,74%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,04%.
- O tempo total necessário para treinamento foi 14,045 segundos.
- A acurácia no conjunto de validação foi 96,81%.
- O tempo necessário para classificação do conjunto de validação foi 0,032 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 97,37%
 - *Fold* 2: 98,68%
 - *Fold* 3: 97,33%
 - *Fold* 4: 96,00%

- *Fold 5*: 98,67%
 - *Fold 6*: 96,00%
 - *Fold 7*: 98,67%
 - *Fold 8*: 97,33%
 - *Fold 9*: 98,67%
 - *Fold 10*: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CAT com 0,029s.
 - O tipo de falta com menor tempo operacional foi a falta AT com 0,028s.

A Tabela 5 apresenta a matriz de confusão gerada pelo método B treinado com sinais com 1/16 ciclo pós falta.

Tabela 5 – Matriz de confusão do método B com 1/16 ciclo pós falta.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	94,74%							5,26%		
		18/19							1/19		
	ABC		100%								
			18/18								
	ABT			100%							
				19/19							
	AT			5,26%	94,74%						
				1/19	18/19						
	BC	15,79%				84,21%					
		3/19				16/19					
BCT						94,74%	5,26%				
						18/19	1/19				
BT								100%			
								19/19			
CA									100%		
									19/19		
CAT										100%	
										18/18	
CT											100%
											19/19

A.1.6 1/32 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 97,08%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,54%.
- O tempo total necessário para treinamento foi 4.687 segundos.
- A acurácia no conjunto de validação foi 94,15%.
- O tempo necessário para classificação do conjunto de validação foi 0,023 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 97,74%
 - *Fold* 2: 97,37%
 - *Fold* 3: 98,67%
 - *Fold* 4: 96,00%
 - *Fold* 5: 100,00%
 - *Fold* 6: 96,00%
 - *Fold* 7: 97,33%
 - *Fold* 8: 96,00%
 - *Fold* 9: 96,00%
 - *Fold* 10: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta BT com 0,024s.
 - O tipo de falta com menor tempo operacional foi a falta AB com 0,024s.

A Tabela 6 apresenta a matriz de confusão gerada pelo método B treinado com sinais com 1/32 ciclo pós falta.

Tabela 6 – Matriz de confusão do método B com 1/32 ciclo pós falta.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	78,95%							21,05%		
		15/19							4/19		
	ABC		100%								
			18/18								
	ABT			100%							
				19/19							
	AT			5,26%	94,74%						
				1/19	18/19						
	BC	15,79%				84,21%					
		3/19				16/19					
BCT						94,74%	5,26%				
						18/19	1/19				
BT						5,26%	94,74%				
						1/19	18/19				
CA								100%			
								19/19			
CAT									100%		
									18/18		
CT							5,26%			94,74%	
							1/19			18/19	

Do autor (2021).

A.1.7 1/64 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 96,94%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,86%.
- O tempo total necessário para treinamento foi 10,672 segundos.
- A acurácia no conjunto de validação foi 96,28%.
- O tempo necessário para classificação do conjunto de validação foi 0,026 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 97,37%
 - *Fold* 2: 97,37%
 - *Fold* 3: 98,67%
 - *Fold* 4: 96,00%

- *Fold 5*: 97,33%
 - *Fold 6*: 97,33%
 - *Fold 7*: 96,00%
 - *Fold 8*: 97,33%
 - *Fold 9*: 96,00%
 - *Fold 10*: 96,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta ABC com 0,024s.
 - O tipo de falta com menor tempo operacional foi a falta CA com 0,024s.

A Tabela 7 apresenta a matriz de confusão gerada pelo método B treinado com sinais com 1/64 ciclo pós falta.

Tabela 7 – Matriz de confusão do método B com 1/64 ciclo pós falta.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	100%									
		19/19									
	ABC		100%								
			18/18								
	ABT			100%							
				19/19							
	AT			5,26%	94,74%						
				1/19	18/19						
	BC	15,79%				84,21%					
		3/19				16/19					
BCT						94,74%	5,26%				
						18/19	1/19				
BT								100%			
								19/19			
CA					5,26%			94,74%			
					1/19			18/19			
CAT									100%		
									18/18		
CT						5,26%				94,74%	
						1/19				18/19	

Do autor (2021).

A.1.8 1/128 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 96,15%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,50%.
- O tempo total necessário para treinamento foi 4,841 segundos.
- A acurácia no conjunto de validação foi 95,74%.
- O tempo necessário para classificação do conjunto de validação foi 0,032 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 93,42%
 - *Fold* 2: 97,37%
 - *Fold* 3: 97,33%
 - *Fold* 4: 96,00%
 - *Fold* 5: 97,33%
 - *Fold* 6: 94,67%
 - *Fold* 7: 96,00%
 - *Fold* 8: 96,00%
 - *Fold* 9: 94,67%
 - *Fold* 10: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta AB com 0,026s.
 - O tipo de falta com menor tempo operacional foi a falta BCT com 0,026s.

A Tabela 8 apresenta a matriz de confusão gerada pelo método B treinado com sinais com 1/128 ciclo pós falta.

Tabela 8 – Matriz de confusão do método B com 1/128 ciclo pós falta.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100%											
		19/19											
	ABC		100%										
			18/18										
	ABT			94,74%	5,26%								
				18/19	1/19								
	AT				5,26%	94,74%							
					1/19	18/19							
	BC	15,79%				84,21%							
		3/19				16/19							
BCT						94,74%	5,26%						
						18/19	1/19						
BT								100%					
								19/19					
CA					5,26%			94,74%					
					1/19			18/19					
CAT									100%				
									18/18				
CT							5,26%				94,74%		
							1/19				18/19		

Do autor (2021).

A.2 Método C

A.2.1 1 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 98,20%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,12%.
- O tempo total necessário para treinamento foi 72,014 segundos.
- A acurácia no conjunto de validação foi 98,14%.
- O tempo necessário para classificação do conjunto de validação foi 0,053 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 98,03%
 - *Fold* 2: 100,00%
 - *Fold* 3: 99,33%

- *Fold 4*: 96,00%
 - *Fold 5*: 98,67%
 - *Fold 6*: 96,67%
 - *Fold 7*: 98,67%
 - *Fold 8*: 98,00%
 - *Fold 9*: 98,67%
 - *Fold 10*: 98,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CAT com 0,073s.
 - O tipo de falta com menor tempo operacional foi a falta ABT com 0,071s.

A Tabela 9 apresenta a matriz de confusão gerada pelo método C treinado com sinais com 1 ciclo pós falta.

Tabela 9 – Matriz de confusão do método C com 1 ciclo pós falta.

		Valores preditos										
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT	
Valores reais	AB	100%										
		38/38										
	ABC		91,67%			8,33%						
			33/36			3/36						
	ABT			97,37%	2,63%							
				37/38	1/38							
	AT				100%							
					38/38							
	BC					100%						
						38/38						
BCT						100%						
						38/38						
BT							97,37%			2,63%		
							37/38			1/38		
CA								100%				
								38/38				
CAT									100%			
									36/36			
CT						5,26%				94,74%		
						2/38				36/38		

Do autor (2021).

A.2.2 1/2 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 98,80%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,65%.
- O tempo total necessário para treinamento foi 32,569 segundos.
- A acurácia no conjunto de validação foi 98,40%.
- O tempo necessário para classificação do conjunto de validação foi 0,048 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 97,37%
 - *Fold* 2: 99,34%
 - *Fold* 3: 99,33%
 - *Fold* 4: 99,33%
 - *Fold* 5: 99,33%
 - *Fold* 6: 98,67%
 - *Fold* 7: 99,33%
 - *Fold* 8: 98,00%
 - *Fold* 9: 98,67%
 - *Fold* 10: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta ABC com 0,053s.
 - O tipo de falta com menor tempo operacional foi a falta BC com 0,051s.

A Tabela 10 apresenta a matriz de confusão gerada pelo método C treinado com sinais com 1/2 ciclo pós falta.

Tabela 10 – Matriz de confusão do método C com 1/2 ciclo pós falta.

		Valores preditos										
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT	
Valores reais	AB	100%										
		38/38										
	ABC	2,78%	94,44%			2,78%						
		1/36	34/36			1/36						
	ABT			100%								
				38/38								
	AT			2,63%	97,37%							
				1/38	37/38							
	BC		2,63%			97,37%						
			1/38			37/38						
	BCT					97,37%	2,63%					
						37/38	1/38					
BT						97,37%				2,63%		
						37/38				1/38		
CA								100%				
								38/38				
CAT									100%			
									36/36			
CT										100%		
										38/38		

Do autor (2021).

A.2.3 1/4 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 97,14%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,59%.
- O tempo total necessário para treinamento foi 21,71 segundos.
- A acurácia no conjunto de validação foi 97,61%.
- O tempo necessário para classificação do conjunto de validação foi 0,036 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 93,42%
 - *Fold* 2: 98,68%
 - *Fold* 3: 95,33%
 - *Fold* 4: 97,33%
 - *Fold* 5: 98,67%
 - *Fold* 6: 98,00%

- *Fold 7*: 96,67%
 - *Fold 8*: 97,33%
 - *Fold 9*: 97,33%
 - *Fold 10*: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta ABT com 0,038s.
 - O tipo de falta com menor tempo operacional foi a falta ABC com 0,037s.

A Tabela 11 apresenta a matriz de confusão gerada pelo método C treinado com sinais com 1/4 ciclo pós falta.

Tabela 11 – Matriz de confusão do método C com 1/4 ciclo pós falta.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	100%									
		38/38									
	ABC	5,55%	91,67%						2,78%		
		2/36	33/36						1/36		
	ABT			97,37%	2,63%						
				37/38	1/38						
	AT			2,63%	97,37%						
				1/38	37/38						
	BC	5,26%				94,74%					
		2/38				36/38					
	BCT						100%				
						38/38					
BT							100%				
							38/38				
CA								100%			
								38/38			
CAT									94,44%	5,56%	
									34/36	2/36	
CT										100%	
										38/38	

Do autor (2021).

A.2.4 1/8 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 96,34%.

- O desvio padrão da acurácia nos 10 *folds* foi 1,16%.
- O tempo total necessário para treinamento foi 19,646 segundos.
- A acurácia no conjunto de validação foi 96,54%.
- O tempo necessário para classificação do conjunto de validação foi 0,023 segundos.
- Acurácia em cada *fold* :
 - *Fold* 1: 96,71%
 - *Fold* 2: 94,74%
 - *Fold* 3: 96,00%
 - *Fold* 4: 96,00%
 - *Fold* 5: 97,33%
 - *Fold* 6: 94,00%
 - *Fold* 7: 97,33%
 - *Fold* 8: 96,67%
 - *Fold* 9: 98,00%
 - *Fold* 10: 96,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta AB com 0,027s.
 - O tipo de falta com menor tempo operacional foi a falta BCT com 0,026s.

A Tabela 12 apresenta a matriz de confusão gerada pelo método C treinado com sinais com 1/8 ciclo pós falta.

Tabela 12 – Matriz de confusão do método C com 1/8 ciclo pós falta.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	92,11% 35/38				2,63% 1/38			5,26% 2/38		
	ABC		94,44% 34/36			2,78% 1/36			2,78% 1/36		
	ABT			97,37% 37/38				2,63% 1/38			
	AT			7,89% 3/38	92,11% 35/38						
	BC	2,63% 1/38				97,37% 37/38					
	BCT						97,37% 37/38				2,63% 1/38
	BT							100% 38/38			
	CA	2,63% 1/38							97,37% 37/38		
	CAT									100% 36/36	
	CT						2,63% 1/38				97,37% 37/38

Do autor (2021).

A.2.5 1/16 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 93,62%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,53%.
- O tempo total necessário para treinamento foi 40,766 segundos.
- A acurácia no conjunto de validação foi 94,68%.
- O tempo necessário para classificação do conjunto de validação foi 0,04 segundos.
- Acurácia em cada fold:
 - *Fold* 1: 94,08%
 - *Fold* 2: 90,79%
 - *Fold* 3: 92,00%
 - *Fold* 4: 92,67%

- *Fold 5*: 94,67%
 - *Fold 6*: 93,33%
 - *Fold 7*: 96,00%
 - *Fold 8*: 92,67%
 - *Fold 9*: 95,33%
 - *Fold 10*: 94,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta ABC com 0,032s.
 - O tipo de falta com menor tempo operacional foi a falta BC com 0,030s.

A Tabela 13 apresenta a matriz de confusão gerada pelo método C treinado com sinais com 1/16 ciclo pós falta.

Tabela 13 – Matriz de confusão do método C com 1/16 ciclo pós falta.

		Valores preditos												
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT			
Valores reais	AB	89,47%				10,53%								
		34/38				4/38								
	ABC	2,78%	97,22%											
		1/36	35/36											
	ABT	2,63%		97,37%										
		1/38		37/38										
	AT			2,63%	97,37%									
				1/38	37/38									
	BC					94,74%			5,26%					
						36/38			2/38					
	BCT						92,11%	5,26%					2,63%	
							35/38	2/38					1/38	
	BT			5,26%			2,63%	92,11%						
				2/38			1/38	35/38						
CA		2,63%							97,37%					
		1/38							37/38					
CAT				2,78%						94,44%		2,78%		
				1/36						34/36		1/36		
CT							5,26%					94,74%		
							2/38					36/38		

Do autor (2021).

A.2.6 1/32 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 69,34%.
- O desvio padrão da acurácia nos 10 *folds* foi 3,50%.
- O tempo total necessário para treinamento foi 37,186 segundos.
- A acurácia no conjunto de validação foi 69,68%.
- O tempo necessário para classificação do conjunto de validação foi 0,031 segundos.
- Acurácia em cada fold:
 - *Fold* 1: 67,11%
 - *Fold* 2: 74,34%
 - *Fold* 3: 66,67%
 - *Fold* 4: 70,67%
 - *Fold* 5: 72,67%
 - *Fold* 6: 67,33%
 - *Fold* 7: 72,00%
 - *Fold* 8: 62,00%
 - *Fold* 9: 72,00%
 - *Fold* 10: 68,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CA com 0,026s.
 - O tipo de falta com menor tempo operacional foi a falta ABC com 0,025s.

A Tabela 14 apresenta a matriz de confusão gerada pelo método C treinado com sinais com 1/32 ciclo pós falta.

Tabela 14 – Matriz de confusão do método C com 1/32 ciclo pós falta.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	71,05% 27/38	7,90% 3/38	5,26% 2/38		2,63% 1/38			13,16% 5/38		
	ABC	5,55% 2/36	75,00% 27/36			5,55% 2/36		2,78% 1/36	11,12% 4/36		
	ABT	13,16% 5/38	5,26% 2/38	65,79% 25/38	5,26% 2/38			2,63% 1/38	7,90% 3/38		
	AT	18,42% 7/38		2,63% 1/38	68,42% 26/38				7,90% 3/38	2,63% 1/38	
	BC	2,63% 1/38	7,90% 3/38			76,31% 29/38		7,90% 3/38	5,26% 2/38		
	BCT	5,26% 2/38	2,63% 1/38		2,63% 1/38	7,90% 3/38	63,15% 24/38	7,90% 3/38	7,90% 3/38	2,63% 1/38	
	BT	10,52% 4/38	2,63% 1/38	7,90% 3/38		5,26% 2/38		65,79% 25/38	7,90% 3/38		
	CA	7,90% 3/38	5,26% 2/38			2,63% 1/38		2,63% 1/38	76,31% 29/38	5,26% 2/38	
	CAT	2,78% 1/36						5,55% 2/36	19,44% 7/36	69,44% 25/36	2,78% 1/36
	CT	2,63% 1/38				13,16% 5/38	2,63% 1/38	2,63% 1/38	13,16% 5/38		65,79% 25/38

Do autor (2021).

A.2.7 1/64 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 63,97%.
- O desvio padrão da acurácia nos 10 *folds* foi 2,89%.
- O tempo total necessário para treinamento foi 42,392 segundos.
- A acurácia no conjunto de validação foi 61,70%.
- O tempo necessário para classificação do conjunto de validação foi 0,033 segundos.
- Acurácia em cada fold:
 - *Fold 1*: 58,55%
 - *Fold 2*: 64,47%
 - *Fold 3*: 64,67%
 - *Fold 4*: 62,00%
 - *Fold 5*: 65,33%

- *Fold 6*: 62,00%
 - *Fold 7*: 67,33%
 - *Fold 8*: 62,00%
 - *Fold 9*: 69,33%
 - *Fold 10*: 64,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta ABC com 0,030s.
 - O tipo de falta com menor tempo operacional foi a falta CT com 0,027s.

A Tabela 15 apresenta a matriz de confusão gerada pelo método C treinado com sinais com 1/64 ciclo pós falta.

Tabela 15 – Matriz de confusão do método C com 1/64 ciclo pós falta.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	63,15%	2,63%		5,26%	5,26%			18,42%	5,26%	
		24/38	1/38		2/38	2/38			7/38	2/38	
	ABC	8,33%	69,44%	2,78%		2,78%	2,78%	2,78%	8,33%	2,78%	
		3/36	25/36	1/36		1/36	1/36	1/36	3/36	1/36	
	ABT	2,63%	7,90%	60,53%	2,63%	10,53%			7,90%	7,90%	
		1/38	3/38	23/38	1/38	4/38			3/38	3/38	
	AT	13,16%	2,63%	2,63%	60,53%		5,26%		10,53%	5,26%	
		5/38	1/38	1/38	23/38		2/38		4/38	2/38	
	BC	7,90%	7,90%	2,63%	2,63%	57,89%	5,26%	2,63%	7,90%	2,63%	2,63%
		3/38	3/38	1/38	1/38	22/38	2/38	1/38	3/38	1/38	1/38
BCT	5,26%	5,26%	7,90%		5,26%	60,53%	5,26%	5,26%	2,63%	2,63%	
	2/38	2/38	3/38		2/38	23/38	2/38	2/38	1/38	1/38	
BT	5,26%	5,26%	5,26%	2,63%	2,63%	2,63%	63,16%	7,90%	2,63%	2,63%	
	2/38	2/38	2/38	1/38	1/38	1/38	24/38	3/38	1/38	1/38	
CA		21,05%		5,26%	5,26%	2,63%	2,63%	57,89%	2,63%	2,63%	
		8/38		2/38	2/38	1/38	1/38	22/38	1/38	1/38	
CAT	2,78%	5,55%			5,55%	2,78%	2,78%	8,33%	69,44%	2,78%	
	1/36	2/36			2/36	1/36	1/36	3/36	25/36	1/36	
CT	5,26%	2,63%	2,63%	7,90%	7,90%	10,53%	5,26%	2,63%		55,26%	
	2/38	1/38	1/38	3/38	3/38	4/38	2/38	1/38		21/38	

Do autor (2021).

A.2.8 1/128 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 52,99%.

- O desvio padrão da acurácia nos 10 *folds* foi 2,26%.
- O tempo total necessário para treinamento foi 14,114 segundos.
- A acurácia no conjunto de validação foi 51,86%.
- O tempo necessário para classificação do conjunto de validação foi 0,015 segundos.
- Acurácia em cada fold:
 - *Fold* 1: 52,63%
 - *Fold* 2: 53,29%
 - *Fold* 3: 55,33%
 - *Fold* 4: 55,33%
 - *Fold* 5: 56,00%
 - *Fold* 6: 50,67%
 - *Fold* 7: 53,33%
 - *Fold* 8: 48,67%
 - *Fold* 9: 50,67%
 - *Fold* 10: 54,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta BC com 0,019s.
 - O tipo de falta com menor tempo operacional foi a falta AT com 0,019s.

A Tabela 16 apresenta a matriz de confusão gerada pelo método C treinado com sinais com 1/128 ciclo pós falta.

Tabela 16 – Matriz de confusão do método C com 1/128 ciclo pós falta.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	47,37%	10,53%	5,26%	2,63%	2,63%	10,53%		7,89%	10,53%	2,63%
		18/38	4/38	2/38	1/38	1/38	4/38		3/38	4/38	1/38
	ABC	2,78%	55,56%	8,33%	2,78%	5,56%		5,56%	8,33%	5,56%	5,56%
		1/36	20/36	3/36	1/36	2/36		2/36	3/36	2/36	2/36
	ABT	7,89%	2,63%	50,0%	15,79%	7,89%		10,53%	2,63%	2,63%	
		3/38	1/38	19/38	6/38	3/38		4/38	1/38	1/38	
	AT	21,05%		7,89%	52,63%	2,63%	2,63%	2,63%	5,26%		5,26%
		8/38		3/38	20/38	1/38	1/38	1/38	2/38		2/38
	BC	5,26%	7,89%	2,63%	10,53%	47,37%		5,26%	7,89%	7,89%	5,26%
		2/38	3/38	1/38	4/38	18/38		2/38	3/38	3/38	2/38
	BCT	10,53%	7,89%	2,63%		7,89%	55,26%	2,63%	2,63%	10,53%	
		4/38	3/38	1/38		3/38	21/38	1/38	1/38	4/38	
	BT	5,26%	2,63%	5,26%	7,89%	2,63%	7,89%	52,63%	7,89%	5,26%	2,63%
		2/38	1/38	2/38	3/38	1/38	3/38	20/38	3/38	2/38	1/38
	CA	10,53%	7,89%	2,63%	2,63%	5,26%	7,89%	5,26%	50,0%	5,26%	2,63%
		4/38	3/38	1/38	1/38	2/38	3/38	2/38	19/38	2/38	1/38
CAT	11,11%	8,33%	2,78%	2,78%	5,56%		5,56%	5,56%	58,33%		
	4/36	3/36	1/36	1/36	2/36		2/36	2/36	21/36		
CT	13,16%	5,26%	2,63%	2,63%		10,53%	2,63%	10,53%	2,63%	50,0%	
	5/38	2/38	1/38	1/38		4/38	1/38	4/38	1/38	19/38	

Do autor (2021).

A.3 Método D

A.3.1 1 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 99,60%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,61%.
- O tempo total necessário para treinamento foi 23,826 segundos.
- A acurácia no conjunto de validação foi 99,47%.
- O tempo necessário para classificação do conjunto de validação foi 0,045 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 100,00%
 - *Fold* 3: 100,00%

- *Fold 4*: 100,00%
 - *Fold 5*: 98,67%
 - *Fold 6*: 98,67%
 - *Fold 7*: 100,00%
 - *Fold 8*: 100,00%
 - *Fold 9*: 98,67%
 - *Fold 10*: 100,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CAT com 0,070s.
 - O tipo de falta com menor tempo operacional foi a falta AT com 0,069s.

A Tabela 17 apresenta a matriz de confusão gerada pelo método D treinado com sinais com 1 ciclo pós falta.

Tabela 17 – Matriz de confusão do método D com 1 ciclo pós falta.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0%											
		19/19											
	ABC		100,0%										
			18/18										
	ABT			100,0%									
				19/19									
	AT				100,0%								
					19/19								
	BC					100,0%							
						19/19							
BCT						100,0%							
						19/19							
BT							100,0%						
							19/19						
CA								100,0%					
								19/19					
CAT									5,56%	94,44%			
									1/18	17/18			
CT											100,0%		
											19/19		

Do autor (2021).

A.3.2 1/2 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 99,47%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,65%.
- O tempo total necessário para treinamento foi 4,856 segundos.
- A acurácia no conjunto de validação foi 100,00%.
- O tempo necessário para classificação do conjunto de validação foi 0,032 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 100,00%
 - *Fold* 3: 100,00%
 - *Fold* 4: 100,00%
 - *Fold* 5: 98,67%
 - *Fold* 6: 98,67%
 - *Fold* 7: 100,00%
 - *Fold* 8: 100,00%
 - *Fold* 9: 98,67%
 - *Fold* 10: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CT com 0,046s.
 - O tipo de falta com menor tempo operacional foi a falta ABC com 0,045s.

A Tabela 18 apresenta a matriz de confusão gerada pelo método D treinado com sinais com 1/2 ciclo pós falta.

Tabela 18 – Matriz de confusão do método D com 1/2 ciclo pós falta.

		Valores preditos										
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT	
Valores reais	AB	100,0% 19/19										
	ABC		100,0% 18/18									
	ABT			100,0% 19/19								
	AT				100,0% 19/19							
	BC					100,0% 19/19						
	BCT						100,0% 19/19					
	BT							100,0% 19/19				
	CA								100,0% 19/19			
	CAT									100,0% 18/18		
	CT										100,0% 19/19	

Do autor (2021).

A.3.3 1/4 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 98,93%.
- O desvio padrão da acurácia nos 10 *fold*s foi 0,80%.
- O tempo total necessário para treinamento foi 4,838 segundos.
- A acurácia no conjunto de validação foi 97,87%.
- O tempo necessário para classificação do conjunto de validação foi 0,018 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 100,00%
 - *Fold* 3: 98,67%
 - *Fold* 4: 98,67%
 - *Fold* 5: 97,33%
 - *Fold* 6: 98,67%

- *Fold 7*: 98,67%
 - *Fold 8*: 100,00%
 - *Fold 9*: 98,67%
 - *Fold 10*: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CA com 0,030s.
 - O tipo de falta com menor tempo operacional foi a falta ABC com 0,030s.

A Tabela 19 apresenta a matriz de confusão gerada pelo método D treinado com sinais com 1/4 ciclo pós falta.

Tabela 19 – Matriz de confusão do método D com 1/4 ciclo pós falta.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0%											
		19/19											
	ABC		100,0%										
			18/18										
	ABT			84,21%	15,79%								
				16/19	3/19								
	AT				100,0%								
					19/19								
	BC					100,0%							
						19/19							
	BCT						100,0%						
						19/19							
BT							100,0%						
							19/19						
CA								100,0%					
								19/19					
CAT				5,56%					94,44%				
				1/18					17/18				
CT											100,0%		
											19/19		

Do autor (2021).

A.3.4 1/8 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 99,34%.

- O desvio padrão da acurácia nos 10 *folds* foi 0,66%.
- O tempo total necessário para treinamento foi 3,194 segundos.
- A acurácia no conjunto de validação foi 97,87%.
- O tempo necessário para classificação do conjunto de validação foi 0,019 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 98,68%
 - *Fold* 3: 100,00%
 - *Fold* 4: 100,00%
 - *Fold* 5: 100,00%
 - *Fold* 6: 98,67%
 - *Fold* 7: 98,67%
 - *Fold* 8: 98,67%
 - *Fold* 9: 100,00%
 - *Fold* 10: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta AB com 0,026s.
 - O tipo de falta com menor tempo operacional foi a falta BT com 0,025s.

A Tabela 20 apresenta a matriz de confusão gerada pelo método D treinado com sinais com 1/8 ciclo pós falta.

Tabela 20 – Matriz de confusão do método D com 1/8 ciclo pós falta.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0%											
		19/19											
	ABC		100,0%										
			18/18										
	ABT			84,21%	15,79%								
				16/19	3/19								
	AT				100,0%								
					19/19								
	BC					100,0%							
						19/19							
	BCT						94,74%					5,26%	
							18/19					1/19	
	BT							100,0%					
							19/19						
CA								100,0%					
								19/19					
CAT									100,0%				
									18/18				
CT											100,0%		
											19/19		

Do autor (2021).

A.3.5 1/16 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 97,74%.
- O desvio padrão da acurácia nos 10 *fold*s foi 1,34%.
- O tempo total necessário para treinamento foi 2,744 segundos.
- A acurácia no conjunto de validação foi 95,74%.
- O tempo necessário para classificação do conjunto de validação foi 0,012 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 98,68%
 - *Fold* 2: 100,00%
 - *Fold* 3: 98,67%
 - *Fold* 4: 96,00%
 - *Fold* 5: 97,33%

- *Fold 6*: 96,00%
 - *Fold 7*: 97,33%
 - *Fold 8*: 96,00%
 - *Fold 9*: 98,67%
 - *Fold 10*: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CAT com 0,020s.
 - O tipo de falta com menor tempo operacional foi a falta ABT com 0,020s.

A Tabela 21 apresenta a matriz de confusão gerada pelo método D treinado com sinais com 1/16 ciclo pós falta.

Tabela 21 – Matriz de confusão do método D com 1/16 ciclo pós falta.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0%											
		19/19											
	ABC		100,0%										
			18/18										
	ABT			73,68%	26,32%								
				14/19	5/19								
	AT				100,0%								
					19/19								
	BC					100,0%							
						19/19							
BCT						89,47%	10,53%						
						17/19	2/19						
BT							100,0%						
							19/19						
CA								100,0%					
								19/19					
CAT									94,44%	5,56%			
									17/18	1/18			
CT											100,0%		
											19/19		

Do autor (2021).

A.3.6 1/32 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 96,94%.

- O desvio padrão da acurácia nos 10 *folds* foi 1,47%.
- O tempo total necessário para treinamento foi 3,664 segundos.
- A acurácia no conjunto de validação foi 95,21%.
- O tempo necessário para classificação do conjunto de validação foi 0,021 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 97,37%
 - *Fold* 2: 98,68%
 - *Fold* 3: 96,00%
 - *Fold* 4: 94,67%
 - *Fold* 5: 98,67%
 - *Fold* 6: 97,33%
 - *Fold* 7: 94,67%
 - *Fold* 8: 96,00%
 - *Fold* 9: 98,67%
 - *Fold* 10: 97,33%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CT com 0,022s.
 - O tipo de falta com menor tempo operacional foi a falta CAT com 0,022s.

A Tabela 22 apresenta a matriz de confusão gerada pelo método D treinado com sinais com 1/32 ciclo pós falta.

Tabela 22 – Matriz de confusão do método D com 1/32 ciclo pós falta.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0% 19/19											
	ABC		100,0% 18/18										
	ABT			68,42% 13/19	31,58% 6/19								
	AT				100,0% 19/19								
	BC					100,0% 19/19							
	BCT						89,47% 17/19	10,53% 2/19					
	BT							100,0% 19/19					
	CA								100,0% 19/19				
	CAT									94,44% 17/18	5,56% 1/18		
	CT											100,0% 19/19	

Do autor (2021).

A.3.7 1/64 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 94,68%.
- O desvio padrão da acurácia nos 10 *fold*s foi 1,87%.
- O tempo total necessário para treinamento foi 3,909 segundos.
- A acurácia no conjunto de validação foi 93,62%.
- O tempo necessário para classificação do conjunto de validação foi 0,015 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 93,42%
 - *Fold* 2: 93,42%
 - *Fold* 3: 92,00%
 - *Fold* 4: 94,67%
 - *Fold* 5: 94,67%

- *Fold 6*: 92,00%
 - *Fold 7*: 96,00%
 - *Fold 8*: 96,00%
 - *Fold 9*: 97,33%
 - *Fold 10*: 97,33%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta BCT com 0,019s.
 - O tipo de falta com menor tempo operacional foi a falta AT com 0,019s.

A Tabela 23 apresenta a matriz de confusão gerada pelo método D treinado com sinais com 1/64 ciclo pós falta.

Tabela 23 – Matriz de confusão do método D com 1/64 ciclo pós falta.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	94,74%	5,26%										
		18/19	1/19										
	ABC		100,0%										
			18/18										
	ABT			73,68%	26,32%								
				14/19	5/19								
	AT				100,0%								
					19/19								
	BC		5,26%			94,74%							
			1/19			18/19							
BCT						84,21%	15,79%						
						16/19	3/19						
BT							100,0%						
							19/19						
CA								100,0%					
								19/19					
CAT									94,44%	5,56%			
									17/18	1/18			
CT											94,74%		
											18/19		

Do autor (2021).

A.3.8 1/128 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 93,75%.

- O desvio padrão da acurácia nos 10 *folds* foi 1,46%.
- O tempo total necessário para treinamento foi 7,299 segundos.
- A acurácia no conjunto de validação foi 90,96%.
- O tempo necessário para classificação do conjunto de validação foi 0,032 segundos.
- Acurácia em cada *fold* :
 - *Fold* 1: 92,11%
 - *Fold* 2: 94,74%
 - *Fold* 3: 92,00%
 - *Fold* 4: 93,33%
 - *Fold* 5: 92,00%
 - *Fold* 6: 93,33%
 - *Fold* 7: 94,67%
 - *Fold* 8: 96,00%
 - *Fold* 9: 93,33%
 - *Fold* 10: 96,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta AB com 0,027s.
 - O tipo de falta com menor tempo operacional foi a falta BCT com 0,027s.

A Tabela 24 apresenta a matriz de confusão gerada pelo método D treinado com sinais com 1/128 ciclo pós falta.

Tabela 24 – Matriz de confusão do método D com 1/128 ciclo pós falta.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0%											
		19/19											
	ABC		100,0%										
			18/18										
	ABT			68,42%	26,32%			5,26%					
				13/19	5/19			1/19					
	AT				100,0%								
					19/19								
	BC					100,0%							
						19/19							
	BCT					5,26%	68,42%	21,05%				5,26%	
						1/19	13/19	4/19				1/19	
	BT						5,26%	94,74%					
						1/19	18/19						
CA								100,0%					
								19/19					
CAT				5,56%				5,56%	88,89%				
				1/18				1/18	16/18				
CT						5,26%			5,26%	89,47%			
						1/19			1/19	17/19			

Do autor (2021).

A.4 Método E

A.4.1 1 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 99,60%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,44%.
- O tempo total necessário para treinamento foi 66,643 segundos.
- A acurácia no conjunto de validação foi 99,73%.
- O tempo necessário para classificação do conjunto de validação foi 0,055 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 99,34%
 - *Fold* 3: 100,00%

- *Fold 4*: 100,00%
 - *Fold 5*: 99,33%
 - *Fold 6*: 100,00%
 - *Fold 7*: 99,33%
 - *Fold 8*: 100,00%
 - *Fold 9*: 98,67%
 - *Fold 10*: 99,33%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CA com 0,072s.
 - O tipo de falta com menor tempo operacional foi a falta ABC com 0,071s.

A Tabela 25 apresenta a matriz de confusão gerada pelo método E treinado com sinais com 1 ciclo pós falta.

Tabela 25 – Matriz de confusão do método E com 1 ciclo pós falta.

		Valores preditos										
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT	
Valores reais	AB	100,0%										
		38/38										
	ABC		100,0%									
			36/36									
	ABT			100,0%								
				38/38								
	AT				97,37%			2,63%				
					37/38			1/38				
	BC					100,0%						
						38/38						
	BCT						100,0%					
						38/38						
BT							100,0%					
							38/38					
CA								100,0%				
								38/38				
CAT									100,0%			
									36/36			
CT										100,0%		
										38/38		

Do autor (2021).

A.4.2 1/2 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 99,93%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,20%.
- O tempo total necessário para treinamento foi 17,998 segundos.
- A acurácia no conjunto de validação foi 99,73%.
- O tempo necessário para classificação do conjunto de validação foi 0,035 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 100,00%
 - *Fold* 3: 100,00%
 - *Fold* 4: 100,00%
 - *Fold* 5: 100,00%
 - *Fold* 6: 100,00%
 - *Fold* 7: 100,00%
 - *Fold* 8: 99,33%
 - *Fold* 9: 100,00%
 - *Fold* 10: 100,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta ABC com 0,047s.
 - O tipo de falta com menor tempo operacional foi a falta CAT com 0,046s.

A Tabela 26 apresenta a matriz de confusão gerada pelo método E treinado com sinais com 1/2 ciclo pós falta.

Tabela 26 – Matriz de confusão do método E com 1/2 ciclo pós falta.

		Valores preditos										
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT	
Valores reais	AB	100,0% 38/38										
	ABC		100,0% 36/36									
	ABT			100,0% 38/38								
	AT				97,37% 37/38						2,63% 1/38	
	BC					100,0% 38/38						
	BCT						100,0% 38/38					
	BT							100,0% 38/38				
	CA								100,0% 38/38			
	CAT									100,0% 36/36		
	CT										100,0% 38/38	

Do autor (2021).

A.4.3 1/4 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 99,53%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,52%.
- O tempo total necessário para treinamento foi 12,179 segundos.
- A acurácia no conjunto de validação foi 99,73%.
- O tempo necessário para classificação do conjunto de validação foi 0,039 segundos.
- Acurácia em cada *fold* :
 - *Fold* 1: 100,00%
 - *Fold* 2: 100,00%
 - *Fold* 3: 100,00%
 - *Fold* 4: 99,33%
 - *Fold* 5: 98,67%
 - *Fold* 6: 100,00%

- *Fold 7*: 100,00%
 - *Fold 8*: 99,33%
 - *Fold 9*: 98,67%
 - *Fold 10*: 99,33%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta BT com 0,037s.
 - O tipo de falta com menor tempo operacional foi a falta AB com 0,037s.

A Tabela 27 apresenta a matriz de confusão gerada pelo método E treinado com sinais com 1/4 ciclo pós falta.

Tabela 27 – Matriz de confusão do método E com 1/4 ciclo pós falta.

		Valores preditos										
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT	
Valores reais	AB	100,0% 38/38										
	ABC		100,0% 36/36									
	ABT			100,0% 38/38								
	AT				100,0% 38/38							
	BC					100,0% 38/38						
	BCT						100,0% 38/38					
	BT			2,63% 1/38				97,37% 37/38				
	CA								100,0% 38/38			
	CAT									100,0% 36/36		
	CT										100,0% 38/38	

Do autor (2021).

A.4.4 1/8 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 99,20%.

- O desvio padrão da acurácia nos 10 *fold*s foi 0,78%.
- O tempo total necessário para treinamento foi 27,061 segundos.
- A acurácia no conjunto de validação foi 98,67%.
- O tempo necessário para classificação do conjunto de validação foi 0,033 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 99,34%
 - *Fold* 3: 100,00%
 - *Fold* 4: 99,33%
 - *Fold* 5: 98,67%
 - *Fold* 6: 100,00%
 - *Fold* 7: 100,00%
 - *Fold* 8: 98,00%
 - *Fold* 9: 98,67%
 - *Fold* 10: 98,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta BT com 0,029s.
 - O tipo de falta com menor tempo operacional foi a falta AB com 0,029s.

A Tabela 28 apresenta a matriz de confusão gerada pelo método E treinado com sinais com 1/8 ciclo pós falta.

Tabela 28 – Matriz de confusão do método E com 1/8 ciclo pós falta.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0% 38/38											
	ABC		100,0% 36/36										
	ABT			100,0% 38/38									
	AT				100,0% 38/38								
	BC					100,0% 38/38							
	BCT						100,0% 38/38						
	BT			2,63% 1/38				97,37% 37/38					
	CA								100,0% 38/38				
	CAT				2,78% 1/36					94,44% 34/36	2,78% 1/36		
	CT									5,26% 2/38	94,74% 36/38		

Do autor (2021).

A.4.5 1/16 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 94,14%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,77%.
- O tempo total necessário para treinamento foi 18,78 segundos.
- A acurácia no conjunto de validação foi 92,82%.
- O tempo necessário para classificação do conjunto de validação foi 0,025 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 96,05%
 - *Fold* 2: 96,71%
 - *Fold* 3: 94,67%
 - *Fold* 4: 95,33%
 - *Fold* 5: 93,33%

- *Fold 6*: 92,00%
 - *Fold 7*: 92,67%
 - *Fold 8*: 91,33%
 - *Fold 9*: 96,00%
 - *Fold 10*: 93,33%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta ABT com 0,024s.
 - O tipo de falta com menor tempo operacional foi a falta BCT com 0,024s.

A Tabela 29 apresenta a matriz de confusão gerada pelo método E treinado com sinais com 1/16 ciclo pós falta.

Tabela 29 – Matriz de confusão do método E com 1/16 ciclo pós falta.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0%											
		38/38											
	ABC		97,22%			2,78%							
			35/36			1/36							
	ABT			76,32%	18,42%			5,26%					
				29/38	7/38			2/38					
	AT				2,63%	97,37%							
					1/38	37/38							
	BC		2,63%			97,37%							
			1/38			37/38							
BCT					2,63%	86,84%	5,26%				5,26%		
					1/38	33/38	2/38				2/38		
BT			2,63%				97,37%						
			1/38				37/38						
CA		5,26%						94,74%					
		2/38						36/38					
CAT				5,56%					88,89%	5,56%			
				2/36					32/36	2/36			
CT						2,63%			5,26%	92,11%			
						1/38			2/38	35/38			

Do autor (2021).

A.4.6 1/32 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 68,28%.

- O desvio padrão da acurácia nos 10 *folds* foi 2,23%.
- O tempo total necessário para treinamento foi 55,042 segundos.
- A acurácia no conjunto de validação foi 67,82%.
- O tempo necessário para classificação do conjunto de validação foi 0,043 segundos.
- Acurácia em cada *fold* :
 - *Fold* 1: 69,08%
 - *Fold* 2: 71,05%
 - *Fold* 3: 66,00%
 - *Fold* 4: 64,67%
 - *Fold* 5: 68,00%
 - *Fold* 6: 67,33%
 - *Fold* 7: 72,00%
 - *Fold* 8: 66,00%
 - *Fold* 9: 68,67%
 - *Fold* 10: 70,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CT com 0,030s.
 - O tipo de falta com menor tempo operacional foi a falta AB com 0,030s.

A Tabela 30 apresenta a matriz de confusão gerada pelo método E treinado com sinais com 1/32 ciclo pós falta.

Tabela 30 – Matriz de confusão do método E com 1/32 ciclo pós falta.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	71,05% 27/38	2,63% 1/38	10,53% 4/38		2,63% 1/38	2,63% 1/38	5,26% 2/38	2,63% 1/38		2,63% 1/38
	ABC	11,11% 4/36	72,22% 26/36			8,33% 3/36	2,78% 1/36	5,56% 2/36			
	ABT	21,05% 8/38	2,63% 1/38	52,63% 20/38	15,79% 6/38			2,63% 1/38			5,26% 2/38
	AT	2,63% 1/38	2,63% 1/38		71,05% 27/38	2,63% 1/38	5,26% 2/38	7,89% 3/38	2,63% 1/38	2,63% 1/38	2,63% 1/38
	BC	5,26% 2/38	2,63% 1/38	2,63% 1/38		81,58% 31/38	5,26% 2/38	2,63% 1/38			
	BCT	2,63% 1/38				15,79% 6/38	63,16% 24/38	7,89% 3/38	7,89% 3/38		2,63% 1/38
	BT	5,26% 2/38		5,26% 2/38		7,89% 3/38	7,89% 3/38	73,68% 28/38			
	CA		5,26% 2/38		5,26% 2/38	5,26% 2/38	5,26% 2/38	2,63% 1/38	65,79% 25/38	10,53% 4/38	
	CAT	8,33% 3/36			11,11% 4/36	5,56% 2/36	2,78% 1/36	5,56% 2/36	2,78% 1/36	61,11% 22/36	2,78% 1/36
	CT	2,63% 1/38		2,63% 1/38	2,63% 1/38	5,26% 2/38	10,53% 4/38	2,63% 1/38	5,26% 2/38	2,63% 1/38	65,79% 25/38

Do autor (2021).

A.4.7 1/64 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 57,31%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,98%.
- O tempo total necessário para treinamento foi 32,164 segundos.
- A acurácia no conjunto de validação foi 57,98%.
- O tempo necessário para classificação do conjunto de validação foi 0,026 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 56,58%
 - *Fold* 2: 57,24%
 - *Fold* 3: 55,33%
 - *Fold* 4: 57,33%
 - *Fold* 5: 58,00%

- Fold 6: 54,67%
- Fold 7: 58,67%
- Fold 8: 60,00%
- Fold 9: 54,67%
- Fold 10: 60,67%

- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:

- O tipo de falta com maior tempo operacional foi a falta ABC com 0,025s.
- O tipo de falta com menor tempo operacional foi a falta AB com 0,024s.

A Tabela 31 apresenta a matriz de confusão gerada pelo método E treinado com sinais com 1/64 ciclo pós falta.

Tabela 31 – Matriz de confusão do método E com 1/64 ciclo pós falta.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	55,26% 21/38	7,89% 3/38	5,26% 2/38	2,63% 1/38		5,26% 2/38	2,63% 1/38	15,79% 6/38		5,26% 2/38
	ABC	8,33% 3/36	69,44% 25/36			2,78% 1/36	5,56% 2/36	2,78% 1/36	11,11% 4/36		
	ABT		13,16% 5/38	42,11% 16/38	15,79% 6/38	2,63% 1/38	2,63% 1/38	2,63% 1/38	15,79% 6/38	2,63% 1/38	2,63% 1/38
	AT	10,53% 4/38	2,63% 1/38	2,63% 1/38	60,53% 23/38	5,26% 2/38		2,63% 1/38	7,89% 3/38	7,89% 3/38	
	BC	13,16% 5/38	10,53% 4/38	7,89% 3/38		55,26% 21/38	2,63% 1/38	2,63% 1/38	2,63% 1/38	2,63% 1/38	2,63% 1/38
	BCT	5,26% 2/38		5,26% 2/38	2,63% 1/38	7,89% 3/38	57,89% 22/38	5,26% 2/38	10,53% 4/38	2,63% 1/38	2,63% 1/38
	BT	7,89% 3/38	10,53% 4/38	5,26% 2/38	2,63% 1/38	2,63% 1/38	7,89% 3/38	63,16% 24/38			
	CA	7,89% 3/38		5,26% 2/38			5,26% 2/38	7,89% 3/38	63,16% 24/38	7,89% 3/38	2,63% 1/38
	CAT	11,11% 4/36	5,56% 2/36		8,33% 3/36	5,56% 2/36	5,56% 2/36	2,78% 1/36		55,56% 20/36	5,56% 2/36
	CT	10,53% 4/38		7,89% 3/38	2,63% 1/38	2,63% 1/38	7,89% 3/38		2,63% 1/38	7,89% 3/38	57,89% 22/38

Do autor (2021).

A.4.8 1/128 Ciclo Pós Falta

- A acurácia média da validação cruzada foi 51,19%.

- O desvio padrão da acurácia nos 10 *fold*s foi 1,80%.
- O tempo total necessário para treinamento foi 6,622 segundos.
- A acurácia no conjunto de validação foi 50,80%.
- O tempo necessário para classificação do conjunto de validação foi 0,01 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 51,97%
 - *Fold* 2: 52,63%
 - *Fold* 3: 51,33%
 - *Fold* 4: 52,67%
 - *Fold* 5: 48,00%
 - *Fold* 6: 48,00%
 - *Fold* 7: 50,67%
 - *Fold* 8: 52,67%
 - *Fold* 9: 53,33%
 - *Fold* 10: 50,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta AB com 0,018s.
 - O tipo de falta com menor tempo operacional foi a falta BC com 0,017s.

A Tabela 32 apresenta a matriz de confusão gerada pelo método E treinado com sinais com 1/128 ciclo pós falta.

Tabela 32 – Matriz de confusão do método E com 1/128 ciclo pós falta.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	52,63% 20/38		7,89% 3/38	2,63% 1/38	2,63% 1/38	5,26% 2/38	10,53% 4/38	10,53% 4/38		7,89% 3/38
	ABC	2,78% 1/36	61,11% 22/36	2,78% 1/36	11,11% 4/36		2,78% 1/36	2,78% 1/36	5,56% 2/36	8,33% 3/36	2,78% 1/36
	ABT	2,63% 1/38	2,63% 1/38	44,74% 17/38	21,05% 8/38	7,89% 3/38	2,63% 1/38	2,63% 1/38	5,26% 2/38	5,26% 2/38	5,26% 2/38
	AT	5,26% 2/38	2,63% 1/38		50,0% 19/38	2,63% 1/38	5,26% 2/38	7,89% 3/38	7,89% 3/38	10,53% 4/38	7,89% 3/38
	BC	2,63% 1/38	5,26% 2/38	7,89% 3/38	7,89% 3/38	52,63% 20/38	5,26% 2/38	7,89% 3/38	2,63% 1/38	5,26% 2/38	2,63% 1/38
	BCT	7,89% 3/38	5,26% 2/38	7,89% 3/38	2,63% 1/38	7,89% 3/38	39,47% 15/38	7,89% 3/38	5,26% 2/38	2,63% 1/38	13,16% 5/38
	BT	2,63% 1/38	5,26% 2/38	5,26% 2/38	7,89% 3/38	10,53% 4/38		52,63% 20/38		10,53% 4/38	5,26% 2/38
	CA	5,26% 2/38		5,26% 2/38	5,26% 2/38	5,26% 2/38	7,89% 3/38		60,53% 23/38	5,26% 2/38	5,26% 2/38
	CAT	11,11% 4/36	5,56% 2/36			13,89% 5/36		8,33% 3/36	8,33% 3/36	47,22% 17/36	5,56% 2/36
	CT	10,53% 4/38			10,53% 4/38	2,63% 1/38	5,26% 2/38	5,26% 2/38	13,16% 5/38	5,26% 2/38	47,37% 18/38

Do autor (2021).

A.5 Método F

A.5.1 Rocket

A.5.1.1 1 Ciclo Pós Falta

Treinamento realizado com 300 kernels

- A acurácia média da validação cruzada foi 100,00%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,00%.
- O tempo total necessário para treinamento foi 31,621 segundos.
- A acurácia no conjunto de validação foi 100,00%.
- O tempo necessário para classificação do conjunto de validação foi 0,04 segundos.
- Acurácia em cada *fold*:

– *Fold* 1: 100,00%

- *Fold 2*: 100,00%
 - *Fold 3*: 100,00%
 - *Fold 4*: 100,00%
 - *Fold 5*: 100,00%
 - *Fold 6*: 100,00%
 - *Fold 7*: 100,00%
 - *Fold 8*: 100,00%
 - *Fold 9*: 100,00%
 - *Fold 10*: 100,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta AB com 0,015s.
 - O tipo de falta com menor tempo operacional foi a falta BC com 0,014s.

A Tabela 33 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 33 – Matriz de confusão do método F com 1 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos										
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT	
Valores reais	AB	100,0%										
		19/19										
	ABC		100,0%									
			18/18									
	ABT			100,0%								
				19/19								
	AT				100,0%							
					19/19							
	BC					100,0%						
						19/19						
	BCT						100,0%					
							19/19					
BT							100,0%					
							19/19					
CA								100,0%				
								19/19				
CAT									100,0%			
									18/18			
CT										100,0%		
										19/19		

Do autor (2021).

A.5.1.2 1/2 Ciclo Pós Falta

Treinamento realizado com 400 *kernels*

- A acurácia média da validação cruzada foi 99,87%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,40%.
- O tempo total necessário para treinamento foi 28,459 segundos.
- A acurácia no conjunto de validação foi 100,00%.
- O tempo necessário para classificação do conjunto de validação foi 0,048 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 100,00%
 - *Fold* 3: 100,00%
 - *Fold* 4: 100,00%

- *Fold 5*: 100,00%
- *Fold 6*: 100,00%
- *Fold 7*: 100,00%
- *Fold 8*: 100,00%
- *Fold 9*: 98,67%
- *Fold 10*: 100,00%

- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CAT com 0,014s.
 - O tipo de falta com menor tempo operacional foi a falta AT com 0,014s.

A Tabela 34 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1/2 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 34 – Matriz de confusão do método F com 1/2 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos										
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT	
Valores reais	AB	100,0%										
		19/19										
	ABC		100,0%									
			18/18									
	ABT			100,0%								
				19/19								
	AT				100,0%							
					19/19							
	BC					100,0%						
						19/19						
BCT						100,0%						
						19/19						
BT							100,0%					
							19/19					
CA								100,0%				
								19/19				
CAT									100,0%			
									18/18			
CT											100,0%	
											19/19	

Do autor (2021).

A.5.1.3 1/4 Ciclo Pós Falta

Treinamento realizado com 400 *kernels*

- A acurácia média da validação cruzada foi 99,73%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,53%.
- O tempo total necessário para treinamento foi 21,47 segundos.
- A acurácia no conjunto de validação foi 100,00%.
- O tempo necessário para classificação do conjunto de validação foi 0,041 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 100,00%
 - *Fold* 3: 100,00%
 - *Fold* 4: 100,00%
 - *Fold* 5: 100,00%
 - *Fold* 6: 100,00%
 - *Fold* 7: 98,67%
 - *Fold* 8: 100,00%
 - *Fold* 9: 98,67%
 - *Fold* 10: 100,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta AB com 0,015s.
 - O tipo de falta com menor tempo operacional foi a falta BT com 0,013s.

A Tabela 35 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1/4 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 35 – Matriz de confusão do método F com 1/4 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos										
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT	
Valores reais	AB	100,0% 19/19										
	ABC		100,0% 18/18									
	ABT			100,0% 19/19								
	AT				100,0% 19/19							
	BC					100,0% 19/19						
	BCT						100,0% 19/19					
	BT							100,0% 19/19				
	CA								100,0% 19/19			
	CAT									100,0% 18/18		
	CT										100,0% 19/19	

Do autor (2021).

A.5.1.4 1/8 Ciclo Pós Falta

Treinamento realizado com 500 *kernels*

- A acurácia média da validação cruzada foi 99,07%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,04%.
- O tempo total necessário para treinamento foi 20,451 segundos.
- A acurácia no conjunto de validação foi 100,00%.
- O tempo necessário para classificação do conjunto de validação foi 0,052 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 100,00%
 - *Fold* 3: 100,00%

- *Fold 4*: 98,67%
 - *Fold 5*: 97,33%
 - *Fold 6*: 100,00%
 - *Fold 7*: 100,00%
 - *Fold 8*: 98,67%
 - *Fold 9*: 97,33%
 - *Fold 10*: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta ABC com 0,030s.
 - O tipo de falta com menor tempo operacional foi a falta CAT com 0,014s.

A Tabela 36 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1/8 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 36 – Matriz de confusão do método F com 1/8 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos										
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT	
Valores reais	AB	100,0%										
		19/19										
	ABC		100,0%									
			18/18									
	ABT			100,0%								
				19/19								
	AT				100,0%							
					19/19							
	BC					100,0%						
						19/19						
	BCT						100,0%					
						19/19						
BT							100,0%					
							19/19					
CA								100,0%				
								19/19				
CAT									100,0%			
									18/18			
CT										100,0%		
										19/19		

Do autor (2021).

A.5.1.5 1/16 Ciclo Pós Falta

Treinamento realizado com 900 *kernels*

- A acurácia média da validação cruzada foi 97,87%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,60%.
- O tempo total necessário para treinamento foi 27,911 segundos.
- A acurácia no conjunto de validação foi 98,40%.
- O tempo necessário para classificação do conjunto de validação foi 0,076 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 98,68%
 - *Fold* 3: 97,33%
 - *Fold* 4: 97,33%
 - *Fold* 5: 96,00%
 - *Fold* 6: 98,67%
 - *Fold* 7: 100,00%
 - *Fold* 8: 94,67%
 - *Fold* 9: 98,67%
 - *Fold* 10: 97,33%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta BC com 0,019s.
 - O tipo de falta com menor tempo operacional foi a falta CA com 0,018s.

A Tabela 37 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1/16 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 37 – Matriz de confusão do método F com 1/16 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0% 19/19											
	ABC		100,0% 18/18										
	ABT			94,74% 18/19	5,26% 1/19								
	AT				100,0% 19/19								
	BC					100,0% 19/19							
	BCT						100,0% 19/19						
	BT							100,0% 19/19					
	CA								94,74% 18/19	5,26% 1/19			
	CAT									5,56% 1/18	94,44% 17/18		
	CT											100,0% 19/19	

Do autor (2021).

A.5.1.6 1/32 Ciclo Pós Falta

Treinamento realizado com 1000 *kernels*

- A acurácia média da validação cruzada foi 95,21%.
- O desvio padrão da acurácia nos 10 *folds* foi 3,28%.
- O tempo total necessário para treinamento foi 27,177 segundos.
- A acurácia no conjunto de validação foi 98,40%.
- O tempo necessário para classificação do conjunto de validação foi 0,106 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 97,37%
 - *Fold* 2: 97,37%
 - *Fold* 3: 93,33%

- *Fold 4*: 93,33%
 - *Fold 5*: 88,00%
 - *Fold 6*: 97,33%
 - *Fold 7*: 98,67%
 - *Fold 8*: 92,00%
 - *Fold 9*: 98,67%
 - *Fold 10*: 96,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta ABT com 0,021s.
 - O tipo de falta com menor tempo operacional foi a falta BT com 0,019s.

A Tabela 38 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1/32 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 38 – Matriz de confusão do método F com 1/32 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	100,0% 19/19									
	ABC		100,0% 18/18								
	ABT			94,74% 18/19	5,26% 1/19						
	AT				100,0% 19/19						
	BC					100,0% 19/19					
	BCT						100,0% 19/19				
	BT							100,0% 19/19			
	CA								89,47% 17/19	10,53% 2/19	
	CAT									100,0% 18/18	
	CT										100,0% 19/19

Do autor (2021).

A.5.1.7 1/64 Ciclo Pós Falta

Treinamento realizado com 900 *kernels*

- A acurácia média da validação cruzada foi 94,01%.
- O desvio padrão da acurácia nos 10 *folds* foi 2,75%.
- O tempo total necessário para treinamento foi 25,0 segundos.
- A acurácia no conjunto de validação foi 96,81%.
- O tempo necessário para classificação do conjunto de validação foi 0,077 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 97,37%
 - *Fold* 2: 93,42%
 - *Fold* 3: 92,00%
 - *Fold* 4: 92,00%
 - *Fold* 5: 88,00%
 - *Fold* 6: 96,00%
 - *Fold* 7: 97,33%
 - *Fold* 8: 94,67%
 - *Fold* 9: 93,33%
 - *Fold* 10: 96,00%

 - O tipo de falta com maior tempo operacional foi a falta CT com 0,026s.
 - O tipo de falta com menor tempo operacional foi a falta AB com 0,021s.

A Tabela 39 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1/64 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 39 – Matriz de confusão do método F com 1/64 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos										
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT	
Valores reais	AB	100,0% 19/19										
	ABC		100,0% 18/18									
	ABT	5,26% 1/19		84,21% 16/19	10,53% 2/19							
	AT				100,0% 19/19							
	BC					100,0% 19/19						
	BCT					5,26% 1/19	94,74% 18/19					
	BT							100,0% 19/19				
	CA								94,74% 18/19	5,26% 1/19		
	CAT								5,56% 1/18	94,44% 17/18		
	CT										100,0% 19/19	

Do autor (2021).

A.5.1.8 1/128 Ciclo Pós Falta

Treinamento realizado com 600 *kernels*

- A acurácia média da validação cruzada foi 93,48%.
- O desvio padrão da acurácia nos 10 *folds* foi 3,00%.
- O tempo total necessário para treinamento foi 18,781 segundos.
- A acurácia no conjunto de validação foi 94,68%.
- O tempo necessário para classificação do conjunto de validação foi 0,067 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 96,05%
 - *Fold* 2: 90,79%
 - *Fold* 3: 93,33%

- *Fold 4*: 90,67%
 - *Fold 5*: 86,67%
 - *Fold 6*: 94,67%
 - *Fold 7*: 96,00%
 - *Fold 8*: 94,67%
 - *Fold 9*: 96,00%
 - *Fold 10*: 96,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CT com 0,023s.
 - O tipo de falta com menor tempo operacional foi a falta ABT com 0,022s.

A Tabela 40 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1/128 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 40 – Matriz de confusão do método F com 1/128 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	100,0% 19/19									
	ABC		100,0% 18/18								
	ABT			94,74% 18/19	5,26% 1/19						
	AT				94,74% 18/19				5,26% 1/19		
	BC					100,0% 19/19					
	BCT					5,26% 1/19	84,21% 16/19	5,26% 1/19			5,26% 1/19
	BT							100,0% 19/19			
	CA								78,95% 15/19	21,05% 4/19	
	CAT									94,44% 17/18	5,56% 1/18
	CT										100,0% 19/19

Do autor (2021).

A.5.2 MiniRocket

Para utilizar a técnica *MiniRocket* variou-se a quantidade *features* e foram registradas a acurácia de treinamento e validação, além dos tempos necessários para treinar e validar os modelos. No apêndice B foi incluída a Tabela 66 que apresenta os resultados obtidos nesses testes.

A.5.2.1 1 Ciclo Pós Falta

Treinamento realizado com 100 *features*

- A acurácia média da validação cruzada foi 100,00%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,00%.
- O tempo total necessário para treinamento foi 7,018 segundos.
- A acurácia no conjunto de validação foi 100,00%.
- O tempo necessário para classificação do conjunto de validação foi 0,011 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 100,00%
 - *Fold* 3: 100,00%
 - *Fold* 4: 100,00%
 - *Fold* 5: 100,00%
 - *Fold* 6: 100,00%
 - *Fold* 7: 100,00%
 - *Fold* 8: 100,00%
 - *Fold* 9: 100,00%
 - *Fold* 10: 100,00%
- O tipo de falta com maior tempo operacional foi a falta AB com 0,014s.
- O tipo de falta com menor tempo operacional foi a falta CT com 0,012s.

A Tabela 41 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 41 – Matriz de confusão do método F com 1 ciclo pós falta e extração de características com *MiniRocket*.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0%											
		19/19											
	ABC		100,0%										
			18/18										
	ABT			100,0%									
				19/19									
	AT				100,0%								
					19/19								
	BC					100,0%							
						19/19							
	BCT						100,0%						
						19/19							
BT							100,0%						
							19/19						
CA								100,0%					
								19/19					
CAT									100,0%				
									18/18				
CT											100,0%		
											19/19		

Do autor (2021).

A.5.2.2 1/2 Ciclo Pós Falta

Treinamento realizado com 300 *features*

- A acurácia média da validação cruzada foi 100,00%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,00%.
- O tempo total necessário para treinamento foi 9,605 segundos.
- A acurácia no conjunto de validação foi 100,00%.
- O tempo necessário para classificação do conjunto de validação foi 0,02 segundos.
- Acurácia em cada *fold*:

– *Fold* 1: 100,00%

- *Fold 2*: 100,00%
 - *Fold 3*: 100,00%
 - *Fold 4*: 100,00%
 - *Fold 5*: 100,00%
 - *Fold 6*: 100,00%
 - *Fold 7*: 100,00%
 - *Fold 8*: 100,00%
 - *Fold 9*: 100,00%
 - *Fold 10*: 100,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta BC com 0,012s.
 - O tipo de falta com menor tempo operacional foi a falta BT com 0,012s.

A Tabela 42 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1/2 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 42 – Matriz de confusão do método F com 1/2 ciclo pós falta e extração de características com *MiniRocket*.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0%											
		19/19											
	ABC		100,0%										
			18/18										
	ABT			100,0%									
				19/19									
	AT				100,0%								
					19/19								
	BC					100,0%							
						19/19							
	BCT						100,0%						
						19/19							
BT							100,0%						
							19/19						
CA								100,0%					
								19/19					
CAT									100,0%				
									18/18				
CT											100,0%		
											19/19		

Do autor (2021).

A.5.2.3 1/4 Ciclo Pós Falta

Treinamento realizado com 400 *features*

- A acurácia média da validação cruzada foi 100,00%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,00%.
- O tempo total necessário para treinamento foi 10,985 segundos.
- A acurácia no conjunto de validação foi 100,00%.
- O tempo necessário para classificação do conjunto de validação foi 0,045 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 100,00%
 - *Fold* 3: 100,00%

- *Fold 4*: 100,00%
 - *Fold 5*: 100,00%
 - *Fold 6*: 100,00%
 - *Fold 7*: 100,00%
 - *Fold 8*: 100,00%
 - *Fold 9*: 100,00%
 - *Fold 10*: 100,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta AB com 0,012s.
 - O tipo de falta com menor tempo operacional foi a falta BT com 0,012s.

A Tabela 43 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1/4 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 43 – Matriz de confusão do método F com 1/4 ciclo pós falta e extração de características com *MiniRocket*.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	100,0% 19/19									
	ABC		100,0% 18/18								
	ABT			100,0% 19/19							
	AT				100,0% 19/19						
	BC					100,0% 19/19					
	BCT						100,0% 19/19				
	BT							100,0% 19/19			
	CA								100,0% 19/19		
	CAT									100,0% 18/18	
	CT										100,0% 19/19

A.5.2.4 1/8 Ciclo Pós Falta

Treinamento realizado com 400 *features*

- A acurácia média da validação cruzada foi 99,87%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,40%.
- O tempo total necessário para treinamento foi 10,865 segundos.
- A acurácia no conjunto de validação foi 100,00%.
- O tempo necessário para classificação do conjunto de validação foi 0,034 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 100,00%
 - *Fold* 3: 100,00%
 - *Fold* 4: 100,00%
 - *Fold* 5: 98,67%
 - *Fold* 6: 100,00%
 - *Fold* 7: 100,00%
 - *Fold* 8: 100,00%
 - *Fold* 9: 100,00%
 - *Fold* 10: 100,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta AB com 0,012s.
 - O tipo de falta com menor tempo operacional foi a falta CAT com 0,012s.

A Tabela 44 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1/8 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 44 – Matriz de confusão do método F com 1/8 ciclo pós falta e extração de características com *MiniRocket*.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	100,0% 19/19									
	ABC		100,0% 18/18								
	ABT			100,0% 19/19							
	AT				100,0% 19/19						
	BC					100,0% 19/19					
	BCT						100,0% 19/19				
	BT							100,0% 19/19			
	CA								100,0% 19/19		
	CAT									100,0% 18/18	
	CT										100,0% 19/19

Do autor (2021).

A.5.2.5 1/16 Ciclo Pós Falta

Treinamento realizado com 300 *features*

- A acurácia média da validação cruzada foi 99,47%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,65%.
- O tempo total necessário para treinamento foi 9,084 segundos.
- A acurácia no conjunto de validação foi 99,47%.
- O tempo necessário para classificação do conjunto de validação foi 0,02 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 100,00%
 - *Fold* 3: 100,00%

- *Fold 4*: 98,67%
- *Fold 5*: 100,00%
- *Fold 6*: 98,67%
- *Fold 7*: 100,00%
- *Fold 8*: 98,67%
- *Fold 9*: 100,00%
- *Fold 10*: 98,67%

- O tipo de falta com maior tempo operacional foi a falta AB com 0,011s.
- O tipo de falta com menor tempo operacional foi a falta ABT com 0,011s.

A Tabela 45 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1/16 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 45 – Matriz de confusão do método F com 1/16 ciclo pós falta e extração de características com *MiniRocket*.

		Valores preditos										
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT	
Valores reais	AB	100,0% 19/19										
	ABC		100,0% 18/18									
	ABT			100,0% 19/19								
	AT				100,0% 19/19							
	BC					100,0% 19/19						
	BCT						100,0% 19/19					
	BT							100,0% 19/19				
	CA								100,0% 19/19			
	CAT									94,44% 17/18	5,56% 1/18	
	CT											100,0% 19/19

Do autor (2021).

A.5.2.6 1/32 Ciclo Pós Falta

Treinamento realizado com 500 *features*

- A acurácia média da validação cruzada foi 99,07%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,34%.
- O tempo total necessário para treinamento foi 10,775 segundos.
- A acurácia no conjunto de validação foi 98,94%.
- O tempo necessário para classificação do conjunto de validação foi 0,037 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 98,68%
 - *Fold* 3: 100,00%
 - *Fold* 4: 96,00%
 - *Fold* 5: 100,00%
 - *Fold* 6: 97,33%
 - *Fold* 7: 100,00%
 - *Fold* 8: 100,00%
 - *Fold* 9: 100,00%
 - *Fold* 10: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta ABT com 0,012s.
 - O tipo de falta com menor tempo operacional foi a falta BT com 0,012s.

A Tabela 46 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1/32 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 46 – Matriz de confusão do método F com 1/32 ciclo pós falta e extração de características com *MiniRocket*.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0% 19/19											
	ABC		100,0% 18/18										
	ABT			94,74% 18/19	5,26% 1/19								
	AT				100,0% 19/19								
	BC					100,0% 19/19							
	BCT						100,0% 19/19						
	BT							100,0% 19/19					
	CA								100,0% 19/19				
	CAT									94,44% 17/18	5,56% 1/18		
	CT											100,0% 19/19	

Do autor (2021).

A.5.2.7 1/64 Ciclo Pós Falta

Treinamento realizado com 300 *features*

- A acurácia média da validação cruzada foi 97,07%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,96%.
- O tempo total necessário para treinamento foi 9,244 segundos.
- A acurácia no conjunto de validação foi 97,87%.
- O tempo necessário para classificação do conjunto de validação foi 0,017 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 98,68%
 - *Fold* 2: 98,68%
 - *Fold* 3: 97,33%

- *Fold 4*: 93,33%
 - *Fold 5*: 94,67%
 - *Fold 6*: 94,67%
 - *Fold 7*: 98,67%
 - *Fold 8*: 98,67%
 - *Fold 9*: 97,33%
 - *Fold 10*: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta AB com 0.011s.
 - O tipo de falta com menor tempo operacional foi a falta CAT com 0.011s.

A Tabela 47 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1/64 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 47 – Matriz de confusão do método F com 1/64 ciclo pós falta e extração de características com *MiniRocket*.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	100,0% 19/19									
	ABC		100,0% 18/18								
	ABT			89,47% 17/19	10,53% 2/19						
	AT				100,0% 19/19						
	BC					100,0% 19/19					
	BCT						94,74% 18/19	5,26% 1/19			
	BT							100,0% 19/19			
	CA								100,0% 19/19		
	CAT									94,44% 17/18	5,56% 1/18
	CT										100,0% 19/19

Do autor (2021).

A.5.2.8 1/128 Ciclo Pós Falta

Treinamento realizado com 800 *features*

- A acurácia média da validação cruzada foi 97,34%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,89%.
- O tempo total necessário para treinamento foi 14,452 segundos.
- A acurácia no conjunto de validação foi 97,87%.
- O tempo necessário para classificação do conjunto de validação foi 0,054 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 97,37%
 - *Fold* 2: 97,37%
 - *Fold* 3: 100,00%
 - *Fold* 4: 93,33%
 - *Fold* 5: 97,33%
 - *Fold* 6: 94,67%
 - *Fold* 7: 97,33%
 - *Fold* 8: 98,67%
 - *Fold* 9: 98,67%
 - *Fold* 10: 98,67%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta AB com 0.014s.
 - O tipo de falta com menor tempo operacional foi a falta CA com 0.014s.

A Tabela 48 apresenta a matriz de confusão gerada pelo método F treinado com sinais com 1/128 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 48 – Matriz de confusão do método F com 1/128 ciclo pós falta e extração de características com *MiniRocket*.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0%											
		19/19											
	ABC		100,0%										
			18/18										
	ABT			94,74%	5,26%								
				18/19	1/19								
	AT				100,0%								
					19/19								
	BC					100,0%							
						19/19							
	BCT						94,74%	5,26%					
							18/19	1/19					
	BT							100,0%					
							19/19						
CA								100,0%					
								19/19					
CAT									94,44%	5,56%			
									17/18	1/18			
CT											94,74%		
											18/19		

Do autor (2021).

A.6 Método G

Por fim, o último método implementado que foi exposto na subseção 4.3.3.4 fez uso dos dados de corrente tanto do *dataset 1* quanto do *dataset 2*. Além disso, ele utilizou as mesmas técnicas do método F, ou seja, o pré-processamento foi realizado utilizando *Rocket* ou *MiniRocket* e a classificação foi conduzida pela regressão *ridge*. Nas subseções a seguir são apresentados os resultados que foram obtidos ao treinar e validar os dados utilizando-se esse método.

A.6.1 Rocket

Nesse método novamente foi utilizada a transformação *Rocket* variando-se a quantidade de *kernels*. Nesse processo registrou-se a acurácia de treinamento e validação e o tempo necessário para treinar e validar os modelos. Os resultados completos obtidos a partir da variação da quantidade de *kernels* são apresentados na Tabela 67 no apêndice B.

A.6.1.1 1 Ciclo Pós Falta

Treinando com 200 *kernels*

- A acurácia média da validação cruzada foi 99,73%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,33%.
- O tempo total necessário para treinamento foi 44,406 segundos.
- A acurácia no conjunto de validação foi 100,00%.
- O tempo necessário para classificação do conjunto de validação foi 0,023 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 100,00%
 - *Fold* 3: 100,00%
 - *Fold* 4: 100,00%
 - *Fold* 5: 100,00%
 - *Fold* 6: 99,33%
 - *Fold* 7: 99,33%
 - *Fold* 8: 100,00%
 - *Fold* 9: 99,33%
 - *Fold* 10: 99,33%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta BC com 0.014s.
 - O tipo de falta com menor tempo operacional foi a falta ABT com 0.013s.

A Tabela 49 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 49 – Matriz de confusão do método G com 1 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0% 38/38											
	ABC		100,0% 36/36										
	ABT			100,0% 38/38									
	AT				100,0% 38/38								
	BC					100,0% 38/38							
	BCT						100,0% 38/38						
	BT							100,0% 38/38					
	CA								100,0% 38/38				
	CAT									100,0% 36/36			
	CT											100,0% 38/38	

Do autor (2021).

A.6.1.2 1/2 Ciclo Pós Falta

Treinando com 400 *kernels*

- A acurácia média da validação cruzada foi 99,47%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,83%.
- O tempo total necessário para treinamento foi 56,53 segundos.
- A acurácia no conjunto de validação foi 100,00%.
- O tempo necessário para classificação do conjunto de validação foi 0,051 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 100,00%
 - *Fold* 3: 100,00%

- *Fold 4*: 100,00%
 - *Fold 5*: 100,00%
 - *Fold 6*: 99,33%
 - *Fold 7*: 99,33%
 - *Fold 8*: 98,67%
 - *Fold 9*: 97,33%
 - *Fold 10*: 100,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta BCT com 0.016s.
 - O tipo de falta com menor tempo operacional foi a falta BC com 0.016s.

A Tabela 50 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1/2 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 50 – Matriz de confusão do método G com 1/2 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	100,0% 38/38									
	ABC		100,0% 36/36								
	ABT			100,0% 38/38							
	AT				100,0% 38/38						
	BC					100,0% 38/38					
	BCT						100,0% 38/38				
	BT							100,0% 38/38			
	CA								100,0% 38/38		
	CAT									100,0% 36/36	
	CT										100,0% 38/38

Do autor (2021).

A.6.1.3 1/4 Ciclo Pós Falta

Treinando com 500 kernels

- A acurácia média da validação cruzada foi 99,27%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,76%.
- O tempo total necessário para treinamento foi 51,591 segundos.
- A acurácia no conjunto de validação foi 100,00%.
- O tempo necessário para classificação do conjunto de validação foi 0,072 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 99,34%
 - *Fold* 3: 100,00%
 - *Fold* 4: 99,33%
 - *Fold* 5: 98,00%
 - *Fold* 6: 98,67%
 - *Fold* 7: 98,00%
 - *Fold* 8: 100,00%
 - *Fold* 9: 99,33%
 - *Fold* 10: 100,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CAT com 0.020s.
 - O tipo de falta com menor tempo operacional foi a falta AB com 0.016s.

A Tabela 51 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1/4 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 51 – Matriz de confusão do método G com 1/4 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0% 38/38											
	ABC		100,0% 36/36										
	ABT			100,0% 38/38									
	AT				100,0% 38/38								
	BC					100,0% 38/38							
	BCT						100,0% 38/38						
	BT							100,0% 38/38					
	CA								100,0% 38/38				
	CAT									100,0% 36/36			
	CT											100,0% 38/38	

Do autor (2021).

A.6.1.4 1/8 Ciclo Pós Falta

Treinando com 500 kernels

- A acurácia média da validação cruzada foi 98,73%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,81%.
- O tempo total necessário para treinamento foi 43,662 segundos.
- A acurácia no conjunto de validação foi 98,94%.
- O tempo necessário para classificação do conjunto de validação foi 0,072 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 99,34%
 - *Fold* 2: 100,00%
 - *Fold* 3: 100,00%

- *Fold 4*: 98,00%
 - *Fold 5*: 98,00%
 - *Fold 6*: 99,33%
 - *Fold 7*: 98,00%
 - *Fold 8*: 98,67%
 - *Fold 9*: 98,00%
 - *Fold 10*: 98,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CA com 0.021s.
 - O tipo de falta com menor tempo operacional foi a falta ABT com 0.019s.

A Tabela 52 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1/8 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 52 – Matriz de confusão do método G com 1/8 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos										
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT	
Valores reais	AB	100,0% 38/38										
	ABC		100,0% 36/36									
	ABT			100,0% 38/38								
	AT				100,0% 38/38							
	BC					97,37% 37/38	2,63% 1/38					
	BCT						100,0% 38/38					
	BT							100,0% 38/38				
	CA								94,74% 36/38	5,26% 2/38		
	CAT									2,78% 1/36	97,22% 35/36	
	CT											100,0% 38/38

Do autor (2021).

A.6.1.5 1/16 Ciclo Pós Falta

Treinando com 800 kernels

- A acurácia média da validação cruzada foi 94,21%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,59%.
- O tempo total necessário para treinamento foi 53,245 segundos.
- A acurácia no conjunto de validação foi 94,15%.
- O tempo necessário para classificação do conjunto de validação foi 0,102 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 94,74%
 - *Fold* 2: 95,39%
 - *Fold* 3: 94,00%
 - *Fold* 4: 94,67%
 - *Fold* 5: 90,00%
 - *Fold* 6: 95,33%
 - *Fold* 7: 93,33%
 - *Fold* 8: 94,67%
 - *Fold* 9: 96,00%
 - *Fold* 10: 94,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CAT com 0.027s.
 - O tipo de falta com menor tempo operacional foi a falta ABT com 0.024s.

A Tabela 53 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1/16 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 53 – Matriz de confusão do método G com 1/16 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	100,0% 38/38									
	ABC	91,67% 33/36				5,56% 2/36		2,78% 1/36			
	ABT		86,84% 33/38	10,53% 4/38			2,63% 1/38				
	AT			100,0% 38/38							
	BC				94,74% 36/38	5,26% 2/38					
	BCT					97,37% 37/38			2,63% 1/38		
	BT		5,26% 2/38				94,74% 36/38				
	CA	2,63% 1/38						92,11% 35/38	5,26% 2/38		
	CAT			5,56% 2/36				2,78% 1/36	91,67% 33/36		
	CT						2,63% 1/38		5,26% 2/38	92,11% 35/38	

Do autor (2021).

A.6.1.6 1/32 Ciclo Pós Falta

Treinando com 800 *kernels*

- A acurácia média da validação cruzada foi 64,89%.
- O desvio padrão da acurácia nos 10 *folds* foi 2,45%.
- O tempo total necessário para treinamento foi 50,027 segundos.
- A acurácia no conjunto de validação foi 66,49%.
- O tempo necessário para classificação do conjunto de validação foi 0,105 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 63,82%
 - *Fold* 2: 65,13%
 - *Fold* 3: 64,00%

- *Fold 4*: 61,33%
- *Fold 5*: 63,33%
- *Fold 6*: 64,67%
- *Fold 7*: 68,67%
- *Fold 8*: 62,00%
- *Fold 9*: 67,33%
- *Fold 10*: 68,67%

- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CT com 0.026s.
 - O tipo de falta com menor tempo operacional foi a falta AT com 0.024s.

A Tabela 54 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1/32 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 54 – Matriz de confusão do método G com 1/32 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	73,68% 28/38	5,26% 2/38	7,89% 3/38		2,63% 1/38		7,89% 3/38		2,63% 1/38	
	ABC		66,67% 24/36	5,56% 2/36	2,78% 1/36	11,11% 4/36	2,78% 1/36	5,56% 2/36	5,56% 2/36		
	ABT	10,53% 4/38	2,63% 1/38	60,53% 23/38	13,16% 5/38		2,63% 1/38		2,63% 1/38	7,89% 3/38	
	AT		2,63% 1/38	2,63% 1/38	76,32% 29/38	2,63% 1/38		2,63% 1/38	5,26% 2/38	5,26% 2/38	2,63% 1/38
	BC	7,89% 3/38	5,26% 2/38			71,05% 27/38	7,89% 3/38	2,63% 1/38	2,63% 1/38		2,63% 1/38
	BCT		2,63% 1/38	5,26% 2/38		21,05% 8/38	60,53% 23/38	5,26% 2/38	5,26% 2/38		
	BT	2,63% 1/38	2,63% 1/38	2,63% 1/38	5,26% 2/38	5,26% 2/38	2,63% 1/38	68,42% 26/38	5,26% 2/38	2,63% 1/38	2,63% 1/38
	CA	2,63% 1/38	2,63% 1/38			5,26% 2/38	7,89% 3/38	7,89% 3/38	60,53% 23/38	10,53% 4/38	2,63% 1/38
	CAT	2,78% 1/36		2,78% 1/36	13,89% 5/36	2,78% 1/36	2,78% 1/36		5,56% 2/36	69,44% 25/36	
	CT	7,89% 3/38				7,89% 3/38	15,79% 6/38	2,63% 1/38	5,26% 2/38	2,63% 1/38	57,89% 22/38

Do autor (2021).

A.6.1.7 1/64 Ciclo Pós Falta

Treinando com 700 *kernels*

- A acurácia média da validação cruzada foi 54,39%.
- O desvio padrão da acurácia nos 10 *folds* foi 2,73%.
- O tempo total necessário para treinamento foi 44,816 segundos.
- A acurácia no conjunto de validação foi 56,12%.
- O tempo necessário para classificação do conjunto de validação foi 0,092 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 55,92%
 - *Fold* 2: 51,97%
 - *Fold* 3: 54,00%
 - *Fold* 4: 51,33%
 - *Fold* 5: 51,33%
 - *Fold* 6: 56,67%
 - *Fold* 7: 56,67%
 - *Fold* 8: 52,00%
 - *Fold* 9: 54,00%
 - *Fold* 10: 60,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CT com 0.025s.
 - O tipo de falta com menor tempo operacional foi a falta BCT com 0.023s.

A Tabela 55 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1/64 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 55 – Matriz de confusão do método G com 1/64 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	55,26% 21/38	5,26% 2/38	2,63% 1/38		13,16% 5/38		7,89% 3/38	7,89% 3/38	2,63% 1/38	5,26% 2/38
	ABC	8,33% 3/36	63,89% 23/36	2,78% 1/36		8,33% 3/36	2,78% 1/36	5,56% 2/36	5,56% 2/36	2,78% 1/36	
	ABT	2,63% 1/38	15,79% 6/38	36,84% 14/38	18,42% 7/38	7,89% 3/38	5,26% 2/38		10,53% 4/38		2,63% 1/38
	AT		5,26% 2/38	5,26% 2/38	71,05% 27/38	5,26% 2/38	2,63% 1/38		2,63% 1/38	5,26% 2/38	2,63% 1/38
	BC	10,53% 4/38		2,63% 1/38	2,63% 1/38	68,42% 26/38	5,26% 2/38	5,26% 2/38	5,26% 2/38		
	BCT			5,26% 2/38	2,63% 1/38	13,16% 5/38	57,89% 22/38	5,26% 2/38	5,26% 2/38	7,89% 3/38	2,63% 1/38
	BT	5,26% 2/38	10,53% 4/38		10,53% 4/38	5,26% 2/38	5,26% 2/38	55,26% 21/38	2,63% 1/38		5,26% 2/38
	CA		13,16% 5/38	5,26% 2/38	2,63% 1/38	7,89% 3/38	2,63% 1/38	7,89% 3/38	44,74% 17/38	15,79% 6/38	
	CAT	5,56% 2/36	2,78% 1/36		13,89% 5/36	5,56% 2/36	8,33% 3/36		2,78% 1/36	58,33% 21/36	2,78% 1/36
	CT	10,53% 4/38		2,63% 1/38	7,89% 3/38	7,89% 3/38	18,42% 7/38			2,63% 1/38	50,0% 19/38

Do autor (2021).

A.6.1.8 1/128 Ciclo Pós Falta

Treinando com 1000 kernels

- A acurácia média da validação cruzada foi 48,21%.
- O desvio padrão da acurácia nos 10 *folds* foi 2,65%.
- O tempo total necessário para treinamento foi 54,46 segundos.
- A acurácia no conjunto de validação foi 50,80%.
- O tempo necessário para classificação do conjunto de validação foi 0,103 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 48,03%
 - *Fold* 2: 44,08%
 - *Fold* 3: 44,67%

- *Fold 4*: 47,33%
 - *Fold 5*: 46,00%
 - *Fold 6*: 51,33%
 - *Fold 7*: 51,33%
 - *Fold 8*: 48,00%
 - *Fold 9*: 49,33%
 - *Fold 10*: 52,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CA com 0.029s.
 - O tipo de falta com menor tempo operacional foi a falta ABC com 0.027s.

A Tabela 56 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1/128 ciclo pós falta e extração de *features* usando *Rocket*.

Tabela 56 – Matriz de confusão do método G com 1/128 ciclo pós falta e extração de características com *Rocket*.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	57,89% 22/38		7,89% 3/38	5,26% 2/38	7,89% 3/38	2,63% 1/38	2,63% 1/38	5,26% 2/38	7,89% 3/38	2,63% 1/38
	ABC	2,78% 1/36	61,11% 22/36	2,78% 1/36		5,56% 2/36		8,33% 3/36	2,78% 1/36	8,33% 3/36	8,33% 3/36
	ABT	10,53% 4/38	2,63% 1/38	42,11% 16/38	15,79% 6/38	5,26% 2/38	10,53% 4/38		2,63% 1/38	2,63% 1/38	7,89% 3/38
	AT		13,16% 5/38	2,63% 1/38	52,63% 20/38	7,89% 3/38	7,89% 3/38	2,63% 1/38	2,63% 1/38	2,63% 1/38	7,89% 3/38
	BC	7,89% 3/38	2,63% 1/38	5,26% 2/38	2,63% 1/38	42,11% 16/38	13,16% 5/38	2,63% 1/38	5,26% 2/38	5,26% 2/38	13,16% 5/38
	BCT	2,63% 1/38	5,26% 2/38	2,63% 1/38	7,89% 3/38	7,89% 3/38	47,37% 18/38	5,26% 2/38	13,16% 5/38	2,63% 1/38	5,26% 2/38
	BT	5,26% 2/38	2,63% 1/38		5,26% 2/38		10,53% 4/38	55,26% 21/38	5,26% 2/38	15,79% 6/38	
	CA	2,63% 1/38		2,63% 1/38		10,53% 4/38	7,89% 3/38	2,63% 1/38	47,37% 18/38	21,05% 8/38	5,26% 2/38
	CAT	5,56% 2/36	2,78% 1/36	8,33% 3/36	5,56% 2/36	5,56% 2/36	5,56% 2/36	2,78% 1/36	5,56% 2/36	52,78% 19/36	5,56% 2/36
	CT		7,89% 3/38	5,26% 2/38	5,26% 2/38	2,63% 1/38	10,53% 4/38	7,89% 3/38	7,89% 3/38	2,63% 1/38	50,0% 19/38

Do autor (2021).

A.6.2 MiniRocket

Utilizando a transformação *MiniRocket* variou-se a quantidade de *features* utilizadas e foram obtidas as mais variadas acurácias de treinamento e validação, além de verificar o tempo necessário nessas etapas. Os resultados completos obtidos durante esses testes foram incluídos na Tabela 68 no apêndice B.

A.6.2.1 1 Ciclo Pós Falta

Treinando com 500 *features*

- A acurácia média da validação cruzada foi 99,73%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,61%.
- O tempo total necessário para treinamento foi 24,819 segundos.
- A acurácia no conjunto de validação foi 100,00%.
- O tempo necessário para classificação do conjunto de validação foi 0,044 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 100,00%
 - *Fold* 3: 100,00%
 - *Fold* 4: 99,33%
 - *Fold* 5: 100,00%
 - *Fold* 6: 100,00%
 - *Fold* 7: 100,00%
 - *Fold* 8: 98,00%
 - *Fold* 9: 100,00%
 - *Fold* 10: 100,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:

- O tipo de falta com maior tempo operacional foi a falta ABT com 0.014s.
- O tipo de falta com menor tempo operacional foi a falta BT com 0.014s.

A Tabela 57 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 57 – Matriz de confusão do método G com 1 ciclo pós falta e extração de características com *MiniRocket*.

	Valores preditos									
	AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais AB	100,0% 38/38									
ABC		100,0% 36/36								
ABT			100,0% 38/38							
AT				100,0% 38/38						
BC					100,0% 38/38					
BCT						100,0% 38/38				
BT							100,0% 38/38			
CA								100,0% 38/38		
CAT									100,0% 36/36	
CT										100,0% 38/38

Do autor (2021).

A.6.2.2 1/2 Ciclo Pós Falta

Treinando com 500 *features*

- A acurácia média da validação cruzada foi 99,47%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,40%.
- O tempo total necessário para treinamento foi 23,812 segundos.
- A acurácia no conjunto de validação foi 100,00%.
- O tempo necessário para classificação do conjunto de validação foi 0,031 segundos.

- Acurácia em cada *fold*:
 - *Fold* 1: 100,00%
 - *Fold* 2: 99,34%
 - *Fold* 3: 99,33%
 - *Fold* 4: 100,00%
 - *Fold* 5: 99,33%
 - *Fold* 6: 99,33%
 - *Fold* 7: 99,33%
 - *Fold* 8: 99,33%
 - *Fold* 9: 100,00%
 - *Fold* 10: 98,67%

- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CA com 0.013s.
 - O tipo de falta com menor tempo operacional foi a falta ABT com 0.013s.

A Tabela 58 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1/2 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 58 – Matriz de confusão do método G com 1/2 ciclo pós falta e extração de características com *MiniRocket*.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0% 38/38											
	ABC		100,0% 36/36										
	ABT			100,0% 38/38									
	AT				100,0% 38/38								
	BC					100,0% 38/38							
	BCT						100,0% 38/38						
	BT							100,0% 38/38					
	CA								100,0% 38/38				
	CAT									100,0% 36/36			
	CT											100,0% 38/38	

Do autor (2021).

A.6.2.3 1/4 Ciclo Pós Falta

Treinando com 500 features

- A acurácia média da validação cruzada foi 98,67%.
- O desvio padrão da acurácia nos 10 *folds* foi 0,60%.
- O tempo total necessário para treinamento foi 21,19 segundos.
- A acurácia no conjunto de validação foi 99,73%.
- O tempo necessário para classificação do conjunto de validação foi 0,048 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 99,34%
 - *Fold* 2: 98,68%
 - *Fold* 3: 98,67%

- *Fold 4*: 99,33%
 - *Fold 5*: 98,67%
 - *Fold 6*: 99,33%
 - *Fold 7*: 98,67%
 - *Fold 8*: 98,67%
 - *Fold 9*: 98,00%
 - *Fold 10*: 97,33%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta ABC com 0.013s.
 - O tipo de falta com menor tempo operacional foi a falta AB com 0.012s.

A Tabela 59 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1/4 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 59 – Matriz de confusão do método G com 1/4 ciclo pós falta e extração de características com *MiniRocket*.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0% 38/38											
	ABC		100,0% 36/36										
	ABT			100,0% 38/38									
	AT				97,37% 37/38		2,63% 1/38						
	BC					100,0% 38/38							
	BCT						100,0% 38/38						
	BT							100,0% 38/38					
	CA								100,0% 38/38				
	CAT									100,0% 36/36			
	CT											100,0% 38/38	

Do autor (2021).

A.6.2.4 1/8 Ciclo Pós Falta

Treinando com 900 *features*

- A acurácia média da validação cruzada foi 98,27%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,20%.
- O tempo total necessário para treinamento foi 32,711 segundos.
- A acurácia no conjunto de validação foi 99,47%.
- O tempo necessário para classificação do conjunto de validação foi 0,066 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 99,34%
 - *Fold* 2: 98,03%
 - *Fold* 3: 98,00%
 - *Fold* 4: 99,33%
 - *Fold* 5: 99,33%
 - *Fold* 6: 100,00%
 - *Fold* 7: 98,00%
 - *Fold* 8: 98,00%
 - *Fold* 9: 96,67%
 - *Fold* 10: 96,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta BC com 0.014s.
 - O tipo de falta com menor tempo operacional foi a falta AT com 0.013s.

A Tabela 60 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1/8 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 60 – Matriz de confusão do método G com 1/8 ciclo pós falta e extração de características com *MiniRocket*.

		Valores preditos											
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT		
Valores reais	AB	100,0% 38/38											
	ABC		100,0% 36/36										
	ABT			97,37% 37/38	2,63% 1/38								
	AT				100,0% 38/38								
	BC					100,0% 38/38							
	BCT						100,0% 38/38						
	BT							100,0% 38/38					
	CA								100,0% 38/38				
	CAT									100,0% 36/36			
	CT										2,63% 1/38	97,37% 37/38	

Do autor (2021).

A.6.2.5 1/16 Ciclo Pós Falta

Treinando com 700 features

- A acurácia média da validação cruzada foi 92,08%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,94%.
- O tempo total necessário para treinamento foi 27,777 segundos.
- A acurácia no conjunto de validação foi 93,09%.
- O tempo necessário para classificação do conjunto de validação foi 0,06 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 94,74%
 - *Fold* 2: 94,74%
 - *Fold* 3: 93,33%

- *Fold 4*: 92,67%
 - *Fold 5*: 88,67%
 - *Fold 6*: 92,67%
 - *Fold 7*: 92,00%
 - *Fold 8*: 92,00%
 - *Fold 9*: 90,67%
 - *Fold 10*: 89,33%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta ABT com 0.013s.
 - O tipo de falta com menor tempo operacional foi a falta BCT com 0.013s.

A Tabela 61 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1/16 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 61 – Matriz de confusão do método G com 1/16 ciclo pós falta e extração de características com *MiniRocket*.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	100,0% 38/38									
	ABC		94,44% 34/36			5,56% 2/36					
	ABT			86,84% 33/38	10,53% 4/38			2,63% 1/38			
	AT				97,37% 37/38				2,63% 1/38		
	BC		2,63% 1/38			97,37% 37/38					
	BCT		2,63% 1/38				86,84% 33/38	2,63% 1/38			7,89% 3/38
	BT			2,63% 1/38			2,63% 1/38	94,74% 36/38			
	CA								100,0% 38/38		
	CAT				11,11% 4/36				8,33% 3/36	77,78% 28/36	2,78% 1/36
	CT								5,26% 2/38		94,74% 36/38

Do autor (2021).

A.6.2.6 1/32 Ciclo Pós Falta

Treinando com 600 *features*

- A acurácia média da validação cruzada foi 62,30%.
- O desvio padrão da acurácia nos 10 *folds* foi 2,30%.
- O tempo total necessário para treinamento foi 24,333 segundos.
- A acurácia no conjunto de validação foi 65,96%.
- O tempo necessário para classificação do conjunto de validação foi 0,051 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 62,50%
 - *Fold* 2: 61,84%
 - *Fold* 3: 59,33%
 - *Fold* 4: 57,33%
 - *Fold* 5: 62,00%
 - *Fold* 6: 64,67%
 - *Fold* 7: 63,33%
 - *Fold* 8: 62,67%
 - *Fold* 9: 64,00%
 - *Fold* 10: 65,33%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta AB com 0.013s.
 - O tipo de falta com menor tempo operacional foi a falta AT com 0.012s.

A Tabela 62 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1/32 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 62 – Matriz de confusão do método G com 1/32 ciclo pós falta e extração de características com *MiniRocket*.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	65,79% 25/38	5,26% 2/38	7,89% 3/38		10,53% 4/38			10,53% 4/38		
	ABC	5,56% 2/36	75,0% 27/36			2,78% 1/36	5,56% 2/36	5,56% 2/36	2,78% 1/36		2,78% 1/36
	ABT	5,26% 2/38	10,53% 4/38	47,37% 18/38	13,16% 5/38	7,89% 3/38	2,63% 1/38	2,63% 1/38	7,89% 3/38		2,63% 1/38
	AT	7,89% 3/38	7,89% 3/38		71,05% 27/38	2,63% 1/38	5,26% 2/38		5,26% 2/38		
	BC	7,89% 3/38	5,26% 2/38			81,58% 31/38	2,63% 1/38	2,63% 1/38			
	BCT			7,89% 3/38	2,63% 1/38	7,89% 3/38	63,16% 24/38	2,63% 1/38	10,53% 4/38	2,63% 1/38	2,63% 1/38
	BT		10,53% 4/38	2,63% 1/38	15,79% 6/38	5,26% 2/38		65,79% 25/38			
	CA	2,63% 1/38	7,89% 3/38		2,63% 1/38		2,63% 1/38	2,63% 1/38	68,42% 26/38	5,26% 2/38	7,89% 3/38
	CAT	2,78% 1/36	2,78% 1/36		8,33% 3/36	8,33% 3/36	2,78% 1/36		5,56% 2/36	61,11% 22/36	8,33% 3/36
	CT	5,26% 2/38	2,63% 1/38	2,63% 1/38		5,26% 2/38	7,89% 3/38	2,63% 1/38	5,26% 2/38	7,89% 3/38	60,53% 23/38

Do autor (2021).

A.6.2.7 1/64 Ciclo Pós Falta

Treinando com 600 features

- A acurácia média da validação cruzada foi 53,79%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,49%.
- O tempo total necessário para treinamento foi 24,296 segundos.
- A acurácia no conjunto de validação foi 57,71%.
- O tempo necessário para classificação do conjunto de validação foi 0,057 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 53,29%
 - *Fold* 2: 55,26%
 - *Fold* 3: 52,67%

- *Fold 4*: 53,33%
 - *Fold 5*: 52,67%
 - *Fold 6*: 53,33%
 - *Fold 7*: 52,00%
 - *Fold 8*: 57,33%
 - *Fold 9*: 54,67%
 - *Fold 10*: 53,33%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta CAT com 0.013s.
 - O tipo de falta com menor tempo operacional foi a falta AT com 0.012s.

A Tabela 63 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1/64 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 63 – Matriz de confusão do método G com 1/64 ciclo pós falta e extração de características com *MiniRocket*.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	55,26% 21/38	13,16% 5/38	5,26% 2/38	2,63% 1/38	10,53% 4/38	2,63% 1/38		7,89% 3/38		2,63% 1/38
	ABC	8,33% 3/36	63,89% 23/36	5,56% 2/36			5,56% 2/36		8,33% 3/36	5,56% 2/36	2,78% 1/36
	ABT	5,26% 2/38	7,89% 3/38	50,0% 19/38	13,16% 5/38		5,26% 2/38	5,26% 2/38	5,26% 2/38	2,63% 1/38	5,26% 2/38
	AT		7,89% 3/38		68,42% 26/38	10,53% 4/38		5,26% 2/38	5,26% 2/38		2,63% 1/38
	BC	10,53% 4/38	7,89% 3/38	2,63% 1/38		60,53% 23/38		5,26% 2/38	5,26% 2/38	5,26% 2/38	2,63% 1/38
	BCT	2,63% 1/38	2,63% 1/38	5,26% 2/38	5,26% 2/38	5,26% 2/38	52,63% 20/38	10,53% 4/38	2,63% 1/38	5,26% 2/38	7,89% 3/38
	BT	5,26% 2/38	5,26% 2/38		7,89% 3/38	2,63% 1/38	7,89% 3/38	55,26% 21/38	7,89% 3/38	2,63% 1/38	5,26% 2/38
	CA		5,26% 2/38		7,89% 3/38	10,53% 4/38	5,26% 2/38	2,63% 1/38	57,89% 22/38	5,26% 2/38	5,26% 2/38
	CAT	5,56% 2/36	8,33% 3/36	5,56% 2/36	11,11% 4/36	2,78% 1/36		2,78% 1/36	8,33% 3/36	50,0% 18/36	5,56% 2/36
	CT	2,63% 1/38		5,26% 2/38		2,63% 1/38	18,42% 7/38	5,26% 2/38	2,63% 1/38		63,16% 24/38

Do autor (2021).

A.6.2.8 1/128 Ciclo Pós Falta

Treinando com 500 *features*

- A acurácia média da validação cruzada foi 54,05%.
- O desvio padrão da acurácia nos 10 *folds* foi 1,09%.
- O tempo total necessário para treinamento foi 20,496 segundos.
- A acurácia no conjunto de validação foi 55,59%.
- O tempo necessário para classificação do conjunto de validação foi 0,043 segundos.
- Acurácia em cada *fold*:
 - *Fold* 1: 53,95%
 - *Fold* 2: 55,26%
 - *Fold* 3: 54,67%
 - *Fold* 4: 53,33%
 - *Fold* 5: 54,00%
 - *Fold* 6: 52,00%
 - *Fold* 7: 52,67%
 - *Fold* 8: 55,33%
 - *Fold* 9: 55,33%
 - *Fold* 10: 54,00%
- Tipos de faltas que levaram maior e menor tempo para serem classificadas, considerando-se pré-processamento e classificação com os modelos previamente treinados:
 - O tipo de falta com maior tempo operacional foi a falta ABC com 0.013s.
 - O tipo de falta com menor tempo operacional foi a falta AB com 0.011s.

A Tabela 64 apresenta a matriz de confusão gerada pelo método G treinado com sinais com 1/128 ciclo pós falta e extração de *features* usando *MiniRocket*.

Tabela 64 – Matriz de confusão do método G com 1/128 ciclo pós falta e extração de características com *MiniRocket*.

		Valores preditos									
		AB	ABC	ABT	AT	BC	BCT	BT	CA	CAT	CT
Valores reais	AB	63,16% 24/38	10,53% 4/38	2,63% 1/38		2,63% 1/38	5,26% 2/38	2,63% 1/38	7,89% 3/38	2,63% 1/38	2,63% 1/38
	ABC		61,11% 22/36	2,78% 1/36		2,78% 1/36	5,56% 2/36	2,78% 1/36	11,11% 4/36	5,56% 2/36	8,33% 3/36
	ABT	10,53% 4/38	10,53% 4/38	50,0% 19/38	5,26% 2/38	10,53% 4/38	2,63% 1/38		10,53% 4/38		
	AT		10,53% 4/38	5,26% 2/38	50,0% 19/38	10,53% 4/38	2,63% 1/38		10,53% 4/38	2,63% 1/38	7,89% 3/38
	BC	10,53% 4/38	5,26% 2/38	2,63% 1/38	5,26% 2/38	52,63% 20/38		2,63% 1/38	2,63% 1/38	7,89% 3/38	10,53% 4/38
	BCT	10,53% 4/38		2,63% 1/38	2,63% 1/38	5,26% 2/38	50,0% 19/38	10,53% 4/38	10,53% 4/38	5,26% 2/38	2,63% 1/38
	BT	2,63% 1/38	13,16% 5/38		5,26% 2/38	2,63% 1/38	7,89% 3/38	55,26% 21/38	2,63% 1/38	7,89% 3/38	2,63% 1/38
	CA		7,89% 3/38	7,89% 3/38		7,89% 3/38	2,63% 1/38	5,26% 2/38	55,26% 21/38	5,26% 2/38	7,89% 3/38
	CAT	2,78% 1/36	8,33% 3/36	5,56% 2/36	2,78% 1/36	8,33% 3/36	2,78% 1/36		8,33% 3/36	52,78% 19/36	8,33% 3/36
	CT	2,63% 1/38	5,26% 2/38	7,89% 3/38		2,63% 1/38	5,26% 2/38	5,26% 2/38	5,26% 2/38		65,79% 25/38

Do autor (2021).

APÊNDICE B – Tabelas de resultados

Tabela 65 – Resultados obtidos usando *Rocket* no método F variando-se a quantidade de *kernels*.

Ciclo	<i>kernels</i>	Acurácia trein.	Acurácia val.	Tempo de trein.	Tempo de val.
1	100	99,60	100,00	13,965	0,012
1	200	99,87	99,47	23,165	0,021
1	300	100,00	100,00	31,621	0,04
1	400	100,00	100,00	41,481	0,046
1	500	100,00	100,00	48,4	0,052
1	600	100,00	100,00	55,801	0,062
1	700	100,00	100,00	65,593	0,077
1	800	100,00	100,00	72,661	0,075
1	900	100,00	100,00	84,835	0,084
1	1000	100,00	100,00	89,894	0,095
1	10000	100,00	100,00	811,934	0,489
1/2	100	99,20	100,00	10,475	0,012
1/2	200	99,47	100,00	16,691	0,021
1/2	300	99,73	100,00	22,138	0,041
1/2	400	99,87	100,00	28,459	0,048
1/2	500	99,87	100,00	32,387	0,061
1/2	600	99,87	100,00	36,506	0,066
1/2	700	99,87	100,00	42,74	0,087
1/2	800	99,87	100,00	47,143	0,078
1/2	900	100,00	100,00	53,736	0,077
1/2	1000	99,87	99,47	57,214	0,089
1/2	10000	99,87	100,00	489,612	0,493
1/4	100	97,07	98,94	8,955	0,012
1/4	200	98,53	100,00	13,414	0,021
1/4	300	98,80	99,47	17,466	0,041
1/4	400	99,73	100,00	21,47	0,041
1/4	500	99,34	99,47	24,97	0,054

1/4	600	99,73	100,00	26,887	0,083
1/4	700	99,60	100,00	31,611	0,066
1/4	800	99,73	99,47	34,036	0,07
1/4	900	99,47	100,00	38,548	0,083
1/4	1000	99,33	100,00	40,716	0,087
1/4	10000	99,47	100,00	330,312	0,494
1/8	100	96,14	94,15	8,418	0,012
1/8	200	98,00	99,47	11,677	0,018
1/8	300	98,40	99,47	14,803	0,042
1/8	400	97,34	97,87	18,664	0,048
1/8	500	99,07	100,00	20,451	0,052
1/8	600	98,93	99,47	22,589	0,063
1/8	700	99,07	100,00	25,339	0,105
1/8	800	99,20	98,94	28,715	0,071
1/8	900	99,07	98,94	31,241	0,083
1/8	1000	99,07	100,00	33,084	0,084
1/8	10000	99,73	100,00	250,855	0,489
1/16	100	96,27	93,09	8,085	0,012
1/16	200	96,80	96,81	10,914	0,022
1/16	300	96,54	96,28	13,807	0,041
1/16	400	95,47	96,81	16,657	0,055
1/16	500	97,07	96,28	18,603	0,056
1/16	600	97,60	96,81	20,702	0,055
1/16	700	97,74	97,34	23,67	0,08
1/16	800	97,34	97,87	24,244	0,1
1/16	900	97,87	98,40	27,911	0,076
1/16	1000	97,07	97,34	28,854	0,087
1/16	10000	98,94	98,40	212,223	0,492
1/32	100	93,22	93,62	7,69	0,012
1/32	200	93,48	94,15	10,576	0,021
1/32	300	93,75	93,62	13,334	0,042

1/32	400	94,02	95,74	16,228	0,065
1/32	500	94,55	96,81	17,46	0,054
1/32	600	94,81	95,74	19,524	0,063
1/32	700	94,81	96,81	21,553	0,066
1/32	800	94,41	96,81	22,832	0,069
1/32	900	95,87	97,34	25,897	0,085
1/32	1000	95,21	98,40	27,177	0,106
1/32	10000	96,94	97,87	192,665	0,494
1/64	100	90,42	90,96	7,702	0,013
1/64	200	90,96	93,62	10,024	0,021
1/64	300	93,22	94,15	13,203	0,041
1/64	400	93,22	93,09	15,394	0,047
1/64	500	93,22	96,28	16,846	0,075
1/64	600	92,69	95,74	18,858	0,065
1/64	700	93,22	94,68	20,787	0,069
1/64	800	93,75	96,28	22,624	0,097
1/64	900	94,01	96,81	25,0	0,077
1/64	1000	93,48	95,21	25,65	0,095
1/64	10000	95,47	97,34	182,676	0,491
1/128	100	90,02	89,89	7,54	0,012
1/128	200	90,96	87,77	10,043	0,018
1/128	300	91,36	93,62	12,819	0,03
1/128	400	91,49	91,49	15,229	0,044
1/128	500	91,75	94,68	16,92	0,058
1/128	600	93,48	94,68	18,781	0,067
1/128	700	92,95	92,55	20,559	0,07
1/128	800	93,09	94,15	22,401	0,08
1/128	900	93,22	94,68	24,381	0,099
1/128	1000	92,69	93,62	25,362	0,112
1/128	10000	95,47	95,74	177,606	0,495

Tabela 66 – Resultados obtidos usando *MiniRocket* no método F variando-se a quantidade de *features*.

Ciclo	<i>features</i>	Acurácia trein.	Acurácia val.	Tempo de trein.	Tempo de val.
1	100	100,00	100,00	7,018	0,011
1	200	100,00	100,00	8,349	0,015
1	300	100,00	100,00	9,824	0,02
1	400	100,00	100,00	11,296	0,035
1	500	100,00	100,00	13,286	0,041
1	600	100,00	100,00	15,422	0,049
1	700	100,00	100,00	17,126	0,054
1	800	100,00	100,00	18,563	0,062
1	900	100,00	100,00	19,824	0,076
1	1000	100,00	100,00	21,411	0,063
1	10000	100,00	100,00	39,56	0,357
1/2	100	99,87	99,47	6,58	0,009
1/2	200	99,87	99,47	8,224	0,013
1/2	300	100,00	100,00	9,605	0,02
1/2	400	100,00	100,00	10,929	0,032
1/2	500	100,00	100,00	12,59	0,041
1/2	600	100,00	100,00	14,78	0,049
1/2	700	100,00	100,00	16,423	0,052
1/2	800	100,00	100,00	17,33	0,075
1/2	900	100,00	100,00	17,15	0,062
1/2	1000	100,00	100,00	18,722	0,09
1/2	10000	100,00	100,00	33,836	0,358
1/4	100	98,00	98,94	6,571	0,009
1/4	200	99,60	99,47	8,047	0,013
1/4	300	99,47	100,00	9,46	0,02
1/4	400	100,00	100,00	10,985	0,045
1/4	500	99,87	100,00	11,337	0,04
1/4	600	100,00	100,00	14,308	0,048
1/4	700	100,00	100,00	16,239	0,052

1/4	800	100,00	100,00	16,314	0,055
1/4	900	100,00	100,00	17,064	0,061
1/4	1000	100,00	100,00	17,808	0,065
1/4	10000	100,00	100,00	29,787	0,358
1/8	100	97,48	97,87	6,696	0,009
1/8	200	99,47	99,47	7,989	0,014
1/8	300	99,60	100,00	9,094	0,02
1/8	400	99,87	100,00	10,865	0,034
1/8	500	99,47	99,47	11,097	0,039
1/8	600	99,87	100,00	14,061	0,047
1/8	700	99,87	100,00	14,809	0,052
1/8	800	99,73	99,47	15,82	0,047
1/8	900	99,87	99,47	16,011	0,058
1/8	1000	100,00	99,47	17,354	0,07
1/8	10000	99,87	100,00	24,876	0,359
1/16	100	96,01	96,28	6,693	0,027
1/16	200	98,67	96,81	7,906	0,013
1/16	300	99,47	99,47	9,084	0,02
1/16	400	99,33	99,47	10,776	0,033
1/16	500	99,87	98,94	11,197	0,027
1/16	600	99,34	99,47	14,259	0,047
1/16	700	99,60	98,94	14,805	0,052
1/16	800	99,87	98,40	15,877	0,055
1/16	900	99,87	98,94	15,982	0,071
1/16	1000	100,00	98,94	16,226	0,062
1/16	10000	99,73	99,47	22,519	0,356
1/32	100	96,41	92,02	6,67	0,009
1/32	200	97,61	97,87	7,874	0,013
1/32	300	99,07	97,34	9,172	0,02
1/32	400	98,40	97,87	10,587	0,023
1/32	500	99,07	98,94	10,775	0,037

1/32	600	99,07	98,94	13,096	0,048
1/32	700	98,94	98,94	14,529	0,051
1/32	800	99,07	97,87	14,521	0,076
1/32	900	99,07	98,40	15,628	0,069
1/32	1000	99,34	98,94	15,743	0,074
1/32	10000	98,94	98,94	21,272	0,355
1/64	100	94,54	93,62	6,656	0,009
1/64	200	96,68	97,87	7,584	0,015
1/64	300	97,07	97,87	9,244	0,017
1/64	400	98,00	96,81	10,547	0,024
1/64	500	97,60	97,34	10,868	0,039
1/64	600	97,87	95,74	13,036	0,048
1/64	700	97,74	97,34	14,703	0,068
1/64	800	98,01	97,34	14,565	0,054
1/64	900	98,01	97,34	15,749	0,066
1/64	1000	98,27	96,81	15,286	0,053
1/64	10000	98,27	98,40	21,694	0,356
1/128	100	92,55	93,09	6,608	0,009
1/128	200	95,61	93,09	7,693	0,013
1/128	300	96,81	95,21	9,179	0,02
1/128	400	96,54	95,21	10,611	0,023
1/128	500	97,74	96,81	10,994	0,04
1/128	600	97,47	96,81	13,046	0,045
1/128	700	97,34	97,34	14,352	0,052
1/128	800	97,34	97,87	14,452	0,054
1/128	900	97,74	97,87	15,757	0,07
1/128	1000	97,61	97,34	15,448	0,072
1/128	10000	97,87	98,40	21,533	0,357

Tabela 67 – Resultados obtidos usando *Rocket* no método G variando-se a quantidade de *kernels*.

Ciclo	<i>kernels</i>	Acurácia trein.	Acurácia val.	Tempo de trein.	Tempo de val.
1	100	99,20	98,94	26,397	0,013
1	200	99,73	100,00	44,406	0,023
1	300	99,73	100,00	60,936	0,048
1	400	99,80	100,00	83,206	0,059
1	500	99,93	100,00	99,74	0,072
1	600	100,00	100,00	118,785	0,077
1	700	99,87	100,00	135,407	0,081
1	800	99,93	100,00	148,55	0,103
1	900	100,00	100,00	172,077	0,095
1	1000	99,80	100,00	182,354	0,11
1	10000	100,00	100,00	1631,492	0,505
1/2	100	98,00	98,40	20,074	0,015
1/2	200	98,93	99,73	31,405	0,023
1/2	300	99,40	99,73	42,021	0,045
1/2	400	99,47	100,00	56,53	0,051
1/2	500	99,67	100,00	67,617	0,073
1/2	600	99,73	100,00	80,652	0,076
1/2	700	99,53	100,00	89,035	0,079
1/2	800	99,67	100,00	97,305	0,103
1/2	900	99,87	100,00	111,068	0,113
1/2	1000	99,73	100,00	117,311	0,111
1/2	10000	99,93	100,00	984,028	0,503
1/4	100	95,28	96,01	17,096	0,014
1/4	200	98,27	99,20	24,889	0,023
1/4	300	98,74	99,47	32,732	0,047
1/4	400	99,40	99,47	43,208	0,043
1/4	500	99,27	100,00	51,591	0,072
1/4	600	99,53	100,00	61,702	0,076
1/4	700	99,53	100,00	66,258	0,083

1/4	800	99,67	100,00	71,732	0,099
1/4	900	99,40	100,00	80,885	0,113
1/4	1000	99,60	100,00	85,117	0,127
1/4	10000	99,80	100,00	662,831	0,499
1/8	100	93,61	92,82	15,421	0,013
1/8	200	96,94	97,61	21,663	0,023
1/8	300	97,60	97,34	28,032	0,051
1/8	400	97,80	98,40	36,55	0,056
1/8	500	98,73	98,94	43,662	0,072
1/8	600	98,80	98,14	52,533	0,087
1/8	700	99,47	100,00	54,862	0,096
1/8	800	99,33	98,67	59,181	0,101
1/8	900	99,13	99,47	66,203	0,11
1/8	1000	99,33	99,73	69,12	0,124
1/8	10000	99,73	100,00	504,625	0,502
1/16	100	87,43	87,50	14,721	0,013
1/16	200	90,35	88,03	19,988	0,022
1/16	300	92,68	91,76	25,527	0,045
1/16	400	92,02	93,62	33,398	0,042
1/16	500	93,22	93,62	39,878	0,072
1/16	600	93,21	92,82	47,808	0,083
1/16	700	94,15	93,09	49,178	0,079
1/16	800	94,21	94,15	53,245	0,102
1/16	900	94,21	94,95	58,323	0,093
1/16	1000	94,68	96,01	61,167	0,111
1/16	10000	95,41	96,54	426,134	0,504
1/32	100	60,84	61,97	14,14	0,012
1/32	200	60,04	61,44	19,341	0,023
1/32	300	61,37	63,83	24,546	0,044
1/32	400	62,96	64,63	31,71	0,044
1/32	500	62,90	65,16	37,84	0,068

1/32	600	64,03	65,16	45,572	0,084
1/32	700	63,76	65,16	46,422	0,094
1/32	800	64,89	66,49	50,027	0,105
1/32	900	63,77	63,83	54,563	0,094
1/32	1000	64,30	64,89	57,121	0,109
1/32	10000	67,09	66,76	387,656	0,502
1/64	100	53,86	51,06	14,037	0,013
1/64	200	52,86	54,52	18,818	0,022
1/64	300	54,19	51,60	24,016	0,066
1/64	400	53,86	55,32	30,964	0,059
1/64	500	54,45	53,99	37,051	0,052
1/64	600	54,59	55,85	44,371	0,071
1/64	700	54,39	56,12	44,816	0,092
1/64	800	54,12	54,79	48,337	0,09
1/64	900	54,00	53,19	52,928	0,11
1/64	1000	53,92	55,32	55,398	0,121
1/64	10000	55,78	56,12	367,682	0,499
1/128	100	47,94	48,94	13,845	0,014
1/128	200	48,80	48,67	18,462	0,023
1/128	300	49,07	49,20	23,599	0,054
1/128	400	49,47	47,07	30,375	0,059
1/128	500	48,07	48,40	36,356	0,074
1/128	600	48,28	47,61	43,87	0,085
1/128	700	49,07	48,94	44,209	0,094
1/128	800	48,47	48,67	47,574	0,104
1/128	900	49,07	49,20	51,98	0,108
1/128	1000	48,21	50,80	54,46	0,103
1/128	10000	49,47	50,27	358,033	0,506

Tabela 68 – Resultados obtidos usando *MiniRocket* no método G variando-se a quantidade de *features*.

Ciclo	<i>features</i>	Acurácia trein.	Acurácia val.	Tempo de trein.	Tempo de val.
1	100	92,95	92,55	12,65	0,009
1	200	97,87	98,40	15,53	0,028
1	300	99,20	99,20	18,529	0,022
1	400	99,40	99,47	21,649	0,026
1	500	99,73	100,00	24,819	0,044
1	600	100,00	100,00	29,244	0,066
1	700	99,73	100,00	32,753	0,045
1	800	99,80	100,00	36,591	0,051
1	900	99,80	100,00	40,358	0,072
1	1000	100,00	100,00	44,338	0,078
1	10000	100,00	100,00	82,075	0,38
1/2	100	93,08	93,09	12,617	0,009
1/2	200	96,80	97,87	15,094	0,017
1/2	300	97,87	98,14	17,974	0,023
1/2	400	98,80	99,47	20,754	0,038
1/2	500	99,47	100,00	23,812	0,031
1/2	600	99,67	99,20	27,806	0,063
1/2	700	99,53	98,94	31,279	0,046
1/2	800	100,00	100,00	34,844	0,046
1/2	900	99,53	99,47	36,317	0,076
1/2	1000	99,80	99,73	40,128	0,067
1/2	10000	100,00	100,00	71,564	0,372
1/4	100	89,89	91,76	12,546	0,009
1/4	200	97,07	97,87	15,003	0,017
1/4	300	97,60	97,34	17,71	0,021
1/4	400	99,00	98,67	20,629	0,036
1/4	500	98,67	99,73	21,19	0,048
1/4	600	99,20	98,94	27,428	0,065
1/4	700	99,53	99,47	30,711	0,075

1/4	800	99,47	99,47	32,134	0,047
1/4	900	99,60	100,00	35,532	0,075
1/4	1000	99,54	100,00	37,262	0,058
1/4	10000	100,00	100,00	63,297	0,375
1/8	100	85,50	85,11	12,36	0,009
1/8	200	94,28	93,09	14,794	0,016
1/8	300	94,95	95,74	17,565	0,021
1/8	400	96,94	97,34	20,248	0,026
1/8	500	97,01	98,14	20,827	0,06
1/8	600	96,87	98,67	26,89	0,07
1/8	700	97,94	97,34	27,843	0,054
1/8	800	97,67	98,94	31,378	0,064
1/8	900	98,27	99,47	32,711	0,066
1/8	1000	98,27	99,47	36,442	0,058
1/8	10000	99,14	99,47	52,837	0,377
1/16	100	76,39	78,19	12,364	0,009
1/16	200	84,31	82,98	14,831	0,016
1/16	300	86,57	86,44	17,391	0,021
1/16	400	89,69	89,10	20,1	0,036
1/16	500	89,69	89,63	20,792	0,043
1/16	600	92,41	92,02	26,813	0,039
1/16	700	92,08	93,09	27,777	0,06
1/16	800	92,22	92,82	31,486	0,061
1/16	900	93,21	92,55	33,369	0,064
1/16	1000	93,14	93,09	34,208	0,057
1/16	10000	94,14	94,15	48,598	0,377
1/32	100	56,98	57,45	12,274	0,008
1/32	200	61,11	58,51	14,572	0,013
1/32	300	61,18	63,03	17,138	0,022
1/32	400	63,03	63,56	19,858	0,026
1/32	500	62,10	62,50	20,52	0,043

1/32	600	62,30	65,96	24,333	0,051
1/32	700	63,17	61,97	27,292	0,059
1/32	800	63,10	64,89	28,88	0,047
1/32	900	63,96	64,10	32,145	0,052
1/32	1000	63,83	64,10	33,933	0,076
1/32	10000	66,42	66,76	45,723	0,373
1/64	100	51,33	51,86	12,371	0,01
1/64	200	53,19	50,27	14,598	0,016
1/64	300	54,53	54,26	17,207	0,022
1/64	400	55,25	54,52	19,771	0,037
1/64	500	55,05	56,12	20,554	0,043
1/64	600	53,79	57,71	24,296	0,057
1/64	700	55,05	53,99	27,455	0,063
1/64	800	56,32	53,72	28,71	0,067
1/64	900	55,46	53,19	32,224	0,052
1/64	1000	56,32	55,85	33,938	0,056
1/64	10000	54,65	57,98	46,133	0,376
1/128	100	49,86	49,73	12,245	0,009
1/128	200	49,66	48,94	14,638	0,016
1/128	300	52,92	51,60	17,232	0,021
1/128	400	52,66	52,66	19,781	0,026
1/128	500	54,05	55,59	20,496	0,043
1/128	600	52,99	52,93	24,22	0,053
1/128	700	52,86	52,39	27,486	0,071
1/128	800	52,33	51,06	28,693	0,069
1/128	900	52,92	52,93	32,163	0,073
1/128	1000	51,79	52,13	33,699	0,067
1/128	10000	53,13	53,72	45,635	0,376

REFERÊNCIAS

- ABDOLLAHI, A.; SEYEDTABAI, S. Transmission line fault location estimation by fourier & wavelet transforms using ANN. **PEOCO 2010 - 4th International Power Engineering and Optimization Conference, Program and Abstracts**, n. June, p. 573–578, 2010.
- ABE, H.; NAKAYAMA, M. An ICA approach for extracting task-related components from EEG signals. **2006 International Symposium on Intelligent Signal Processing and Communications, ISPACS'06**, p. 862–865, 2007.
- AGGARWAL, C. C. **Neural Networks and Deep Learning**. Springer International Publishing, 2018. Disponível em: <<http://dx.doi.org/10.1007/978-3-319-94463-0>>.
- ALMEIDA, A. R. et al. **ICA feature extraction for the location and classification of faults in high-voltage transmission lines**. [S.l.]: Elsevier Ltd, 2017. 254–263 p.
- Alves Da Silva, A. P.; LIMA, A. C.; SOUZA, S. M. Fault location on transmission lines using complex-domain neural networks. **International Journal of Electrical Power and Energy Systems**, Elsevier Ltd, v. 43, n. 1, p. 720–727, 2012. ISSN 01420615. Disponível em: <<http://dx.doi.org/10.1016/j.ijepes.2012.05.046>>.
- ANDERSSON, A. **Paper review: Visualizing Data Using t-SNE**. almaan, 2021. Disponível em: <<https://almaan.github.io/post/2021-02-20-tsne/>>.
- BERRAR, D. Cross-validation. p. 1–4, 2017. Disponível em: <<http://arxiv.org/abs/1703.03167>>.
- BREIMAN, L. Bagging predictors. **Machine Learning**, v. 24, n. 2, p. 123–140, 1996. ISSN 1573-0565. Disponível em: <<https://doi.org/10.1023/A:1018054314350>>.
- BREIMAN, L. Random forests. **Machine Learning**, v. 45, n. 1, p. 5–32, Oct 2001. Disponível em: <<https://doi.org/10.1023/A:1010933404324>>.
- BREIMAN, L. et al. **Classification and Regression Trees**. Taylor & Francis, 1984. (The Wadsworth and Brooks-Cole statistics-probability series). ISBN 9780412048418. Disponível em: <<https://books.google.com.br/books?id=JwQx-WOmSyQC>>.
- BROWNLEE, J. **Master Machine Learning Algorithms**. ebook. [S.l.: s.n.], 2016. (1.12).
- CARVALHO, J. R. D. et al. A new transmission line protection approach using cumulants and artificial neural networks. **Journal of Control, Automation and Electrical Systems**, v. 25, n. 2, p. 237–251, 2014. ISSN 21953899.
- CHAITANYA, B.; YADAV, A. Decision tree aided travelling wave based fault section identification and location scheme for multi-terminal transmission lines. **Measurement**, v. 135, p. 312 – 322, 2019. ISSN 0263-2241. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0263224118311229>>.
- CHEN, K.; HUANG, C.; HE, J. Fault detection, classification and location for transmission lines and distribution systems: a review on the methods. **High Voltage**, v. 1, n. 1, p. 25–33, 2016. ISSN 2397-7264. Disponível em: <<https://digital-library.theiet.org/content/journals/10.1049/hve.2016.0005>>.

CHEN, K.; HUANG, C.; HE, J. Fault detection, classification and location for transmission lines and distribution systems: a review on the methods. **High Voltage**, v. 1, n. 1, p. 25–33, 2016. Disponível em: <<https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/hve.2016.0005>>.

CHEN, Y. Q.; FINK, O.; SANSAVINI, G. Combined fault location and classification for power transmission lines fault diagnosis with integrated feature extraction. **IEEE Transactions on Industrial Electronics**, v. 65, n. 1, p. 561–569, Jan 2018. ISSN 0278-0046.

CHOLLET, F. et al. **Keras**. [S.l.]: GitHub, 2015. <<https://github.com/fchollet/keras>>.

Circuit Globe. **Series Compensation**. 2021. Disponível em: <<https://circuitglobe.com/series-compensation.html>>.

COSTA, F.; SOUZA, B.; BRITO, N. Real-time detection of fault-induced transients in transmission lines. **Electronics Letters**, v. 46, p. 753 – 755, 06 2010.

COSTA, F. B.; SOUZA, B. A.; BRITO, N. S. Real-time classification of transmission line faults based on Maximal Overlap Discrete Wavelet Transform. **Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference**, p. 1–8, 2012. ISSN 21608555.

COURY, D. V.; LIMA, M. A.; CERQUEIRA, A. S. An ICA-based method for adaptive decomposition of multiple PQ disturbances. **Proceedings of International Conference on Harmonics and Quality of Power, ICHQP**, n. 1, p. 168–173, 2012. ISSN 15406008.

DAI, W.; JI, W. A mapreduce implementation of C4.5 decision tree algorithm. **International Journal of Database Theory and Application**, v. 7, n. 1, p. 49–60, 2014. ISSN 20054270.

DAS, D.; SINGH, N. K.; SINHA, A. K. A comparison of fourier transform and wavelet transform methods for detection and classification of faults on transmission lines. **2006 IEEE Power India Conference**, v. 2005, p. 620–626, 2005.

DASHTDAR, M.; DASHTI, R.; SHAKER, H. R. Distribution network fault section identification and fault location using artificial neural network. **2018 5th International Conference on Electrical and Electronics Engineering, ICEEE 2018**, n. May, p. 273–278, 2018.

DEMPSTER, A.; PETITJEAN, F.; WEBB, G. I. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. **Data Mining and Knowledge Discovery**, Springer Science and Business Media LLC, v. 34, n. 5, p. 1454–1495, Jul 2020. ISSN 1573-756X. Disponível em: <<http://dx.doi.org/10.1007/s10618-020-00701-z>>.

DEMPSTER, A.; SCHMIDT, D. F.; WEBB, G. I. Minirocket. **Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining**, ACM, Aug 2021. Disponível em: <<http://dx.doi.org/10.1145/3447548.3467231>>.

FAWAZ, H. I. et al. Deep learning for time series classification: a review. **Data Mining and Knowledge Discovery**, v. 33, n. 4, p. 917–963, Jul 2019. ISSN 1573-756X. Disponível em: <<https://doi.org/10.1007/s10618-019-00619-1>>.

FERREIRA, D. D. **Análise de Distúrbios Elétricos em Sistemas de Potência**. 233 p. Tese (Doutorado) — UFRJ/COPPE, 2010.

- FERREIRA, D. D. et al. Method based on independent component analysis for harmonic extraction from power system signals. **Electric Power Systems Research**, Elsevier B.V., v. 119, p. 19–24, 2015. ISSN 03787796. Disponível em: <<http://dx.doi.org/10.1016/j.epsr.2014.09.001>>.
- FERREIRA, D. D.; SEIXAS, J. M. D.; CERQUEIRA, A. S. A method based on independent component analysis for single and multiple power quality disturbance classification. **Electric Power Systems Research**, Elsevier B.V., v. 119, p. 425–431, 2015. ISSN 0378-7796. Disponível em: <<http://dx.doi.org/10.1016/j.epsr.2014.10.028>>.
- FONSECA, G. A. et al. Fault classification in transmission lines using random forest and notch filter. **Journal of Control, Automation and Electrical Systems**, Nov 2021. Disponível em: <<https://doi.org/10.1007/s40313-021-00844-4>>.
- GUO, X.; WU, X. Dynamic ICA Mixing Matrix. n. 60771033, p. 10–13, 2010.
- HAYKIN, S. **Redes Neurais - 2ed.** Bookman, 2001. ISBN 9788573077186. Disponível em: <<https://books.google.com.br/books?id=lBp0X5qfyjUC>>.
- HINTON, G.; ROWEIS, S. Stochastic neighbor embedding. In: **Advances in Neural Information Processing Systems 15**. [S.l.]: MIT Press, 2003. p. 833–840.
- JAMES, C. J.; ABASOLO, D.; GUPTA, D. Space-time ica versus ensemble ica for ictal eeg analysis with component differentiation via lempel-ziv complexity. In: **2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society**. [S.l.: s.n.], 2007. p. 5473–5476. ISSN 1094-687X.
- Junker, M.; Hoch, R.; Dengel, A. On the evaluation of document analysis components by recall, precision, and accuracy. In: **Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR '99 (Cat. No.PR00318)**. [S.l.: s.n.], 1999. p. 713–716.
- KAUR, C.; SINGH, P. EEG artifact suppression based on SOBI based ICA using wavelet thresholding. **2015 2nd International Conference on Recent Advances in Engineering and Computational Sciences, RAECS 2015**, n. December, p. 13–16, 2016.
- KUMAR, S. K.; SWAMY.R, S.; VENKATESH, V. Artificial Neural Network Based Method for Location and Classification of Faults on a Transmission Lines. v. 4, n. 1, p. 2250–3153, 2014. Disponível em: <www.ijserp.org>.
- LEICHTER, C. S. An adaptive system for improved identification and removal of noise from single trial eeg/meg via model order estimation in ica. In: **2006 Sixth International Conference on Hybrid Intelligent Systems (HIS'06)**. [S.l.: s.n.], 2006. p. 71–71.
- LIMA, M. A. et al. A novel method for power quality multiple disturbance decomposition based on Independent Component Analysis. **International Journal of Electrical Power and Energy Systems**, Elsevier Ltd, v. 42, n. 1, p. 593–604, 2012. ISSN 01420615. Disponível em: <<http://dx.doi.org/10.1016/j.ijepes.2012.05.004>>.
- LöNING, M. et al. **alan-turing-institute/sktime: v0.8.2**. Zenodo, 2021. Disponível em: <<https://zenodo.org/record/5610006>>.

- LÖNING, M. et al. **sktime: A Unified Interface for Machine Learning with Time Series**. 2019.
- MAATEN, L. van der; HINTON, G. **Visualizing Data using t-SNE**. 2008.
- MAHESHWARI, A.; AGARWAL, V.; SHARMA, S. K. Comparative Analysis of ANN-Based FL and Travelling Wave-Based FL for Location of Fault on Transmission Lines. **Journal of The Institution of Engineers (India): Series B**, 2019. ISSN 2250-2106.
- MAZZEI, P.; PENN, I.; ROBLES, F. **With Earthquakes and Storms, Puerto Rico's Power Grid Can't Catch a Break**. The New York Times, 2020. Disponível em: <<https://www.nytimes.com/2020/01/10/us/puerto-rico-electricity-power-earthquake.html?searchResultPosition=1>>.
- MICHELMANN, S. et al. Data-driven re-referencing of intracranial EEG based on independent component analysis (ICA). **Journal of Neuroscience Methods**, Elsevier B.V., v. 307, p. 125–137, 2018. ISSN 1872678X. Disponível em: <<https://doi.org/10.1016/j.jneumeth.2018.06.021>>.
- MUKHERJEE, A.; KUNDU, P. K.; DAS, A. Transmission line fault classification under high noise in signal: A direct pca-threshold-based approach. **Journal of The Institution of Engineers (India): Series B**, Jun 2021. ISSN 2250-2114. Disponível em: <<https://doi.org/10.1007/s40031-021-00601-w>>.
- NAGATA, E. A. et al. Voltage sag and swell detection and segmentation based on Independent Component Analysis. **Electric Power Systems Research**, Elsevier B.V., v. 155, p. 274–280, 2018. ISSN 03787796. Disponível em: <<http://dx.doi.org/10.1016/j.epsr.2017.10.029>>.
- NAVLANI, A. **Decision Tree Classification in Python**. 2018. Disponível em: <<https://www.datacamp.com/community/tutorials/decision-tree-classification-python>>.
- ONS. **Plano de Ampliações e Reforços nas Instalações de Transmissão do SIN**. [S.l.]: Operador Nacional do Sistema Elétrico, 2017.
- ONS. Indicadores de desempenho do sin junho de 2019. **Operador Nacional do Sistema Elétrico**, junho 2019. Disponível em: <http://www.ons.org.br/AcervoDigitalDocumentosEPublicacoes/201906_Relat%c3%b3rio_Indicadores_junho_resumido.pdf>. Acesso em: 07 out. 2019.
- ONS. Qualidade do suprimento. **Operador Nacional do Sistema Elétrico**, 03 out. 2019. Disponível em: <<http://www.ons.org.br/paginas/resultados-da-operacao/qualidade-do-suprimento>>. Acesso em: 03 out. 2019.
- ONS. Qualidade do suprimento - painéis. **Operador Nacional do Sistema Elétrico**, 03 out. 2019. Disponível em: <<http://www.ons.org.br/Paginas/resultados-da-operacao/qualidade-do-suprimento-paineis.aspx>>. Acesso em: 05 out. 2019.
- PARIKH, U. B.; DAS, B.; MAHESHWARI, R. Fault classification technique for series compensated transmission line using support vector machine. **International Journal of Electrical Power & Energy Systems**, v. 32, n. 6, p. 629–636, 2010. ISSN 0142-0615. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0142061509001938>>.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PRASAD, A.; Belwin Edward, J.; RAVI, K. A review on fault classification methodologies in power transmission systems: Part-II. **Journal of Electrical Systems and Information Technology**, Electronics Research Institute (ERI), v. 5, n. 1, p. 48–60, 2017b. ISSN 23147172. Disponível em: <<http://dx.doi.org/10.1016/j.jesit.2016.10.003>>.

PRASAD, A.; EDWARD, J. B.; RAVI, K. A review on fault classification methodologies in power transmission systems: Part-I. **Journal of Electrical Systems and Information Technology**, Electronics Research Institute (ERI), v. 5, n. 1, p. 48–60, 2017a. ISSN 23147172. Disponível em: <<http://dx.doi.org/10.1016/j.jesit.2017.01.004>>.

QUINLAN, J. R. Induction of decision trees. **Machine Learning**, v. 1, n. 1, p. 81–106, Mar 1986. ISSN 1573-0565. Disponível em: <<https://doi.org/10.1007/BF00116251>>.

RAJAGURU, H.; PRABHAKAR, S. K. Non linear ICA and logistic regression for classification of epilepsy from EEG signals. **Proceedings of the International Conference on Electronics, Communication and Aerospace Technology, ICECA 2017**, v. 2017-Janua, p. 577–580, 2017.

RAN, X.; ZHOU, Q. Power quality harmonic detection based on Fast-ICA. **PEAM 2011 - Proceedings: 2011 IEEE Power Engineering and Automation Conference**, v. 3, p. 26–29, 2011.

REFAEILZADEH, P.; TANG, L.; LIU, H. Cross-validation. In: _____. **Encyclopedia of Database Systems**. Boston, MA: Springer US, 2009. p. 532–538. ISBN 978-0-387-39940-9. Disponível em: <https://doi.org/10.1007/978-0-387-39940-9_565>.

REFAEILZADEH, P.; TANG, L.; LIU, H. Cross Validation. **Central Asia in Historical Perspective**, v. 25, p. 4–22, 2018.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, v. 323, n. 6088, p. 533–536, 1986. ISSN 1476-4687. Disponível em: <<https://doi.org/10.1038/323533a0>>.

SARAVANAN, N.; RATHINAM, A. A comparative study on ANN based fault location and classification technique for double circuit transmission line. **Proceedings - 4th International Conference on Computational Intelligence and Communication Networks, CICN 2012**, p. 824–830, 2012.

SCIKIT-LEARN, D. **Cross-validation: evaluating estimator performance**. scikit-learn developers, 2019. Disponível em: <https://scikit-learn.org/stable/modules/cross_validation.html>.

SCIKIT-LEARN, D. **Decision Trees - Scikit-learn Documentation**. [S.l.], 2019. Disponível em: <<https://scikit-learn.org/stable/modules/tree.html>>.

SCIKIT-LEARN, D. **Plot Ridge coefficients as a function of the regularization**. scikit-learn developers, 2021. Disponível em: <https://scikit-learn.org/stable/auto_examples/linear_model/plot_ridge_path.html#sphx-glr-auto-examples-linear-model-plot-ridge-path-py>.

SCIKIT-LEARN, D. **Ridge regression and classification**. scikit-learn developers, 2021. Disponível em: <https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression-and-classification>.

SHAIK, A. G.; PULIPAKA, R. R. V. A new wavelet based fault detection, classification and location in transmission lines. **International Journal of Electrical Power and Energy Systems**, Elsevier Ltd, v. 64, p. 35–40, 2015. ISSN 01420615. Disponível em: <<http://dx.doi.org/10.1016/j.ijepes.2014.06.065>>.

Shannon, C. E. A mathematical theory of communication. **The Bell System Technical Journal**, v. 27, n. 3, p. 379–423, 1948.

SHAO, X. et al. Application of id3 algorithm in knowledge acquisition for tolerance design. **Journal of Materials Processing Technology**, v. 117, p. 66–74, 11 2001.

SHIMAMOTO, S. et al. Utilization of independent component analysis for accurate pathological ripple detection in intracranial EEG recordings recorded extra- and intra-operatively. **Clinical Neurophysiology**, International Federation of Clinical Neurophysiology, v. 129, n. 1, p. 296–307, 2018. ISSN 18728952. Disponível em: <<https://doi.org/10.1016/j.clinph.2017.08.036>>.

SINGH, A.; PRAKASH, B. S.; CHANDRASEKARAN, K. A comparison of linear discriminant analysis and ridge classifier on twitter data. In: **2016 International Conference on Computing, Communication and Automation (ICCCA)**. [S.l.: s.n.], 2016. p. 133–138.

SINGH, S.; GUPTA, P. Comparative study ID3, cart and C4 . 5 Decision tree algorithm: a survey. **International Journal of Advanced Information Science and Technology (IJAIST)**, v. 3, n. 7, p. 47–52, 2014.

STEVENSON, W. **Elementos de análise de sistemas de potência**. McGraw Hill, 1978. Disponível em: <<https://books.google.com.br/books?id=cvGIHAAACAAJ>>.

Swain, P. H.; Hauska, H. The decision tree classifier: Design and potential. **IEEE Transactions on Geoscience Electronics**, v. 15, n. 3, p. 142–147, July 1977. ISSN 0018-9413.

SWETAPADMA, A.; YADAV, A. A Novel Decision Tree Regression-Based Fault Distance Estimation Scheme for Transmission Lines. **IEEE Transactions on Power Delivery**, v. 32, n. 1, p. 234–245, 2017. ISSN 08858977.

TANTRAY, A. K. **Snow exposes faults in NHPC's transmission line**. 2019. Disponível em: <<https://www.tribuneindia.com/news/archive/snow-exposes-faults-in-nhpc-s-transmission-line-727566>>.

TING, K. M. Confusion matrix. In: _____. **Encyclopedia of Machine Learning and Data Mining**. Boston, MA: Springer US, 2017. p. 260–260. ISBN 978-1-4899-7687-1. Disponível em: <https://doi.org/10.1007/978-1-4899-7687-1_50>.

WATT REZA BORHANI, A. K. J. **Machine Learning Refined: Foundations, Algorithms, and Applications**. 1. ed. [S.l.]: Cambridge University Press, 2016. ISBN 9781107123526; 1107123526.

YADAV, A.; SWETAPADMA, A. A novel transmission line relaying scheme for fault detection and classification using wavelet transform and linear discriminant analysis. **Ain Shams Engineering Journal**, Elsevier, v. 6, n. 1, p. 199–209, mar 2015. ISSN 2090-4479. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2090447914001336>>.

ZHOU, W. ECG Artifacts from EEG Using ICA. **Electronics**, p. 206–207, 2002.

ZHU, D. H.; TONG, J. J.; CHEN, Y. Q. An ICA-based method for automatic eye blink artifact correction in multi-channel EEG. **5th Int. Conference on Information Technology and Applications in Biomedicine, ITAB 2008 in conjunction with 2nd Int. Symposium and Summer School on Biomedical and Health Engineering, IS3BHE 2008**, p. 338–341, 2008.