



**UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**JEAN CARLOS AROUCHE FREIRE**

**ANÁLISE DE DESEMPENHO DE ALGORITMOS PARA  
CLASSIFICAÇÃO DE SEQUÊNCIAS REPRESENTANDO FALTAS DO  
TIPO CURTO-CIRCUITO EM LINHAS DE TRANSMISSÃO DE  
ENERGIA ELÉTRICA**

UFPA / ITEC / PPGEE

Belém - Pará

2019



**UNIVERSIDADE FEDERAL DO PARÁ**  
**INSTITUTO DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

JEAN CARLOS AROUCHE FREIRE

**ANÁLISE DE DESEMPENHO DE ALGORITMOS PARA**  
**CLASSIFICAÇÃO DE SEQUÊNCIAS REPRESENTANDO FALTAS DO**  
**TIPO CURTO-CIRCUITO EM LINHAS DE TRANSMISSÃO DE**  
**ENERGIA ELÉTRICA**

Tese de Doutorado apresentado ao Programa de Pós-Graduação em Engenharia Elétrica. Instituto de Tecnologia. Universidade Federal do Pará.

Área de Concentração: Inteligência Computacional.

Orientadora: Profa. Dra. Adriana Rosa Garcez Castro.

UFPA / ITEC / PPGEE

Belém - Pará

2019

JEAN CARLOS AROUCHE FREIRE

**ANÁLISE DE DESEMPENHO DE ALGORITMOS PARA  
CLASSIFICAÇÃO DE SEQUÊNCIAS REPRESENTANDO FALTAS DO  
TIPO CURTO-CIRCUITO EM LINHAS DE TRANSMISSÃO DE  
ENERGIA ELÉTRICA**

Tese de Doutorado apresentado ao Programa de Pós-Graduação em Engenharia Elétrica. Instituto de Tecnologia. Universidade Federal do Pará.

Área de Concentração: Inteligência Computacional.

Orientadora: Profa. Dra. Adriana Rosa Garcez Castro.

Data de defesa: Belém, 05 de Dezembro de 2019.

**BANCA EXAMINADORA**

---

**Profa. Dra. Adriana Rosa Garcez Castro** - Orientadora  
Programa de Pós-Graduação em Engenharia Elétrica  
Universidade Federal do Pará

---

**Prof. Dr. Jefferson Magalhães de Moraes** - Co-Orientador  
Programa de Pós-Graduação em Ciência da Computação  
Universidade Federal do Pará

---

**Prof. Dr. Marcos Paulo Alves de Sousa** - Membro externo  
Centro de Estudos Superiores do Pará

---

**Prof. Dr. Orlando Shigueo Ohasi Junior** - Membro externo  
Universidade Federal Rural da Amazônia

---

**Prof. Dr. Ubiratan Holanda Bezerra** - Membro interno  
Universidade Federal do Pará

UFPA / ITEC / PPGEE

Belém - Pará

2019

À minha esposa...  
À minha mãe e toda minha família...  
Com todo amor e carinho.

# AGRADECIMENTOS

Agradeço primeiro à Lei Divina, por ter me dado a oportunidade de minha existência, pela minha saúde, pela minha família, pela minha esposa e por tudo que conquistei ao longo desses anos.

À minha esposa Ilza Léia Ramos Arouche, pelo apoio, compreensão e compartilhamento dos momentos alegres e difíceis de minha vida. Uma eterna companheira que Deus me agraciou em colocá-la no meu caminho.

À minha família, especialmente à minha mãe, por ter sempre cuidado de mim em todos os momentos da minha vida com sua paciência, carinho e apreço.

A minha orientadora, Profa. Dra. Adriana Rosa Garcez Castro, por todo seu conhecimento, compreensão, dedicação e contribuição de orientação para o desenvolvimento deste trabalho, e principalmente por ter acreditado em mim.

À meu co-orientador Prof. Dr. Jefferson Magalhães de Moraes, por ter dado seu tempo, conhecimento, compreensão e coorientação em todos momentos que precisei.

A meus colegas de grupo de pesquisas, professores e todos auxiliares do Instituto de Tecnologia (ITEC) e do Instituto de Ciência Exatas e Naturais (ICEN), pelo apoio, orientação, contribuição e dedicação para o desenvolvimento desse trabalho.

Ao Programa de Pós-graduação em Engenharia Elétrica (PPGEE) da Universidade Federal do Pará, especialmente a Profa. Dra. Adriana Rosa Garcez Castro, por ter me aceitado como orientando e que tornou possível a realização desse trabalho.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), por contribuir de forma direta e indiretamente para a realização do meu estudo, principalmente na parte financeira.

A todos aqueles que não foram citados, porém que de certa forma contribuíram para realização do meu estudo.

*“O que dá, recebe, e quanto mais der, mais receberá, mas o que nada dá, até o que tem lhe será tirado. Esta é a Lei.”  
(Samael Aun Weor)*

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>16</b>
<b>1.1</b>	<b>Motivação e descrição geral do problema . . . . .</b>	<b>16</b>
<b>1.2</b>	<b>Objetivos . . . . .</b>	<b>18</b>
1.2.1	Objetivo Geral . . . . .	18
1.2.2	Objetivos Específicos . . . . .	18
<b>1.3</b>	<b>Estado da arte . . . . .</b>	<b>19</b>
<b>1.4</b>	<b>Publicações realizadas . . . . .</b>	<b>23</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>25</b>
<b>2.1</b>	<b>Faltas em Linhas de Transmissão . . . . .</b>	<b>25</b>
<b>2.2</b>	<b>Classificação de Faltas em Linhas de Transmissão . . . . .</b>	<b>27</b>
<b>2.3</b>	<b>Técnicas para desenvolvimento de Classificadores Convencionais . . . . .</b>	<b>29</b>
2.3.1	Redes Neurais Artificiais . . . . .	29
2.3.2	Máquinas de Vetores de Suporte . . . . .	31
2.3.3	K-Vizinhos mais próximos . . . . .	32
2.3.4	<i>Random forest</i> . . . . .	33
<b>2.4</b>	<b>Técnicas para desenvolvimento de Classificadores baseados em Sequências(variável) . . . . .</b>	<b>34</b>
2.4.1	K-vizinhos mais próximos com Alinhamento Temporal Dinâmico . . . . .	34
2.4.2	Modelo Oculto de Markov . . . . .	37
2.4.3	Processo de Treinamento com o algoritmo HMM . . . . .	38
2.4.4	Processo de Teste do algoritmo HMM . . . . .	41
<b>2.5</b>	<b>Classificadores baseados na Arquitetura de Sequência baseada em Quadros . . . . .</b>	<b>42</b>
2.5.1	<i>Front ends</i> . . . . .	43
2.5.2	Direto da forma de onda - <i>Raw</i> . . . . .	43
2.5.3	<i>Front end RMS</i> . . . . .	45
2.5.4	<i>Front ends wavelets</i> . . . . .	45
2.5.5	Concatenação de <i>fronts ends</i> . . . . .	49
<b>2.6</b>	<b>Medidas de desempenho adotadas para os classificadores . . . . .</b>	<b>50</b>
2.6.1	Taxa de Erro . . . . .	50
2.6.2	Custo computacional . . . . .	50
2.6.3	Testes de significância estatística . . . . .	51
<b>3</b>	<b>METODOLOGIA DO ESTUDO . . . . .</b>	<b>53</b>
<b>3.1</b>	<b>Base de dados UFPAFaults . . . . .</b>	<b>53</b>
<b>3.2</b>	<b>Configurações Gerais dos Experimentos para os algoritmos KNN-DTW, HMM e para a arquitetura FBSC . . . . .</b>	<b>55</b>
3.2.1	Configurações específicas para os experimentos com o algoritmo KNN-DTW	58

3.2.2	Configurações específicas para os experimentos com a arquitetura FBSC . . .	58
3.2.3	Configurações específicas para os experimentos com o algoritmo HMM . . .	59
<b>4</b>	<b>RESULTADOS E ANÁLISES . . . . .</b>	<b>60</b>
<b>4.1</b>	<b>Resultados dos Experimentos do algoritmo KNN-DTW . . . . .</b>	<b>60</b>
<b>4.2</b>	<b>Resultados dos Experimentos da arquitetura FBSC . . . . .</b>	<b>60</b>
<b>4.3</b>	<b>Resultado dos Experimentos do algoritmo HMM . . . . .</b>	<b>63</b>
<b>4.4</b>	<b>Comparação dos Resultados entre a arquitetura FBSC, o algoritmo KNN-DTW e o algoritmo HMM . . . . .</b>	<b>67</b>
<b>5</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>69</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>71</b>
<b>A</b>	<b>APÊNDICE A . . . . .</b>	<b>76</b>
<b>B</b>	<b>APÊNDICE B . . . . .</b>	<b>77</b>
<b>C</b>	<b>APÊNDICE C . . . . .</b>	<b>78</b>

## LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de um sistema elétrico de potência. . . . .	16
Figura 2 – Exemplo de um sistema elétrico de potência. . . . .	25
Figura 3 – Linha de Transmissão Trifásica em um SEP. . . . .	26
Figura 4 – Sinais de medição fasorial em três fases na ocorrência de uma falta. . . . .	27
Figura 5 – Estrutura de um neurônio artificial. . . . .	30
Figura 6 – Rede Neural Perceptron Multicamadas. Fonte: (HAYKIN et al., 2009) . . . . .	31
Figura 7 – Exemplo de estrutura de uma árvore de decisão. . . . .	33
Figura 8 – Monotocidade. . . . .	35
Figura 9 – Continuidade. . . . .	35
Figura 10 – Condição de contorno. . . . .	35
Figura 11 – Alinhamento de janela. . . . .	36
Figura 12 – Restrição de inclinação. . . . .	36
Figura 13 – Sequência encontrada após o cálculo DTW. . . . .	37
Figura 14 – Diagrama da Relação entre Estados Ocultos $C^{(t)}$ e Valores Observados $X^{(t)}$ de um modelo HMM. . . . .	38
Figura 15 – Estrutura do algoritmo HMM adotado no Estudo com Topologia Esquerda-Direita. . . . .	39
Figura 16 – Fluxo de processamento da arquitetura FBSC. A saída do classificador de sequência $G(Z)$ depende das decisões do classificador $F(z_n)$ . . . . .	42
Figura 17 – Demonstração do <i>front end</i> Raw, organizando os vetores de padrão $\mathbf{z}$ . Neste exemplo, $L = 2$ e $S = 2$ (não há sobreposição) e a matriz obtida pela aplicação do <i>front end</i> possui dimensão $K = 12$ com quadros da falta “ABT” e um quadro falta “BC”. . . . .	44
Figura 18 – Filtragem de um estágio para geração de aproximações (A) e detalhes (D) de um sinal (S). . . . .	46
Figura 19 – Exemplo de decomposição de 3 níveis. . . . .	46
Figura 20 – Decomposição wavelet a partir da função wavedec do MATLAB. . . . .	47
Figura 21 – Exemplo de uma decomposição <i>wavelet</i> aplicada aos sinais de tensão das fases A e B de uma falta AB simulada no intervalo de 1s. A <i>wavelet</i> mãe é uma <i>Daubechies</i> 4 com três níveis de decomposição ( $\gamma = 3$ ). . . . .	48
Figura 22 – Processo de concatenação dos <i>front ends</i> , mostrando o número de parâmetros $K$ para cada <i>front end</i> , considerando $L_{min} = 9$ e $S_{min} = 4$ . . . . .	49
Figura 23 – Metodologia do Estudo. . . . .	53
Figura 24 – Arquivo ASCII representando as informações associadas disponibilizada na base UFPAFaults. . . . .	55
Figura 25 – Formas de onda no momento de uma falta AB. . . . .	56
Figura 26 – Sinal com reamostragem e normalização. . . . .	57

Figura 27 – Tempo dos experimentos do algoritmo KNN-DTW. . . . .	61
Figura 28 – Melhores Resultados dos Experimentos de Classificação da Arquitetura FBSC. . . . .	62
Figura 29 – Tempo de execução para os classificadores FBSC. . . . .	62

## LISTA DE TABELAS

Tabela 1	– Mapeamento dos trabalhos relacionados. . . . .	23
Tabela 2	– Exemplo de Classificação da Máxima Verossimilhança de uma Classe. . . . .	41
Tabela 3	– Taxa de erro (%) do KNN-DTW para a base de teste, de acordo com número de vizinhos e o números de amostras de treino. . . . .	60
Tabela 4	– Resultado do Grid de Seleção do Modelo dos Classificadores Convencionais. . . . .	61
Tabela 5	– Valores Utilizados dos Parâmetros do algoritmo HMM, Taxa de Erro (na base de teste) e Média da Taxa de Erro para as bases de treino de 100 à 1000. . . . .	64
Tabela 6	– $E_i$ , $M_v$ , $AE_i$ e $G_{Ac}$ nas bases de 100 à 1000. . . . .	65
Tabela 7	– Tempo de execução para o classificador HMM. . . . .	66
Tabela 8	– Melhores Resultados dos Experimentos de Classificação e o Tempo de Execução em Segundos do Custo Computacional da arquitetura FBSC, KNN-DTW e HMM. . . . .	67
Tabela 9	– Resultados da Comparação de Teste Estatístico entre a arquitetura FBSC com concatFrontEnd associada aos Classificadores RNA, SVN, KNN e RF e os algoritmos HMM e KNN-DTW com Significância de $\alpha = 5\%$ . . . . .	68
Tabela 10	– Melhores Resultados dos Experimentos de Classificação da Arquitetura FBSC . . . . .	77
Tabela 11	– Tempo de execução para os classificadores FBSC. . . . .	78

## LISTA DE ABREVIATURAS E SIGLAS

ANN	<i>Artificial Neural Network</i> – Rede Neural Artificial
ARFF	<i>Attribute-Relation File Format</i>
ATP	<i>Alternative Transient Program</i>
DTW	<i>Dynamic Time Warping</i>
DWT	<i>Discrete Wavelet Transform</i> – Transformada Discreta de Wavelet
FBSC	<i>Frame Based-Sequence Classification</i> – Classificador de Sequencia Baseado em Quadros
HMM	<i>Hidden Markov Models</i> – Cadeias Ocultas de Markov
IBK	<i>Instance-Bases Learning With Parameter K</i>
KNN	<i>K-Nearest Neighbors</i> – K-vizinhos mais próximos
MLP	<i>Multilayer Perceptron</i>
PCA	<i>Principal Component Analysis</i> – Componente de Análise Principal
QEE	Qualidade de Energia Elétrica
RBF	<i>Radial Basis Functions</i> – Funções de Base Radial
RMS	<i>Root Mean Square</i>
SVM	<i>Support Vector Machine</i> – Máquinas de Vetores de Suporte
WEKA	<i>Waikato Environment for Knowledge Analysis</i>
GPL	General Public license
$x(t)$	Forma de onda de tensão, $t = 1, \dots, T$
$K$	Número de linhas de $Z$ , onde $K = Q \times L$
$Z$	Matriz chamada “ <i>instance</i> ”, a qual corresponde à matriz $\hat{Z}$ redimensionada, de maneira a ter dimensão $K \times N_n$ , por conveniência
$f_s$	Frequência de amostragem
$L$	Número de amostras (tamanho) do quadro ( <i>frames</i> )
$N_n$	Número total de quadros

$X_n$	Matriz de dimensão $Q \times T_n$ representando o n-ésimo evento em uma base de dados
$T_n$	Número de amostras multidimensionais do n-ésimo evento
$Q$	Número de sinais de tensão e corrente nas fases A, B e C
$y$	Rótulo ou classe, a qual corresponde à saída de um classificador
$G$	Classificador de sequências
$F$	Classificador Convencional
$F$	Matriz $Q \times L$ chamada quadro ( <i>frame</i> ), a qual aglutina $L$ amostras de $X$
	$\hat{Z}$ - Matriz $Q \times L_N$ correspondente à concatenação de todos os quadros de $X$
$S$	Deslocamento do quadro: número de amostras entre as amostras de início de quadros consecutivos
$Es$	Taxa de erro de classificação para o módulo de sequência (pós-falta)
$R$	Número de exemplos em um conjunto de teste.
$I$	Função indicador
$K$	Número de vizinhos mais próximos no classificador KNN
$z[n]$	n-ésimo valor RMS
$a$	Coefficiente de aproximação da transformada <i>wavelet</i>
$d$	Coefficiente de detalhe da transformada <i>wavelet</i>
$Y$	Número de estágios de filtragem e decimação de uma decomposição <i>wavelet</i> diádica.
$L_{min}$	Tamanho do quadro para os coeficientes <i>wavelet</i> com menor $f_s$
$S_{min}$	Deslocamento para os coeficientes <i>wavelet</i> com menor $f_s$
$T_a$	Número de amostras no coeficiente de aproximação $a$

## RESUMO

A manutenção da qualidade de energia em sistemas elétricos de potência depende do tratamento dos principais distúrbios que possam surgir em sua geração, transmissão e distribuição. Dentro deste contexto, muitos estudos vêm sendo desenvolvidos com o objetivo de realizar a detecção e classificação de faltas do tipo curto-circuito em sistemas elétricos através da análise do comportamento do sinal elétrico. Os sistemas de classificação de faltas em linha de transmissão podem ser divididos em dois tipos: sistemas de classificação *on-line* e pós-falta. No cenário pós-falta as sequências do sinal a serem avaliadas para a classificação possuem comprimento (duração) variável. Na classificação de sequências é possível utilizar classificadores convencionais tais como Redes Neurais Artificiais, Máquinas de Vetores de Suporte, K-vizinhos mais próximos e Árvore de Decisão (Floresta aleatória). Nestes casos, o processo de classificação geralmente requer um pré-processamento das sequências ou um estágio de *front end* que converta os dados bruto em parâmetros sensíveis para alimentar o classificador, o que pode aumentar o custo computacional do sistema de classificação. Uma alternativa para este problema é a arquitetura de classificação de sequências baseada em quadros (*FBSC - Frame Based Sequence Classification*). O problema da arquitetura FBSC é que esta possui muitos graus de liberdade na concepção do modelo (*front end* mais classificador) devendo este ser avaliado usando um conjunto de dados completo e uma metodologia rigorosa para evitar conclusões tendenciosas. Considerando a importância do uso de metodologias para classificação de faltas do tipo curto-circuito eficientes e principalmente com baixo custo computacional, este trabalho apresenta os resultados do estudo desenvolvido de análise do algoritmo KNN (K-vizinhos mais próximo) associado a medida de similaridade de Alinhamento Temporal Dinâmico (DTW) e do algoritmo HMM (Modelo Oculto de Markov) para a tarefa de classificação de faltas. Estas duas técnicas permitem o uso direto dos dados sem a necessidade de utilização de *front ends*, além de possuírem a capacidade de poder tratar séries temporais multivariadas e de tamanho variável, que é o caso das sequências de sinais para o caso pós-falta. Para desenvolvimento dos dois sistemas propostos para classificação foram utilizados dados simulados de faltas do tipo curto-circuito oriundos da base de dados pública UFPAFaults. Para comparação de resultados com metodologias já apresentadas na literatura para o problema, foi também avaliada, para o mesmo banco de dados, a arquitetura FBSC. No caso da arquitetura FBSC, diferentes *front ends* e classificadores foram utilizados. A avaliação comparativa foi realizada a partir da medida de taxa de erro, custo computacional e testes estatísticos. Os resultados obtidos mostraram que o classificador baseado no HMM se mostrou mais adequado para o problema de classificação de faltas do tipo curto-circuito em linhas de transmissão.

**Palavras-chave:** Qualidade de energia elétrica, sistemas elétricos de potência, curto-circuito, classificação de faltas, KNN-DTW, HMM.

## ABSTRACT

Maintaining power quality in electrical power systems depends on addressing the major disturbances that may arise in their generation, transmission and distribution. Within this context, many studies have been developed aiming to detect and classify short circuit faults in electrical systems through the analysis of the electrical signal behavior. Transmission line fault classification systems can be divided into two types: online and post fault classification systems. In the post-missing scenario the signal sequences to be evaluated for classification have variable length (duration). In sequence classification it is possible to use conventional classifiers such as Artificial Neural Networks, Support Vector Machine, K-nearest neighbors and Random forest. In these cases, the classification process usually requires a sequence preprocessing or a front end stage that converts the raw data into sensitive parameters to feed the classifier, which may increase the computational cost of the classification system. An alternative to this problem is the FBSC-FrameBased-Sequence Classification (FBSC) architecture. The problem with FBSC architecture is that it has many degrees of freedom in designing the model (front end plus classifier) and it should be evaluated using a complete dataset and rigorous methodology to avoid biased conclusions. Considering the importance of using efficient short-circuit fault classification methodologies and mainly with low computational cost, this paper presents the results of the KNN-DTW (K-Nearest Neighbor) algorithm analysis study associated with Dynamic similarity measurement. Time Warping (DTW) and HMM (Hidden Markov Model) algorithm for fault classification task. These two techniques allow the direct use of data without the need for front ends for signal pre-processing, as well as being able to handle multivariate and variable time series, such as signal sequences for the post-miss case. To develop the two proposed systems for classification, simulated data of short-circuit faults from the UFPAFaults public database were used. To compare results with methodologies already presented in the literature for the problem, the FBSC architecture was also evaluated for the same database. In the case of FBSC architecture, different front ends and classifiers were used. The comparative assessment was performed from the measurement of error rate, computational cost and statistical tests. The results showed that the HMM-based classifier was more suitable for the problem of classification of short circuits on transmission lines.

**Keywords:** Electrical power quality, electrical power system, short circuit, fault classification, KNN-DTW, HMM.

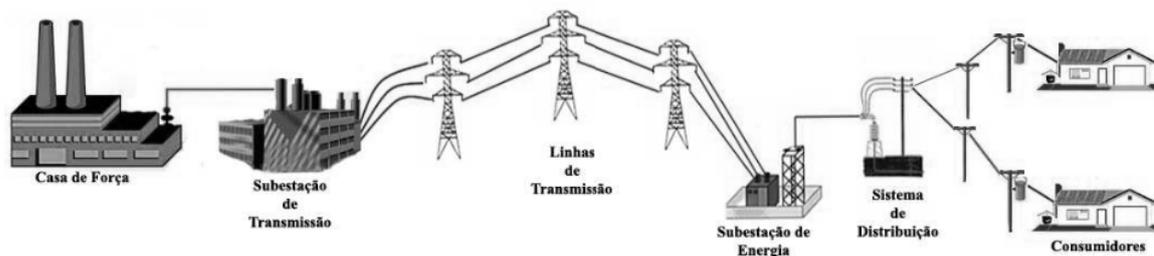
# 1 INTRODUÇÃO

## 1.1 Motivação e descrição geral do problema

A demanda crescente por energia elétrica e o aumento do consumo de equipamentos eletroeletrônicos mais vulneráveis aos distúrbios elétricos acarretam uma maior necessidade por qualidade de energia elétrica (QEE<sup>1</sup>). Essa realidade impulsiona um sistema elétrico de potência (SEP) a possuir uma configuração aceitável de infraestrutura física, operacional e de controle que possam evitar ou diminuir os distúrbios elétricos (YADAV; DASH, 2014).

Um SEP típico, conforme ilustra a Figura 1, é geralmente dividido em três zonas funcionais: geração, transmissão e distribuição (MORAIS et al., 2010a). Tais zonas funcionais estão sujeitas a ocorrência de distúrbios naturais ou provocados pela ação do homem. Como consequência, as formas de onda de tensão ou corrente sofrem certas alterações em seu sinal, desviando de seus valores nominais, caracterizando os chamados eventos de QEE.

**Figura 1 – Exemplo de um sistema elétrico de potência.**



**Fonte: (MORAIS et al., 2010a).**

A linha de transmissão é o componente do SEP mais vulnerável as falhas, especialmente se considerar suas dimensões físicas, complexidade estrutural e o ambiente em que se encontra. Dentre as falhas que podem ocorrer em uma linha de transmissão, as faltas do tipo curto-circuito são as que provocam maiores impactos para o consumidor. Estudos revelaram que essas faltas são responsáveis por cerca de 70% dos distúrbios ocorridos nos sistemas elétricos (ZHANG; KEZUNOVIC, 2007). Neste sentido, fica evidente a necessidade dos SEPs adotarem mecanismos para detecção e classificação desse tipo de falta, auxiliando assim na tomada de decisão a nível operacional, visando a manutenção e o restabelecimento do funcionamento do SEP.

Nos últimos anos têm se observado que o setor elétrico brasileiro vem sofrendo um impacto mais acentuado para conseguir manter a QEE de tal forma que possa atender de forma aceitável e efetiva o aumento da demanda de seus consumidores. Além do mais, o Sistema Elétrico Interligado (SIN) brasileiro enfrenta grandes desafios de investimentos em termoelétricas,

<sup>1</sup> QEE, segundo (BOLLEN et al., 2009), é qualquer distúrbio ou alteração nos valores de tensão, corrente ou desvio da frequência que resulte em falta ou má operação dos equipamentos dos consumidores.

aumento da presença de pequenos produtores independentes, elevação na utilização de energia eólica, entre outros.

Desse modo, a reestruturação do setor elétrico, para garantir a estabilidade no fornecimento de energia, induziu as empresas distribuidoras de energia a buscarem a adoção de novos padrões de desempenho e, reverem suas metodologias de avaliação e atuação no sistema, afim de atender as novas exigências e para evitar sanções de penalidades da Agência Nacional de Energia Elétrica (ANEEL) que é o órgão regulador e fiscalizador que define padrões de QEE do setor elétrico.

Diante deste novo paradigma e procurando soluções para essas questões, as empresas do sistema elétrico brasileiro estão buscando utilizar tecnologias avançadas para aquisição e armazenamento de informações, as quais incluem os chamados distúrbios ou eventos de interesse. Um típico exemplo dessa tecnologia são os equipamentos de oscilografia que implementam algoritmos relativamente simples que detectam se as formas de onda de tensão desviam dos seus valores de amplitude nominal. Se a variação é maior que um limiar, um circuito programável denominado de trigger inicia o armazenamento dos dados, junto com informações adicionais, tais como a data e a hora (BOLLEN et al., 2009).

No campo da análise de faltas, as companhias buscam mecanismos mais sofisticados para detecção de ocorrências elétricas em linhas de transmissão de energia (BAKKEN et al., 2011). Um exemplo desses mecanismos são os relés de proteção, cujo papel é interromper e diagnosticar em menor tempo possível a transmissão de energia diante de uma falta do tipo curto-circuito. Isto é uma alternativa de automatizar o controle, monitoramento e manutenção do sistema, auxiliando na tomada de decisão a nível operacional diante de uma ocorrência de distúrbio elétrico (FONTES, 2015).

Apesar do avanço da tecnologia estabelecer metodologias eficientes para classificação de faltas do tipo curto-circuito, uma das dificuldades ainda encontrada está no tratamento do grande volume de dados coletados pelos equipamentos de oscilografia, que apresentam taxas de amostragem de até 60 amostras por segundo. Outro problema é que os dados são frequentemente armazenados sem serem devidamente rotulados, o que dificulta a aplicação de algoritmos de Aprendizado de Máquina (WITTEN; FRANK, 2005).

Outro ponto importante a ser considerado é que gerar resultados confiáveis e relevantes na área de classificação de faltas não é uma tarefa fácil, pois a ausência de bases de dados de domínio público influencia diretamente nas pesquisas que avaliam algoritmos de mineração de dados. Por esse motivo as empresas do setor elétrico ainda não se beneficiam plenamente com o processamento automático da informação.

Para (MORAIS et al., 2010a) apesar da tecnologia avançada, gerar resultados confiáveis e relevantes na área de classificação de faltas não é uma tarefa fácil, pois a ausência de benchmarks influencia diretamente nas pesquisas que avaliam algoritmos de mineração de dados. Por esse motivo as empresas do setor elétrico, não se beneficiam plenamente com o processamento

automático da informação. Além disso, as bases de dados citadas na literatura são invariavelmente privadas, situação que estimulou a utilização de uma base de dados de domínio público e robusta denominada de UFPAFaults.

Em relação ao estudo e análise de faltas do tipo curto-circuito em linhas de transmissão, os sistemas de classificação podem ser divididos em dois tipos: sistemas de classificação *on-line* e pós-falta. Os sistemas de classificação *on-line* classificam o tipo de falta em um curto espaço de tempo, baseado no segmento de análise (quadro) em tempo real, com amostras do sinal no instante em que a falta ocorre. Já a classificação pós-falta é executada *off-line* e sua entrada consiste em uma série temporal multivariada com comprimento (duração) variável, diferenciando da classificação *on-line* em que a entrada para o sistema é um vetor de tamanho fixo. Neste caso, o processo de classificação requer um pré-processamento ou um estágio de *front end* que converta os dados brutos em parâmetros sensíveis para alimentar o *back-end* (neste caso, o classificador). Os sistemas *on-line* e pós-falta tentam resolver problemas que podem ser tratados como problemas de classificação convencional e de sequência, respectivamente (SUDHA, 2007; ZHONGMIN; BIN, 2000).

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Considerando a importância do problema de classificação de faltas em linhas de transmissão de energia, este trabalho tem como proposta principal a avaliação dos algoritmos KNN-DTW e HMM para o problema. Estas duas técnicas apresentam a vantagem de poderem utilizar como entrada os valores de sequência das faltas sem a necessidade do uso de front ends. Outra vantagem desses algoritmos diz respeito a possibilidade de se trabalhar com séries temporais multivariada e de tamanho variável, o que é propício para o problema de classificação de faltas. Para análise de desempenho dos sistemas foi adotada a base de dados UFPAFaults, sendo que esta base é devidamente rotulada e composta por sequências correspondentes que representam classes de faltas do tipo curto-circuito.

### 1.2.2 Objetivos Específicos

- Determinação dos atributos mais importantes da base de dados UFPAFaults para obter resultados satisfatórios e confiáveis para as arquiteturas classificadoras implementadas.
- Avaliação dos algoritmos KNN-DTW e HMM para classificação de faltas em linha de transmissão observando seu desempenho na taxa de erro e custo computacional utilizando a base de dados UFPAFaults.
- Replicação de resultados da arquitetura de classificação de sequência baseada em quadros FBSC apresentada em (HOMCI et al., 2016), que se baseia no uso de *front-ends* aliados a

classificadores convencionais, tais como rede neurais artificiais e outros, para classificação de faltas.

- Comparação dos resultados de desempenho obtidos dos algoritmos KNN-DTW e HMM com a arquitetura FBSC.
- Realização de testes de significância estatística como forma de provar a igualdade ou diferença nos resultados dos classificadores desenvolvidos no estudo.

### 1.3 Estado da arte

Propostas recentes de classificação de faltas do tipo curto-circuito em linhas de transmissão têm usado técnicas de processamento digital de sinais (janelamento, representação rms, transformada wavelet e outros) como ferramenta para extração das características do sinal, essas características são usadas como entrada para os sistemas de reconhecimento de padrões e de classificação baseados em inteligência computacional. Alguns trabalhos baseado nesses métodos são apresentados a seguir.

Em (MORAIS et al., 2010a) foi proposto um modelo de classificação de sequência baseada em quadros(FBSC). O simulador Alternative Transient Program(ATP) foi utilizado para a criação de uma base de dados de faltas do tipo curto-circuito chamada de UFPAFaults. São apresentados resultados aplicando-se diferentes técnicas de pré-processamento ou front ends (para extração de características), seleção de parâmetros e algoritmos de aprendizado. *Front ends* como *wavelets* (a wavelet mãe utilizada foi Daubechies 4 com  $\gamma = 3$  níveis de decomposição) são comparados com outros mais simples, baseados, por exemplo, nos valores RMS. Esses *front ends* são combinados com diversos algoritmos de Aprendizado de máquina (AM): RNA, SVM, KNN e J.48 (árvore de decisão). A escolha dos valores que definem o modelo do classificador é feita a partir de uma seleção automática de modelo. Os resultados apontam que em termos de *front ends*, a *waveletconcat* foi o que apresentou os melhores resultados para maioria dos classificadores convencionais. No que diz respeito à acurácia dos classificadores, os resultados indicam que SVM e redes neurais apresentam um melhor desempenho com relação aos outros tipos de classificadores. Quando o objetivo é alcançar um bom nível de generalização, o classificador SVM é o mais atrativo.

Em (MOHAMMADI; DEGHANI, 2015) é apresentada uma metodologia para seleção de parâmetros (atributos) e redução de dimensionalidade como um passo fundamental na garantia da segurança em grandes sistemas de potência, onde o número de parâmetros que representa o estado das redes de energia aumenta drasticamente. O trabalho afirma que uma grande quantidade de atributos não é adequada para serem usadas como entrada para algoritmos de inteligência computacional na classificação de faltas, já que podem provocar o aumento do custo computacional inviabilizando análises *on-line*. Assim, o artigo propõe um método para garantia da segurança do valor de tensão baseado em árvores de decisão. A principal característica do método proposto

é uso da Análise de Componentes Principais (PCA) para redução da dimensão das amostras das PMUs (*Phasor Measurement Unit*) selecionadas. Então, um método de seleção de parâmetros baseado em correlação é aplicado para selecionar os atributos mais significantes para o problema. A área de estudo são 39 subestações que compõe o sistema de potência do Irã. Os resultados obtidos mostraram que as árvores de decisão reduzem os dados em regras simples, melhorando a performance em termos de custo e economia na classificação de faltas.

Em (HOMCI et al., 2016) foi feito um estudo sobre as linhas de transmissão considerando que é o elemento mais suscetível a falhas nos sistemas elétricos de potência, e as faltas do tipo curto-circuito são consideradas as piores falhas que podem acontecer neste elemento. Para evitar mais problemas devido a essas falhas, o trabalho acredita que um diagnóstico de falha e o uso de *front ends* é necessário. No entanto, o processo de seleção para escolher os *front ends* não é simples, pois se comporta de maneira diferente para cada um deles. Portanto, propõe um novo *front end* chamado ConcatFrontEnd, que integra outros *front ends*, como wavelet, raw e RMS. Além disso, foi aplicado técnicas de seleção de recursos baseadas em filtro para diminuir a dimensão dos dados de entrada. Assim, foram utilizados os seguintes classificadores: RNA, KNN, RF e SVM sobre a base de dados UFPAFaults para treinar e testar os classificadores. Como resultado, a concatenação de *front ends*, na maioria dos casos, alcançou as menores taxas de erro. Além disso, as técnicas de seleção de parâmetros (*features*) aplicadas mostraram que é possível obter maior precisão usando menos recursos no processo.

Em (FATHABADI, 2016) é proposta uma nova abordagem para detectar, classificar e localizar faltas de tipo curto-circuito em linhas de transmissão de energia. A abordagem proposta consiste em uma estrutura híbrida composta por um filtro de resposta de impulso finito (*FIR - finite impulse response*) de dois estágios, quatro SVMs, e onze regressões vetoriais de suporte (SVRs). Nessa abordagem o filtro FIR de dois estágios proposto junto com as SVMs é usado para detectar e classificar faltas de tipo curto-circuito enquanto os SVRs são utilizados para localizar as mesmas em linhas de transmissão. A estrutura implementada precisa de poucas amostras de treinamento para treinar os SVMs e SVRs. Para uma linha de transmissão de energia com o comprimento de 50 km, apenas 6 amostras de treinamento foram necessárias para treinar cada SVR. O *framework* pode rapidamente executar os processos de detecção, classificação e localização de falhas somente durante 1 ciclo antes que a falta de energia seja realizada por relés de proteção, que é estritamente menor do que o tempo de compensação de faltas. Uma linha de transmissão de energia trifásica de 230 kV, 50 Hz com o comprimento de 50 km é simulada para validar os resultados teóricos e verificar a precisão da técnica proposta.

Em (RAMESH; MOHAN, 2017) é proposto uma metodologia para classificação de faltas em sistemas de energia usando a decomposição do modo empírico (EMD) e SVMs. O EMD é usado para a decomposição das tensões da linha de transmissão nas Funções do Modo Intrínseco (FMI). *Hilbert Huang Transform* (HHT) é usado para extrair recursos característicos dos FMI. Um modelo SVM múltiplo é introduzido para classificar a condição de falha entre dez faltas em sistemas de energia. O algoritmo é validado usando o ambiente MATLAB/SIMULINK.

Os resultados demonstram que a combinação de EMD e SVM pode ser um classificador eficiente com níveis aceitáveis de precisão.

Em (ALMEIDA et al., 2017) foi feita uma proposta de abordagem combinando análises de componentes independentes (ICA) com a teoria da ondas viajantes (TW) e SVM. A abordagem é adequada para localizar e reconhecer faltas em linhas de transmissão de energia em alta tensão (HV), enquanto os sinais adquiridos são ruidosos. Experimentos realizados para tipos distintos e locais de faltas em um modelo de linha de transmissão real mostraram que os métodos combinados propostos são capazes de fornecer excelente desempenho na localização de faltas. Os erros obtidos são inferiores a 1% e a precisão é de 100% para a classificação de sinais de falta com ruído. Pode-se afirmar que este método apresenta um melhor desempenho do que os relativos às principais técnicas convencionais, com o sinais de ondas e redes neurais, na presença de ruído.

No trabalho de (RAY D. P. MISHRA; MISHRA, 2017) é apresentado um estudo de análise sobre a detecção e classificação de falhas em uma longa linha de transmissão, que é compensada em série usando redes neurais artificiais e transformada *wavelet*. O esquema proposto faz uso de uma amostra de pré-falta de ciclo e uma de pós-falta de ciclo dos sinais de corrente de três fases para encontrar o sinal de corrente de terra. Daubechies é usada como *wavelet* mãe usando a técnica de transformada *wavelet* discreta. A energia diferencial, baseada em transformada *wavelet* discreta, é aplicada para alimentar um sistema, projetado para a classificação de todos os onze tipos de falhas. Finalmente, as características ótimas, que são as energias obtidas da transformada *wavelet* discreta dos sinais de corrente selecionados, são alimentadas às redes neurais para fins de classificação de falhas. A confiabilidade da técnica sugerida é experimentada em sistemas de energia de 735 kV, 50 Hz em configurações operacionais alteradas usando o MATLAB. Os resultados indicam que a técnica proposta pode classificar corretamente todas as faltas possíveis com grandes variações nas condições do sistema.

Em (BISWAS K. KUMAR; NAYAK, 2018) é apresentado uma técnica de detecção e classificação de faltas de forma rápida e confiável em linhas de transmissão de energia empregando a análise de frequência-tempo com base na transformada de *wavelet* discreta (DWT) nos sinais de corrente trifásicos medidos por *Thyristor Controlled Series Capacitor* (TCSC). A tarefa de detecção e classificação de faltas é realizada usando a energia espectral (SE). A aplicação do novo recurso utilizando energia espectral é a principal contribuição do trabalho proposto. O desempenho da técnica de detecção e classificação de faltas proposto é testado para uma ampla variedade de cenários de faltas em uma linha de transmissão de circuito duplo de 400 kV através do software EMTDC/PSCAD. Os resultados mostram claramente que, usando o método proposto, é possível realizar uma detecção e classificação rápidas e confiáveis de faltas em linhas de transmissão.

Em (SHARMA O. P. MAHELA; KUMAR, 2018) é apresentado um algoritmo para detecção e classificação de diferentes tipos de faltas em sistemas de energia que ocorrem em linha de transmissão para fornecer a proteção efetiva da linha contra essas falhas. A técnica

proposta utiliza a análise de alta resolução baseada em transformadas de Stockwell dos sinais de corrente para detecção de faltas e a árvore de decisão baseada em regras para classificação de faltas na rede do sistema de energia. As faltas investigadas neste estudo incluem falta linha terra (LG), falta linha dupla (LL), falta linha dupla terra (LLG) e falta trifásica com o envolvimento da terra (LLLG). O estudo proposto é realizado no ambiente MATLAB/Simulink. Os resultados indicam que a técnica tem um bom desempenho em relação a técnicas convencionais.

(CHATTERJEEA; DEBNATHB, 2019) apresenta um novo esquema de classificação de faltas em linhas transmissão baseado na correlação cruzada. O método proposto utiliza sinais de tensão de uma extremidade da linha. As características extraídas do correlograma cruzado dos sinais defeituosos são alimentadas ao classificador baseado em fuzzy para classificação de falhas. O esquema é imune à presença de ruído no ambiente em tempo real, pois a técnica de correlação cruzada minimiza o efeito do conteúdo aleatório não correlacionado no sinal. Além disso, a técnica de correlação é superior a outros métodos convencionais, pois requer baixa custo computacional. O resultado da simulação estabelece a capacidade potencial do esquema sob diferentes resistências de falha e ângulos de início de falha em diferentes locais de falha, e também sob diferentes condições de tensão no meio ciclo do início da falha. A comparação de desempenho com diferentes técnicas existentes revela a robustez e a aplicabilidade do esquema proposto em tempo real.

(FARSHAD, 2019) propõe um novo esquema de proteção para detecção e classificação de falhas internas em linhas de transmissão HVDC bipolares, usando o método de descrição de dados *K-means* (KMDD). No esquema de proteção proposto, são utilizados os sinais de tensão e corrente. Janelas de tempo relativamente curto são consideradas para esses sinais e os valores da soma das janelas de dados são calculados. No estágio de preparação, para cada tipo de falha, o método KMDD é aplicado a alguns dados pós-falta gerados em várias condições. Em seguida, os centróides e os limiares obtidos são usados para detectar e classificar novas faltas, mesmo em condições invisíveis. O desempenho do método proposto é avaliado em 4320 casos de faltas internas e 2816 externas em uma linha de transmissão HVDC aérea bipolar de 1000 km sob várias condições não vistas na etapa de preparação. Os resultados obtidos mostram que é rápido e preciso o suficiente para as faltas internas, e também é estável durante as faltas em condições normais de pré-falta.

Em (JU; ROSSO, 2019) é proposto um novo método para classificação de falhas em linhas de transmissão. A técnica utiliza energia média da wavelet (AWE) para quantificar componentes de alta frequência durante o período de falha para determinar o tipo de falha comparando valores de AWE para diferentes fases. O método proposto é testado com eventos de falha real na biblioteca de formas de onda desenvolvida pela GPA-Grid *Protection Alliance* em parceria com o *Electric Power Research Institute* (EPRI). Os resultados mostram que a metodologia proposta apresenta um desempenho melhor que o método tradicional implementado no openXDA (<https://github.com/GridProtectionAlliance/openXDA/wiki>) em termos da precisão da classificação do tipo de falta.

A Tabela 1 apresenta um resumo do cenário dos trabalhos relacionados, onde observa-se front ends e técnicas de classificação utilizadas em cada estudo.

## 1.4 Publicações realizadas

- Arouche Freire, J. C.; Garcez Castro, A. R.; Homci, M. S.; Meiguins, B.; Morais, J. M. Transmission line fault classification using Hidden Markov Models. IEEE Access, v. 7, p. 113499-113510, 2019.
- Costa, B. G.; Arouche Freire, J. C.; Cavalcante, H. S.; Homci, M.; Castro, A. R. G.;

**Tabela 1 – Mapeamento dos trabalhos relacionados.**

<b>Trabalho Relacionado</b>	<b>Front End</b>	<b>Classificação Convencional</b>	<b>Classificação Direta</b>	<b>Ano</b>
(MORAIS et al., 2010a)	Raw, RMS, Wavelet e Waveletenergy	RNA, SVM, J.48 e KNN	Não	2010
(MOHAMMADI; DEGHANI, 2015)	Análise de Componente Principal	Árvore de Decisão	Não	2015
(HOMCI et al., 2016)	Raw, RMS, Wavelet, Waveletenergy e ConcatFrontEnd	RNA, SVM, Randon Forest e KNN	Não	2016
(FATHABADI, 2016)	FIR e SVR	SVM	Não	2016
(RAMESH; MOHAN, 2017)	FMI, EMD e HHT	SVM	Não	2017
(ALMEIDA et al., 2017)	ICA e TW	SVM	Não	2017
(RAY D. P. MISHRA; MISHRA, 2017)	Wavelet	RNN	Não	2017
(COSTA et al., 2017)	Não	KNN-DTW	Sim	2017
(BISWAS K. KUMAR; NAYAK, 2018)	Wavelet	SE	Não	2018
(SHARMA O. P. MAHELA; KUMAR, 2018)	Transformada de Stockwell	Árvore de Decisão	Não	2018
(CHATTERJEE; DEBNATHB, 2019)	Correlação Cruzada	Fuzzy	Não	2019
(FARSHAD, 2019)	KMDD	Limiares e Centróides	Não	2019
(JU; ROSSO, 2019)	AWE	AWE	Não	2019
(AROUCHE J. C.; MORAIS, 2019)	Não	HMM	Sim	2019

Viegas, R.; Meiguins, B. S.; Morais, J. M. Fault Classification on Transmission Lines using KNN-DTW In: ICCSA - 17th International Conference on Computational Science and Its Applications, Trieste, 2017, Trieste.

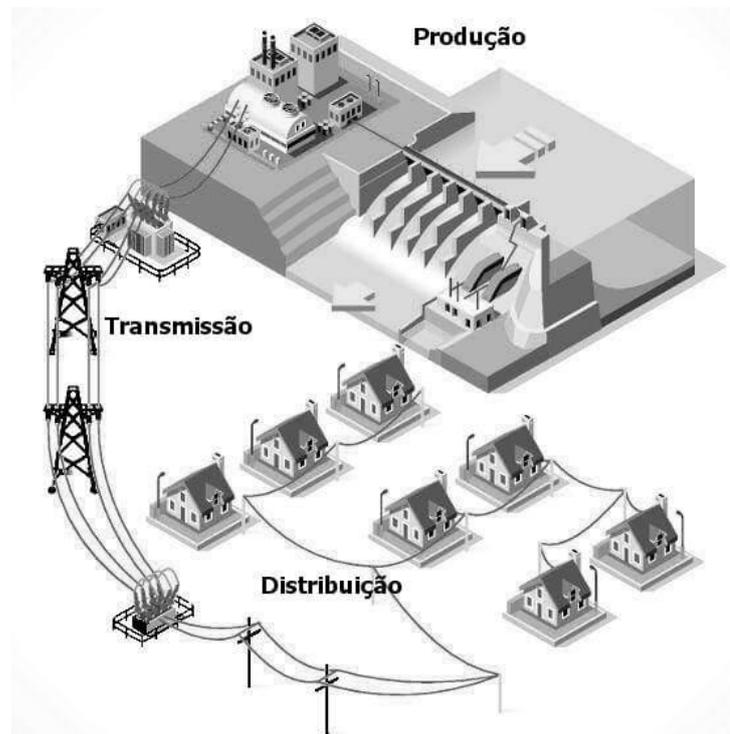
- Homci, M. S.; Chagas JR, P.; Miranda, B.; Arouche Freire, J. C.; Viegas JR, R.; Pires, Y.; Meiguins, B.; Morais, J. A New Strategy Based on Feature Selection for Fault Classification in Transmission Lines In: IBERAMIA, San Jose - Costa Rica. Sociedad Iberoamericana de Inteligencia Artificial, 2016.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Falhas em Linhas de Transmissão

Um Sistema Elétrico de Potência (SEP) é responsável por gerar e entregar a energia elétrica aos consumidores finais, sendo uma infraestrutura normalmente dividida em Geração, Transmissão e Distribuição (Figura 2).

Figura 2 – Exemplo de um sistema elétrico de potência.



Fonte: (SOLAR, 2019).

Considerando a infraestrutura da Transmissão, a linha de transmissão (LT) é considerada um dos componentes mais vulneráveis do SEP, sendo a componente com maior probabilidade de ocorrência de faltas, devido principalmente à sua grande susceptibilidade às descargas atmosféricas. As faltas ocasionadas pelos distúrbios na LT podem ocasionar a interrupção da transmissão de energia na mesma, sendo desta forma fundamental um sistema de proteção adequado que evite a quebra da cadeia de fornecimento de energia elétrica (OLESKOVICZ, 2001). Dentre as falhas que podem ocorrer em uma linha de transmissão, as faltas do tipo curto-circuito são as que provocam maiores impactos para o consumidor. O sistema de proteção a ser adotado pelo SEP deve ter a capacidade de o mais breve possível detectar a falta, procurando informar quais as fases da LT estão envolvidas, além de determinar a localização da falta. Essas informações são fundamentais para agilizar os reparos e desta forma reduzir o tempo de indisponibilidade do sistema (OLIVEIRA, 2005)).

Em um SEP, o sistema trifásico é o formato mais difundido no desenvolvimento de

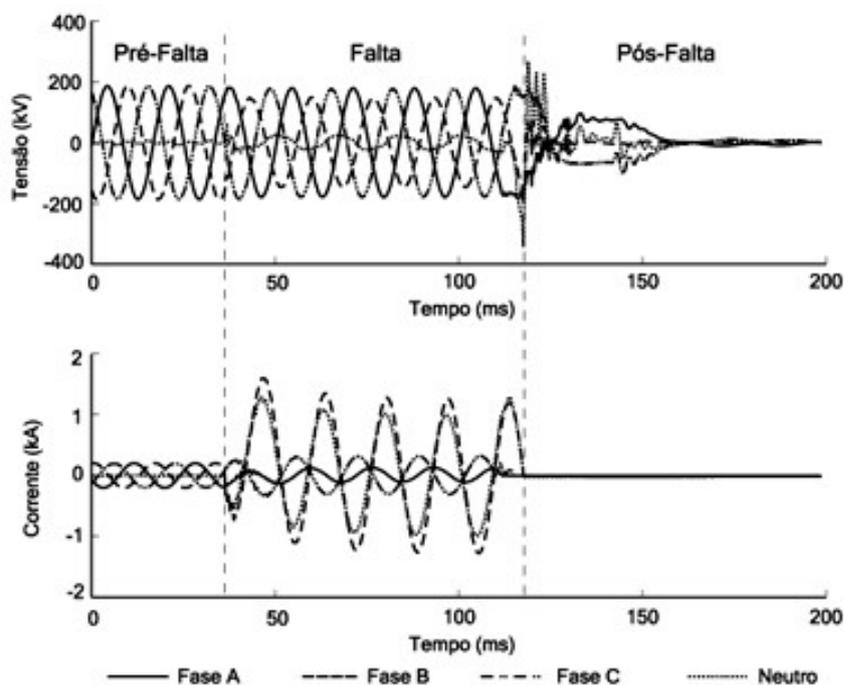
mecanismos de geração, transmissão e distribuição de energia elétrica (SAHA I., 2010). A Figura 3 apresenta uma linha LT trifásica. O sistema trifásico é composto por três tensões alternadas denotadas por A, B e C (fases da LT) e defasadas em  $120^\circ$  uma da outra. Considerando as fases do sistema trifásico, os curtos-circuitos em uma LT podem ser divididos em 3 tipos: o fase-terra, o fase-fase e fase-fase-terra e o trifásico. O curto-circuito fase-terra ocorre quando uma das fases da LT entra em contato com o solo, sendo que alguns fatores podem ocasionar essa situação, como fortes ventanias ou queda de uma árvore. Esse tipo de curto-circuito é o de maior ocorrência e equivale a aproximadamente 70% do total de faltas em uma LT (FAULKENBERRY; COFFER, 1996). Os curtos-circuitos fase-fase-terra podem acontecer normalmente pelo mesmo motivo que os curtos-circuitos fase-terra, tendo o agravante de envolver duas linhas de transmissão. Esse tipo de curto-circuito equivale a 10% dos casos (FAULKENBERRY; COFFER, 1996). O curto-circuito fase-fase ocorre quando dois condutores da LT, de fases diferentes, entram em contato. Normalmente este tipo de falta ocorre devido à fortes ventanias, que aproximam os condutores, ou pelo rompimento e queda de um condutor sobre o outro. Esse tipo de falta equivale a 15% dos casos de curto-circuito em LT (FAULKENBERRY; COFFER, 1996). Os curtos-circuitos trifásico acontecem raramente e ocorrem quando as três fases entram em contato. Este tipo de curto-circuito pode ocorrer quando algum objeto cai sobre a linha, quando algum equipamento, como um transformador, falha ou quando as três fases vão ao solo. Representam 5% das faltas em linhas de transmissão (FAULKENBERRY; COFFER, 1996).

**Figura 3 – Linha de Transmissão Trifásica em um SEP.**



**Fonte: (CARIRI, 2019).**

Os SEPs possuem equipamentos apropriados para realizar a aquisição e armazenamento de dados referentes aos eventos do tipo curto-circuito. Estes eventos geralmente são caracterizados por alterações nos valores de tensão e corrente ou qualquer desvio de frequência do sinais, sendo possível então, a partir de tais dados, extrair informações que possam ser úteis para uma análise da falta e classificação da mesma. A Figura 4 apresenta um exemplo com sinais medidos nas 3 fases no momento que uma falta ocorre.

**Figura 4 – Sinais de medição fasorial em três fases na ocorrência de uma falta.**

Fonte: (SILVA K. M.; COSTA F. B., 2007).

As formas de ondas correspondentes aos eventos de faltas estão na forma de séries temporais. Uma série temporal é caracterizada por uma sequência de observações ao longo do tempo à respeito de um determinado fenômeno. A relação de dependência com o tempo difere a série temporal de outros tipos de dados estatísticos.

As séries temporais podem ser classificadas quanto ao número de variáveis no sistema como sequências univariadas ou multivariadas. As séries temporais univariadas são séries obtidas a partir da amostragem de um único padrão de observação, por exemplo, os valores de uma única variável física ou de um sinal dependente do tempo. Séries temporais multivariadas são geradas a partir da observação simultânea de duas ou mais variáveis ao longo do tempo. Em um SEP, os dados coletados automaticamente por sistemas de monitoramento são dados de séries temporais multivariadas.

## 2.2 Classificação de Faltas em Linhas de Transmissão

No desenvolvimento de um sistema de classificação de faltas busca-se uma função que permita associar as séries temporais características de cada tipo de falta a um rótulo categórico denominado de classe, sendo que a classificação de séries temporais de faltas em uma LT corresponde geralmente a um problema de classificação de sequências, uma tarefa de classificação especial onde os dados de entrada do sistema classificador são representados por uma matriz que possui tamanho variável. O tamanho variável da possível matriz de entrada se deve ao fato do tempo de ocorrência de uma específica falta, que irá gerar uma matriz composta com mais

ou menos amostras do sinal (a partir da série temporal multivariada), dependendo do tempo de duração da falta na LT.

Considerando os sistemas de classificação de sequências, especificamente para séries temporais multivariadas de tamanho variável, em (XING; PEI; KEOGH, 2010) o autor divide os métodos relacionados a essa particularidade em três grandes categorias:

- A primeira categoria é a classificação baseada em características ou parâmetros, que transforma a sequência variável em um vetor de características e, em seguida, aplica métodos de classificação convencionais (que trabalham com vetores de entrada fixo). A seleção de características desempenha um papel importante neste tipo de método de classificação.
- A segunda categoria é a classificação de sequência baseada na distância. Uma função de distância é usada para medir a semelhança entre a sequência atual a ser classificada e uma sequência padrão da classe. A escolha da função distância determina significativamente a qualidade da classificação.
- A terceira categoria é a classificação de sequência baseada em modelo como os Modelos Ocultos de Markov (HMM - *Hidden Markov Model*) e outros modelos estatísticos para classificar sequência.

Os sistemas de classificação de faltas em uma linha de transmissão, podem ser de dois tipos: sistemas de classificação *on-line* e sistemas de classificação pós-falta ou *off-line*.

Os sistemas de classificação *on-line* lidam com um curto segmento do sinal (quadro) para realizar a classificação da falta e geralmente estes sistemas adotam a classificação quadro-a-quadro, sendo um quadro um vetor de tamanho fixo. Para estes casos os sistemas são utilizados a nível de proteção de relés, onde a tomada de decisão deve ser automática e em um curto espaço de tempo. Este tempo é muitas vezes baseado em um quadro correspondente a metade de um ciclo ou um ciclo com sinal de onda senoidal de 50 ou 60 Hz como é visto em (COSTA et al., 2017). Assim, os sistemas de classificação *on-line* tentam resolver problemas que podem ser tratados como problemas de classificação convencional (que lidam com vetores de entrada de tamanho fixo) onde uma decisão deve ser tomada para cada 33 amostras de sinal, por exemplo.

Na classificação pós-falta (*off-line*), a quantidade de informação disponível é maior, e o sistema pode trabalhar diretamente com as séries temporais de dimensão variável representativas de uma falta. Os sistemas de classificação pós-falta buscam resolver problemas que podem ser tratados como problemas de classificação de sequências (que lidam com vetores de entrada de tamanho variável), mas também podem ser tratados como problemas de classificação convencional a partir do uso do janelamento na sequência de tamanho variável. Outra alternativa é o uso da Classificação de Sequências Baseada em Quadro (FBSC - *frame-based sequence classification*) discutida em (MORAIS et al., 2010a), que também usa classificadores convencionais na classificação da falta.

(COSTA et al., 2017) coloca que na classificação de faltas em linhas de transmissão, representada por uma sequência, geralmente encontrada em um sistema elétrico trifásico, pode-se considerar que cada falta é uma série temporal multivariada de duração variável. A  $n$ -ésima falta  $X_n$  pode ser representada por uma matriz  $Q \times T_n$ . A coluna  $x_t$  de  $X_n, t = 1, \dots, T_n$ , é uma amostra multidimensional representada por um vetor de  $Q$  elementos, onde  $Q$  é o número de sinais e  $T_n$  é o número de amostras da falta. Este trabalho adota  $Q = 6$  (formas de onda de tensão e corrente das fases A, B e C) nos experimentos. Como o número de amostras multivariadas depende de  $n$ , um classificador convencional não é viável, sendo este o motivo principal para se utilizar KNN-DTW e HMM que classificam diretamente os sinais elétricos sem necessidade de etapas de pré-processamento (*Front ends*).

Na próxima secção será apresentada uma breve fundamentação teórica sobre as técnicas usadas para desenvolvimento dos classificadores convencionais e os classificadores baseado em sequências variáveis desenvolvidos neste trabalho para classificação de faltas em LT. Será também apresentada a Classificação de Sequências baseada em Quadro proposto em (HOMCI et al., 2016), pois a mesma será também implementada no trabalho para comparação com os resultados da proposta desta tese.

## 2.3 Técnicas para desenvolvimento de Classificadores Convencionais

### 2.3.1 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) são sistemas computacionais distribuídos inspirados nas funções cerebrais biológicas e compostas de unidades de processamento simples que computam funções matemáticas, sendo estruturas densamente interconectadas. As unidades de processamento são conhecidas como neurônios artificiais e ficam dispostas em uma ou mais camadas interligadas por um grande número de conexões que possuem pesos associados que regulam a entrada recebida por cada neurônio na rede para posteriormente produzir a sua saída (KEZUNOVIC; RIKALO, 1996). A Figura 5 exibe a estrutura de um neurônio artificial.

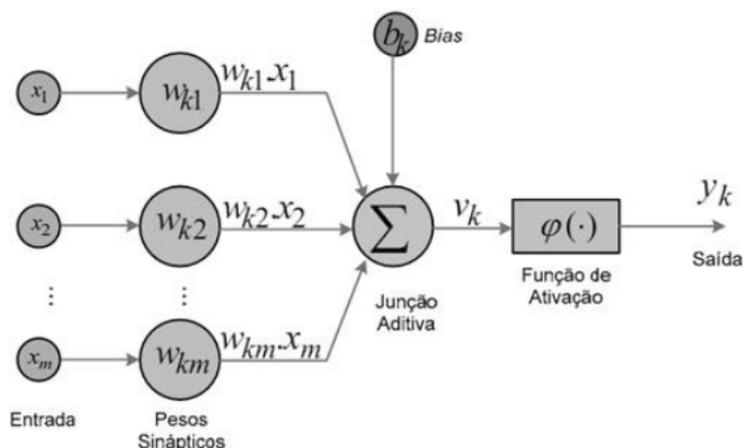
A junção somatória ou aditiva realiza a soma ponderada das entradas de acordo com:

$$V_k = \sum_{i=1}^m W_{ki}X_i + B_k \quad (2.1)$$

Onde  $x_i$  é a  $i$ -ésima entrada do neurônio  $k$ ,  $W_{ki}$  é o peso que liga a entrada  $i$  ao neurônio  $k$  e  $B_k$  é o bias aplicado externamente, que possui o efeito de acrescentar ou reduzir o sinal da junção somatória.

Constituído pela função de ativação  $\varphi(\cdot)$  o ativador, recebe o sinal de  $V_k$  e realiza o cálculo de nível de estímulo interno do neurônio. De acordo com este nível, a saída poderá ser ativada ou não. Na saída do neurônio a função de ativação realiza a normalização da amplitude

Figura 5 – Estrutura de um neurônio artificial.



Fonte: (HAYKIN et al., 2009).

no intervalo de  $[0,1]$ , também podendo ser mensurada no intervalo  $[-1,1]$ . Dessa maneira, a saída do neurônio é calculada por:

$$Y_k = \varphi(V_k) \quad (2.2)$$

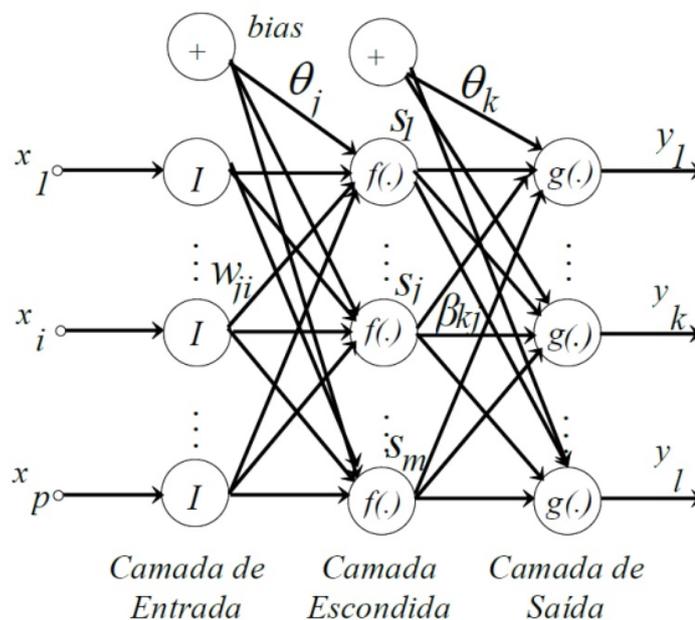
As redes neurais apresentam a capacidade de aprender através de exemplos em um processo denominado de aprendizado da rede neural. O processo de aprendizado é uma metodologia pelo qual os parâmetros variáveis da rede (pesos sinápticos) são ajustados, com o objetivo de minimizar uma função baseada no erro entre a saída da rede e a saída desejada. Existem muitos tipos de algoritmos de aprendizado para redes neurais artificiais. Estes diferem entre si principalmente pelo modo como os pesos são modificados. Os métodos de aprendizado podem ser do tipo supervisionado e não-supervisionado. No aprendizado supervisionado a rede neural recebe um conjunto de entradas e saídas de dados. Neste tipo de algoritmo, o aprendizado ocorre por meio dos ajustes nos pesos, os quais são modificados até que os erros entre os padrões de saída gerados pela rede tenham um valor desejado ou próximo do desejado. Já no aprendizado não supervisionado, a rede neural trabalha os dados de forma a determinar algumas propriedades do conjunto de dados. A partir destas propriedades é que o aprendizado é constituído.

Denomina-se iteração ou época uma apresentação de todos os pares (entrada e saída) do conjunto de treinamento no processo de aprendizado. A correção dos pesos numa iteração pode ser executada de modo *standard* ou em *batch*. No modo *standard* o erro é estimado a cada apresentação de um conjunto de treino à rede. Enquanto que no modo *batch* estima-se o erro médio após todos os exemplos do conjunto de treinamento serem apresentados à rede.

Dentre as arquiteturas de uma rede neural tem-se o Perceptron de múltiplas camadas (MLP). Uma MLP possui, além das camadas de entrada e saída, a disponibilização de neurônios organizados em uma ou mais camadas ocultas, denominadas camadas intermediárias (HAYKIN et al., 2009). Cada camada intermediária possui um número finito de neurônios. Cada neurônio de

uma camada está conectado a todos os neurônios da camada seguinte através de pesos sinápticos (Figura 6).

**Figura 6 – Rede Neural Perceptron Multicamadas. Fonte: (HAYKIN et al., 2009)**



**Fonte: (HAYKIN et al., 2009).**

As MLPs possuem a capacidade de aprender a partir de exemplos, sendo o aprendizado, do tipo supervisionado, realizado através de um processo iterativo de ajustes aplicados aos pesos sinápticos. Para que se possa realizar o processo de aprendizagem, é preciso primeiramente se ter um modelo do ambiente no qual a MLP será inserida. O principal objetivo do aprendizado nas MLPs é a obtenção de modelos com boa capacidade de generalização. A generalização é a capacidade da rede responder adequadamente para situações que não foram apresentadas à mesma durante a etapa de aprendizado. A rede MLP é geralmente treinada através do uso do algoritmo Backpropagation (BRAGA; CARVALHO; LUDERMIR, 2000; HAYKIN et al., 2009).

### 2.3.2 Máquinas de Vetores de Suporte

As Máquinas de Vetores de Suporte (*SVM - Support Vector Machine*) constituem uma técnica fundamentada na teoria de aprendizado estatístico. O objetivo de um classificador baseado em SVM consiste em encontrar um hiperplano (superfície de decisão) que maximize a separação no espaço de classes (VAPNIK, 2013; HAYKIN et al., 2009; FACELI, 2011).

Um ponto importante no algoritmo de aprendizagem por SVM é a função *kernel* que é usada para mapear os dados no espaço amostral, sendo responsável pela busca do hiperplano. Na literatura, várias possibilidades de *kernel* SVM são apresentadas em aplicações envolvendo reconhecimento de padrões, tais como: linear, polinomial, sigmóide e funções de base radial

(*RBF - Radial Basis Function*). As SVMs, e outros métodos kernel, podem ser caracterizados como uma função de estimação  $\phi$  definida por:

$$\frac{1}{V} \sum_{v=1}^V (\phi(x_v), x_v) + \lambda \|\phi\|_{H\kappa}^2 \quad (2.3)$$

Onde:

- $H\kappa$  corresponde ao espaço euclidiano gerado pelo kernel  $\kappa$ ;
- $\phi = h + b$ , onde  $h$  corresponde ao produto do vetor peso ( $\omega$ ) pelo vetor de suporte ( $x_v$ ) com  $h \in H\kappa$ ;
- $b$  corresponde ao bias,  $b \in R$ ;
- $L(\phi(x_v), y_v)$  corresponde a função perda (risco fundamental);
- $\lambda$  corresponde a autovalores;
- $V$  corresponde ao número de exemplos de treino.

### 2.3.3 K-Vizinhos mais próximos

Os classificadores de uma forma geral são caracterizados pelo fato de utilizarem os dados de treinamento para construir um modelo de classificação, o qual, uma vez encontrado e testado, estará pronto para testar qualquer padrão novo. Diferentemente desses classificadores, o classificador K-vizinhos mais próximos (*KNN - K-Nearest Neighbors*) utiliza os próprios dados de treinamento como modelo de classificação, isto é, para cada novo padrão que se quer classificar, utiliza-se os dados do treinamento para verificar quais são os exemplos nessa base de dados que são mais próximos do padrão em análise. A cada novo padrão a ser classificado faz-se uma varredura nos dados de treinamento, o que provoca um grande esforço computacional.

Considerando um conjunto de treinamento e seja  $z = (z_1, \dots, z_k)$  uma nova amostra, ainda não classificada, e a fim de classificá-la, calcula-se as distâncias, através de uma medida de similaridade, entre  $z$  e todos os exemplos do conjunto de treinamento e considera-se os  $K$  exemplos mais próximos (com menores distâncias) em relação a  $z$ . Verifica-se então qual a classe que aparece com mais frequência entre os  $K$  vizinhos encontrados. O padrão  $z$  será classificado de acordo com a classe  $y$  mais frequente dentre os  $K$  exemplos encontrados.

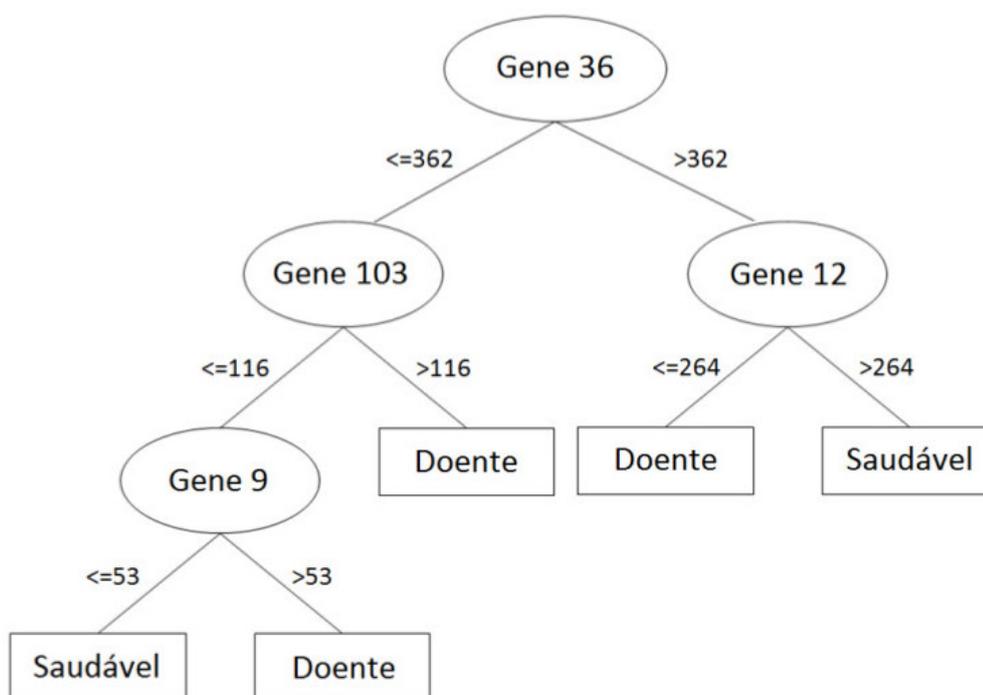
A distância entre duas amostras é calculada utilizando-se uma medida de similaridade. Uma medida de similaridade bastante popular é a distância euclidiana dada por:

$$d_{eucl} = \sqrt{\sum_{i=1}^K (z_i, \hat{z}_i)^2} \quad (2.4)$$

### 2.3.4 *Random forest*

O classificador Random Forest é uma árvore de decisão que utiliza mecanismos de dividir para conquistar para resolver problemas de decisão. A ideia se baseia em que problemas mais complexos podem ser divididos em problemas mais simples em forma recursivas. Essas soluções podem ser combinadas em forma de árvore para gerar uma solução do problema complexo. A proposta divide o espaço de amostras em subespaços que são ajustados em diferentes modelos, sendo formalmente estruturado em um grafo acíclico em cada nó, ou em um nó de divisão com dois ou mais sucessores dotado de um teste condicional, ou em um nó folha rotulado com uma nova classe (ZHOU et al., 2014). A Figura 7 apresenta um exemplo de árvore de decisão com seus nós de decisão e nós-folhas associados as classes desejadas pelo modelo.

**Figura 7 – Exemplo de estrutura de uma árvore de decisão.**



Fonte: Adaptado de (OSHIRO, 2013).

Observa-se na Figura 7 que os nós na árvore de decisão estão representando os genes numerados, e para cada decisão é associado uma escala contendo um valor referencial que determinará a classe saudável ou doente de pacientes em estudo.

Nessa direção, o classificador RF constrói muitas árvores de decisão que serão usadas para classificar um novo exemplo por um mecanismo de voto majoritário. Cada árvore de decisão usa um subconjunto de atributos selecionados aleatoriamente a partir do conjunto original, contendo todos os atributos (OSHIRO, 2013).

## 2.4 Técnicas para desenvolvimento de Classificadores baseados em Sequências(variável)

### 2.4.1 K-vizinhos mais próximos com Alinhamento Temporal Dinâmico

A técnica usando K-vizinhos mais próximos com Alinhamento Temporal Dinâmico (KNN-DTW) utiliza o Alinhamento Temporal Dinâmico, que é uma medida de similaridade baseada no conceito de programação dinâmica que permite comparar sequências de durações variáveis. É um algoritmo que se popularizou em aplicações de reconhecimento de voz, sendo atualmente bastante utilizado em áreas que lidam com séries temporais (GIORGINO, 2009). Para se entender o conceito de DTW é importante um entendimento da definição de programação dinâmica.

A programação dinâmica é um método de programação aplicável em situações nas quais não é fácil chegar a uma sequência ótima de decisões sem testar todas as sequências possíveis para então escolher a melhor. A ideia básica da programação dinâmica é construir por etapas uma resposta ótima combinando respostas já obtidas em partes menores (YURTMAN; BARSHAN, 2013).

Nesse sentido, o alinhamento, usando DTW, entre duas séries temporais  $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_p)$  e  $\hat{\mathbf{Z}} = (\hat{\mathbf{z}}_1, \dots, \hat{\mathbf{z}}_q)$  consiste em gerar uma matriz  $\mathbf{D}$  de dimensão  $p \times q$  cujo elemento  $(i, j)$  contém a distância Euclidiana  $d$  (outras medidas de similaridade podem ser adotadas) entre os vetores  $(\mathbf{z}_i, \hat{\mathbf{z}}_j)$ . Um percurso de ajuste  $W$ , é um conjunto de elementos contínuos da matriz que define um mapeamento entre os vetores de  $\mathbf{Z}$  e  $\hat{\mathbf{Z}}$ . O  $k$ -ésimo elemento de  $W$  é definido como  $w_k = (i, j)_k$ . Portanto:

$$W = w_1, w_2, \dots, w_{|W|}, \max(q, p) \leq |W| < p + q - 1 \quad (2.5)$$

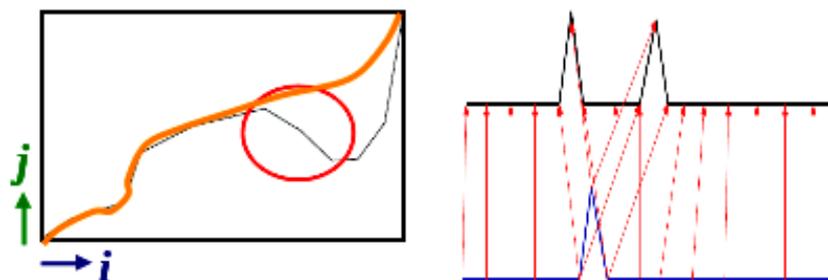
onde  $|W|$  é a cardinalidade (número de elementos) de  $W$ . O elemento  $\mathbf{D}(w_k)$  corresponde à distância euclidiana respectiva ao elemento  $w_k$  da matriz.

O DTW tem objetivo de encontrar o menor caminho entre duas sequências, sendo que conhecer todos os caminhos não é a solução ótima, visto que a quantidade de caminhos possíveis na matriz formada pelas sequências  $\mathbf{Z}$  e  $\hat{\mathbf{Z}}$  é de ordem exponencial, e por esse motivo é necessário reduzir o espaço de busca utilizando algumas estratégias. As estratégias adotadas são: monotocidade, continuidade, condição de contorno, alinhamento de janela e restrições de inclinação.

A monotocidade garante que não se repita o mesmo ponto, pois permanece ou incrementa os valores de  $i$  e  $j$  onde,  $i_{s_1} \leq i_s$  e  $i_{s_1} \leq j_s$ , logo não retorna a mesma posição da matriz.

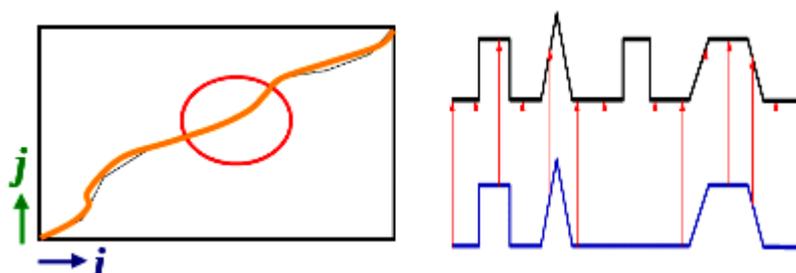
Na continuidade não há saltos na sequência do caminho, a condição  $i_{s_i} - i_{s_i-1} \leq 1$  e  $j_s - j_{s-1} \leq 1$ , garante que o alinhamento não omita uma característica importante.

Figura 8 – Monotocidade.



Fonte: adaptado de (BIOLOGY, 2019).

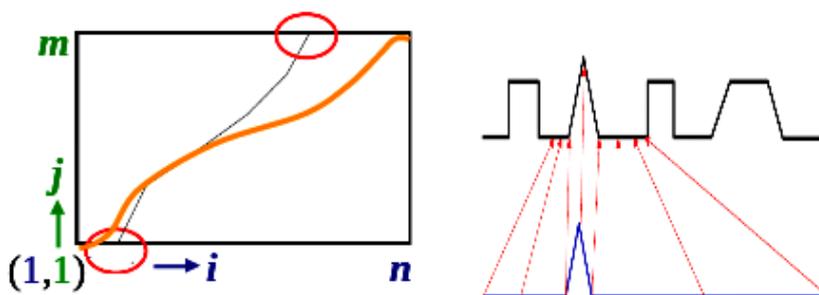
Figura 9 – Continuidade.



Fonte: adaptado de (BIOLOGY, 2019).

Na condição de contorno o caminho começa no canto inferior esquerdo e termina no canto superior direito, a condição  $i_1 = 1, i_k = n$  e  $j_1 = 1, j_k = m$ , garante que o alinhamento não considere uma sequência parcial.

Figura 10 – Condição de contorno.

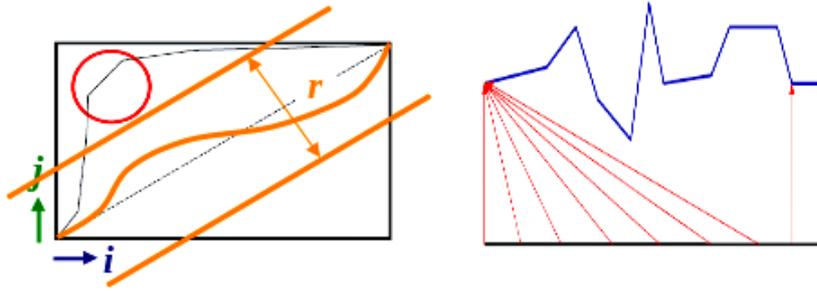


Fonte: adaptado de (BIOLOGY, 2019).

O alinhamento de janela limita o caminho evitando que exceda a largura da janela, garantindo o alinhamento próximo a diagonal. A condição  $|i_s - j_s| \leq r$ , onde  $r > 0$  é o tamanho da janela, garante que o caminho não permita características diferentes e nem estrutura similar.

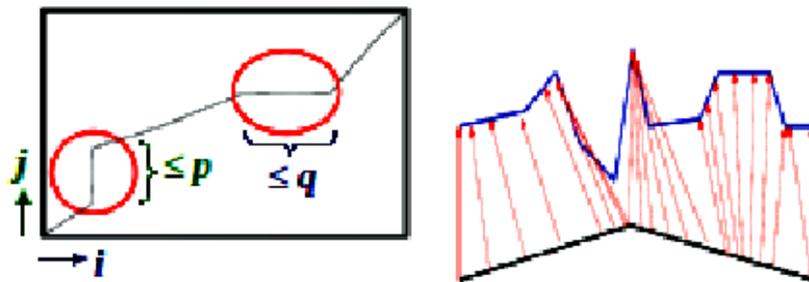
A restrição de inclinação evita que o caminho seja muito inclinado ou muito plano, logo impede que passos curtos coincidam com os longos. A condição é expressa por uma razão de  $S = p/q \in [0, \infty]$  em que  $p$  é o número de passos desejados na mesma direção (vertical ou horizontal) e  $q$  é o tempo na direção diagonal (SILVA; BATISTA, 2016).

Figura 11 – Alinhamento de janela.



Fonte: adaptado de (BIOLOGY, 2019).

Figura 12 – Restrição de inclinação.



Fonte: adaptado de (BIOLOGY, 2019).

A Figura 13 apresenta uma representação clássica do resultado final do algoritmo DTW e exibe as sequências  $\mathbf{Z}$  e  $\hat{\mathbf{Z}}$ , a matriz de alinhamento, as restrições e a sequência de interesse encontrada.

Dos muitos percursos de ajuste que satisfazem às condições acima, o percurso escolhido é aquele que minimiza o custo da deformação:

$$DTW(\mathbf{Z}, \hat{\mathbf{Z}}) = \min_W \sum_{k=1}^{|\mathbf{W}|} \frac{\mathbf{D}(w_k)}{|\mathbf{W}|} \quad (2.6)$$

onde  $|\mathbf{W}|$  é usado para compensar o fato de que o percurso de ajuste tenha diferentes tamanhos.

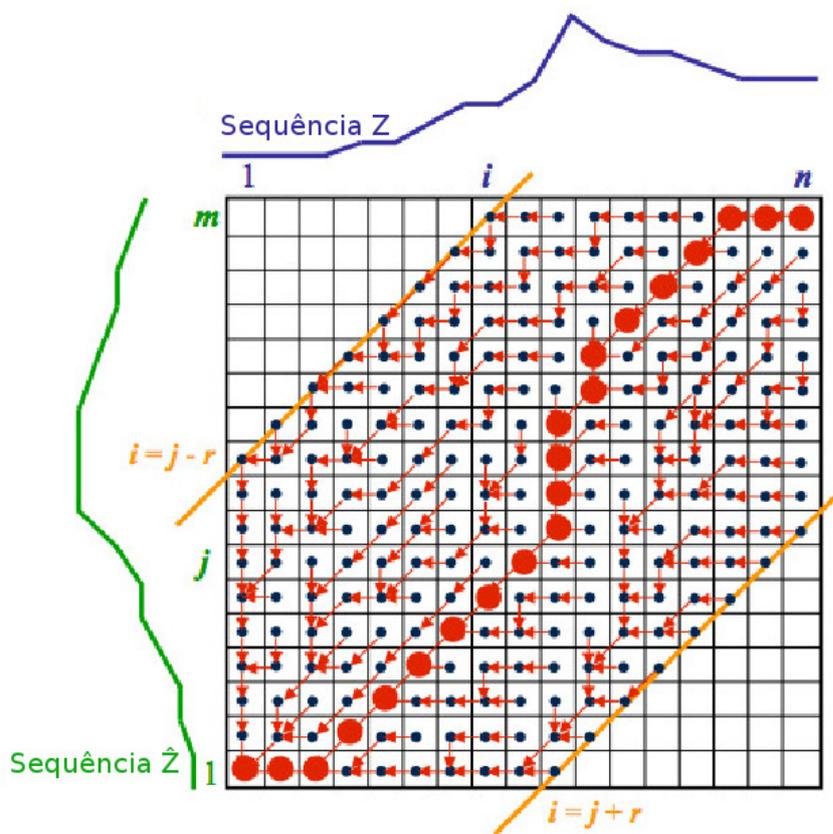
Este percurso pode ser encontrado de modo eficiente usando a programação dinâmica para avaliar a recursão sucessiva que define a distância acumulativa  $\gamma(i, j)$ . Um possível exemplo de cálculo de  $\gamma(i, j)$ , o qual foi usado nesse trabalho, é (vide (GIORGINO, 2009) para outras opções):

$$\gamma(i, j) = \mathbf{D}(i, j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\}. \quad (2.7)$$

Nota-se que tal cálculo é baseado na distância  $\mathbf{D}(i, j)$  encontrada na célula corrente e o mínimo das distâncias acumuladas dos elementos adjacentes.

Como dito anteriormente, o DTW foi usado como medida de similaridade do algoritmo

Figura 13 – Sequência encontrada após o cálculo DTW.



Fonte: adaptado de (BIOLOGY, 2019).

KNN aplicado à classificação de sequências, visto que utilizar a distância euclidiana como medida de similaridade, na classificação de sequências, não é possível em muitos casos ou mesmo o mais recomendável. Neste caso, o classificador KNN para o módulo de classificação de sequências é chamado de KNN-DTW (KEOGH; RATANAMAHATANA, 2005; XI et al., 2006).

#### 2.4.2 Modelo Oculto de Markov

O Modelo Oculto de Markov (*HMM - Hidden Markov Model*) foi desenvolvido no fim dos anos 60, porém apenas nos últimos anos o interesse no mesmo vêm resurgindo, com diversas aplicações desenvolvidas em diversas áreas. Em (CAPPÉ; RYDÉN, 2005) o autor destaca que o HMM se configura como um tipo de modelagem suficientemente genérica para abranger inúmeras complexidades que surgem em dados reais de séries temporais. Certamente sua aplicação mais conhecida é, provavelmente, em reconhecimento de fala. Entretanto já existem estudos aplicados em cadeias de DNA, reconhecimento de escrita manual, reconhecimento de gestos entre outros.

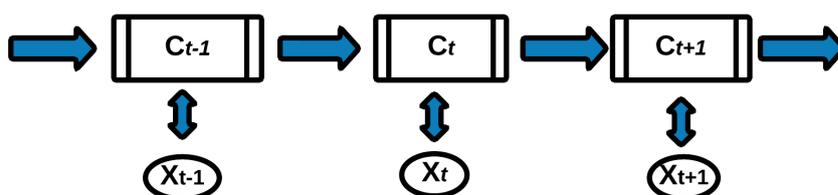
Um HMM  $X_t : t \in \mathbb{N}$  é definido como um conjunto de observações dependentes com registros temporal no tempo 1 a  $t$ ,  $X^{(t)}$  e  $C^{(t)}$ . Onde  $X^{(t)}$  representa as sequências  $X_1, X_2, \dots, X_t$  de valores observados, e  $C^{(t)}$  representa as sequências  $C_1, C_2, \dots, C_t$  de estados ocultos. A es-

estrutura de um modelo HMM é definida pela relação de suas probabilidades de estados ocultos  $P_{ocult}$  (Equação 2.8) e probabilidades de sequências observadas  $P_{obs}$  (Equação 2.9) descritas na Figura 14. Sendo um algoritmo com características probabilísticas sua principal vantagem é de reconhecer e ser ajustável a novos dados onde existe informações incompletas sobre a fonte a partir do qual as sequências com tamanhos variados são geradas, mesmo pertecendo ao mesmo modelo (classe) (LAWRANCE; RABINER, 1989).

$$P_{ocult} = (C_t | C^{(t-1)}) = Pr(C_t | C_{t-1}), t = 2, 3, \dots, n \quad (2.8)$$

$$P_{obs} = (X_t | X^{(t-1)}, C^{(t)}) = Pr(X_t | C_t), t \in \mathbb{N}. \quad (2.9)$$

**Figura 14 – Diagrama da Relação entre Estados Ocultos  $C^{(t)}$  e Valores Observados  $X^{(t)}$  de um modelo HMM.**



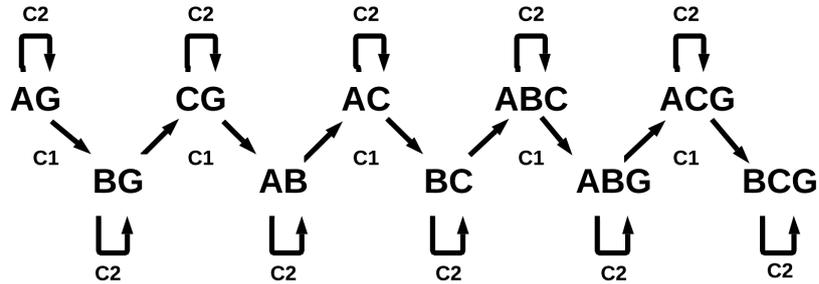
Fonte: (FRONDANA, 2012).

Os parâmetros formais de um modelo HMM discreto (representam classes) com  $X^{(t)}$  sequências de valores observados e  $C^{(t)}$  estados ocultos podem ser representados por  $\lambda = (\delta, \Gamma, \pi)$ . Onde  $\Gamma$  é a matriz de transição de estados,  $\delta$  é o conjunto de funções de probabilidade de observações (geralmente distribuição normal - Gaussiana) e  $\pi$  é um vetor de dimensão S, composto por uma função de densidade de probabilidade dos estados iniciais. O elemento  $a_{ij}$  de  $\delta$  é a probabilidade de realizar uma transição do estado  $i$  para  $j$ , com  $i=1, \dots, S$   $a_{ij} = 1, \forall i$ . Em algumas modelagens, o vetor  $\pi$  foi incorporado em uma matriz diagonal  $P$ , a partir da definição de dois estados que não emitem um símbolo de saída. Ou seja, utilizando  $\lambda = (\delta, \Gamma)$  nos modelos HMM para que condicione a permanecer no estado corrente  $C_2$  ou ir para o próximo estado  $C1$  obedecendo a uma topologia esquerda-direita que é mais utilizada para modelar séries temporais (FRONDANA, 2012). A Figura 15 apresenta a topologia utilizada no estudo para os dez modelos HMMs que representam as classes de faltas em linhas de transmissão.

### 2.4.3 Processo de Treinamento com o algoritmo HMM

O treinamento com o algoritmo HMM é realizado através da estimação do modelo esperado. Suponha uma sequência de observações  $X^{(t)}$  originada de um modelo HMM (Uma classe  $Y$ ). O HMM associado a tal modelo possui  $m$  estados, distribuição inicial  $\delta$  e matriz de transição  $\Gamma$ , com as funções de probabilidade (ou densidade) estado-dependentes  $p_i, i = 1, 2, \dots, i_n$ .

Figura 15 – Estrutura do algoritmo HMM adotado no Estudo com Topologia Esquerda-Direita.



A probabilidade de observar tal sequência  $L_X$  sob o modelo HMM ( $Y$ ) descrito é dita **função de verossimilhança**  $L_{XY}$ , dado pela Equação 2.10.

$$L_{XY} = \delta \Gamma P_{(x_1)} \Gamma P_{(x_2)} \dots \Gamma P_{(x_t)} 1' \quad (2.10)$$

Para (FRONDANA, 2012) o cálculo da verossimilhança requer um número elevado de operações matemáticas, mas a utilização de formas recursivas faz esse número de operações matemáticas diminuir de forma considerável. No estudo uma maneira de alcançar tal objetivo foi fazer uso das probabilidades forward (Equação 2.11) e backward (Equação 2.15) através do algoritmo Baum-Welch para realizar a estimação de máxima verossimilhança das amostras de tamanho  $T$  de cada modelo HMM.

$$\alpha_t = \delta P_{(x_1)} \Gamma P_{(x_2)} \dots \Gamma P_{(x_t)} = \delta P_{(x_1)} \prod_{s=2}^t \Gamma P_{(x_s)} \quad (2.11)$$

$$\beta_t = \Gamma P_{(x_{t+1})} \Gamma P_{(x_{t+2})} \dots \Gamma P_{(x_T)} 1' = \left( \prod_{s=t+1}^T \Gamma P_{(x_s)} \right) 1' \quad (2.12)$$

O algoritmo Baum-Welch é um método de otimização frequentemente utilizado para encontrar estimadores de máxima verossimilhança quando se tem dados incompletos, considerando o fato de que a verossimilhança (ou log-verossimilhança) dos dados completos é provavelmente mais simples de maximizar do que avaliando apenas os dados (incompletos) observados. O algoritmo é composto por dois passos ( $E =$  Expectation) e  $M =$  Maximization)). Para utilização do algoritmo considere que uma sequência de tamanho  $T$  de um HMM foi observada. Então no passo  $E$  a definição será  $C_1, C_2, \dots, C_t$  como a sequência de estados do HMM com um

$$u_j(t) = \begin{cases} 1, & \text{somente } c_t = j, \text{ para } t = 1, 2, \dots, T \\ 0, & \text{caso contrário} \end{cases} \quad (2.13)$$

onde  $u_j(t)$  é a estimativa para a probabilidade de o processo estar no estado  $j$  no tempo  $t(C_t = j)$ , dado que se observou a sequência  $X_t$ , calculado pela Equação 2.14.

$$u_j = \frac{\alpha_t(j)\beta_t(j)}{\delta \Gamma P_{(x_1)} \Gamma P_{(x_2)} \dots \Gamma P_{(x_t)} 1'} \quad (2.14)$$

E o outro termo

$$v_{jk}(t) = \begin{cases} 1, & \text{somente } c_{t-1} = j, \text{ para } k = 2, 3, \dots, T \\ 0, & \text{caso contrário} \end{cases} \quad (2.15)$$

é a estimativa da probabilidade do processo transitar do estado  $j$  para o  $k$  nos tempos  $t-1$  e  $t(C_{t-1} = j, C_t = k)$ , dado que se observou a sequência  $X_t$ . Sendo que este valor pode ser calculado através da Equação 2.16:

$$v_{jk} = \frac{\alpha_{t-1}(j)\gamma_{jk}\rho_k\beta_t(j)}{\delta \Gamma P_{(x_1)} \Gamma P_{(x_2)} \dots \Gamma P_{(x_t)} 1'} \quad (2.16)$$

em que  $\gamma_{jk}$  são as probabilidades de transição que compõem  $\Gamma$ , e  $\rho_k$  são as distribuições de iniciais de estados que compõem  $\delta$ . Com isso, para o passo  $M$ , após o cálculo de  $u_j(t)$  e  $v_{jk}$ , deve-se maximizar cada termo, pois o termo 1 depende apenas de  $\delta$ , o termo 2 de  $\Gamma$  e o termo 3 dos parâmetros da distribuição estado-dependente. Assim, os estimadores dos parâmetros é calculado pela Equação 2.17.

$$\delta = \frac{\hat{u}_j(1)}{\sum_{j=1}^m \hat{u}_j(1)} \quad (2.17)$$

que define estimativa para a probabilidade inicial do estado  $j$ , sendo a estimativa da probabilidade do processo estar no estado  $j$  no tempo 1 ( $C_1 = j$ ), dado que se observou a sequência  $X_t$ . Assim, a máxima verossimilhança  $L_X Y_{max}$  dos dados completos em uma amostra  $X^{(T)}$  de uma classe  $Y$  é dada pela Equação 2.18.

$$L_X Y_{max} = \left( \delta_{c1} \prod_{t=2}^T \gamma_{c_{t-1}, c_t} \prod_{t=1}^T \rho_{c_t}(x_t) \right) \quad (2.18)$$

E o modelo  $M_{L_X Y_{max}}$  de cada HMM esperado é obtido através da média aritmética ( $\mu$ ) da máxima verossimilhança  $L_X Y_{max}$  do conjunto de amostras  $X^{(T)}$  de cada classe  $Y$ .

Assim, no processo de treinamento o algoritmo HMM utiliza um modelo  $M_{L_X Y_{max}}$  para cada classe com os conjunto de parâmetros  $\lambda$  padrão para todas as classes, que foi previamente estimado. Isso envolve, além do emprego das probabilidades iniciais e de transição, as médias ( $\mu$ ) e variâncias ( $\sigma$ ) de cada sequência observada que dependendo de sua variação, podem proporcionar um bom resultado (LAWRANCE; RABINER, 1989; FRONDANA, 2012). Uma sequência  $X_t$  será classificada como de uma determinada classe  $Y$  se a máxima verossimilhança desta sequência para o modelo desta classe ( $M_{L_X Y_{max}} | \lambda_X = L_X Y$ ) for maior que as máximas

Tabela 2 – Exemplo de Classificação da Máxima Verossimilhança de uma Classe.

Sequência	$X_1$	$X_5$	$X_{11}$	$X_{37}$	$X_{42}$
<b>Base de Treino</b>					
Origem	AG	<b>AG</b>	BG	ABC	BG
<b>Base de Teste</b>					
$L_X Y_{max}$	13,1	<b>9,14</b>	11,10	9,42	10,90
$M_{L_X} AG$	13,12	13,12	13,12	13,12	13,12
$M_{L_X} BG$	11,2	11,2	11,2	11,2	11,2
$M_{L_X} CG$	9,23	<b>9,23</b>	9,23	9,23	9,23
$M_{L_X} AB$	13,02	13,02	13,02	13,02	13,02
$M_{L_X} AC$	12,04	12,04	12,04	12,04	12,04
$M_{L_X} BC$	10,70	10,70	10,70	10,70	10,70
$M_{L_X} ABC$	9,56	9,56	9,56	9,56	9,56
$M_{L_X} ABG$	11,79	11,79	11,79	11,79	11,79
$M_{L_X} ACG$	12,93	12,93	12,93	12,93	12,93
$M_{L_X} BCG$	7,88	<b>7,88</b>	7,88	7,88	7,88
<b>Classificação</b>	AG	<b>CG</b>	BG	ABC	BG

verossimilhanças de todos os outros modelos das outras classes. Ou seja, para  $Y$  classes, a sequência  $X_t$  é da classe  $Y$  se

$$L_X Y > M_{L_X Y_{max}}, L \neq M = 1, 2, \dots, M_n \quad (2.19)$$

E também se o rótulo da classe da sequência  $X_{Label}$  for igual ao rótulo da classe do modelo  $M_{Label}$  HMM.

#### 2.4.4 Processo de Teste do algoritmo HMM

O processo de teste do algoritmo HMM segue o mesmo mecanismo utilizado no treinamento. Ou seja, uma sequência de teste  $X_t$  será classificada como de uma determinada classe  $Y$  se a máxima verossimilhança  $L_X Y$  desta sequência para o modelo desta classe  $M_{L_X Y_{max}}$ , estimado no treinamento, for maior que as máximas verossimilhanças de todos os outros modelos das outras classes. E também se o rótulo da classe da sequência de teste  $M_{Label}$  coincide com o rótulo da classe do modelo  $M_{Label}$  HMM obtido no treinamento.

Como exemplo no processo de teste suponha cinco sequências da base de teste  $X_1$ ,  $X_5$ ,  $X_{11}$ ,  $X_{37}$  e  $X_{42}$  conforme apresentado na Tabela 2, onde a sequência  $X_5$  tem máxima verossimilhança 9,14 originada e treinada no modelo AG. Depois de ser comparada com as máximas verossimilhanças dos outros modelos, se torna maior que o modelo BCG de média 7,88, menor do que o modelo CG de média 9,23 e dos demais que também possuem médias superiores. Nesse cenário, a sequência  $X_5$  gera um erro, pois é classificada como do modelo CG. O mesmo não acontece com as outras sequências que são classificadas corretamente.

## 2.5 Classificadores baseados na Arquitetura de Sequência baseada em Quadros

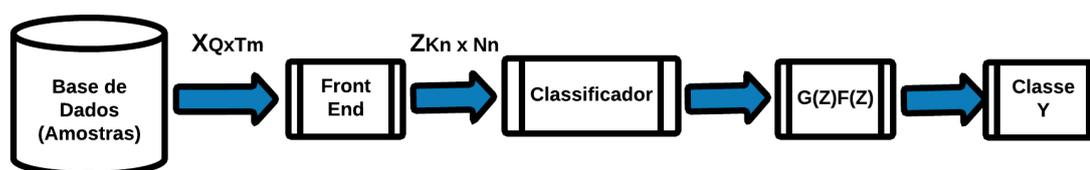
A arquitetura de sequência baseada em quadros (FBSC) faz uso de front ends, uma fase de pré-processamento que organiza os dados em uma matriz de tamanho fixo para ser processado por um classificador convencional. Alguns front ends e a concatenação destes geram um conjunto de dados de alta dimensionalidade, aumentando o custo computacional dos classificadores na fase de treino e teste do modelo. Com objetivo de reduzir a alta dimensionalidade dos dados de entrada, pode ser utilizada uma etapa de seleção de parâmetros antes do estágio de reconhecimento de padrões como apresentado em (HOMCI et al., 2016).

Na arquitetura FBSC o usuário ou algum procedimento automático pode realizar a escolha do *front end*, do classificador e a regra de combinação, permitindo assim um alto grau de liberdade na concepção do classificador, devendo então ser avaliado rigorosamente para evitar conclusões tendenciosas. O front end é responsável por todas as operações de extração de parâmetros (características) de uma base de dados gerando as sequências que serão passadas aos algoritmos de classificação.

Para (MORAIS, 2011) na arquitetura FBSC as sequências de entrada são segmentadas em vetores de tamanho fixo denominados de quadros (frames). Esta arquitetura utiliza um classificador convencional  $F$  como base para processar cada quadro e obter *scores*  $y = \{f_1(z), \dots, f_y(n)\}$  para cada classe. Os scores de todos os  $N$  quadros são contabilizados para se obter uma decisão final, isto é, verifica-se qual a classe classificada com mais frequência nos quadros. Este processo é denominado de *Voting*.

Um classificador convencional  $F$  é um mapeamento  $f : \mathbb{R}^K \rightarrow \{1, \dots, Y\}$ , onde  $K$  é a dimensão do vetor de entrada  $z \in \mathbb{R}^K$  e o rótulo  $y \in \{1, \dots, Y\}$  é a classe. É utilizado um conjunto de treinamento  $T = \{(z_1, y_1), \dots, (z_v, y_v)\}$  contendo  $V$  exemplos de  $(z, y)$ , para treinar um classificador convencional.

**Figura 16 – Fluxo de processamento da arquitetura FBSC. A saída do classificador de sequência  $G(Z)$  depende das decisões do classificador  $F(z_n)$**



O processo de classificação da arquitetura FBSC, ocorre após a execução do classificador  $F$ , seguindo uma sequência iniciada com a submissão das amostras de tamanho variável, a um extrator de parâmetros denominado de *front end*, que converte uma matriz  $X$  representando as faltas em uma matriz  $Z$  com dimensão  $K \times N_n$ , onde  $K$  é o número de parâmetros e  $N_n$  o número de vetores de parâmetros do  $n$ -ésimo exemplo. Esta arquitetura executa a classificação na matriz

$Z$  e não em  $X$ . A matriz  $X$  é composta pela concatenação das amostras originais e organizadas em uma matriz com dimensões  $Q \times L$ , onde  $Q$  são valores de tensão e corrente da amostra e  $L$  é o tamanho do quadro e sua concatenação é  $\hat{Z} = [F_1, \dots, F_{N_n}]$ , gerando uma matriz com dimensão  $Q \times L_n$ , e  $N$  é número de quadros. No caso de sobreposição é considerado um deslocamento  $S$  (quantidade de amostras entre o início de dois quadro consecutivos), podendo ser menor que o tamanho da janela. Uma falta  $X_n$  é representada pelo número de quadros  $N_n$

$$N_n = 1 + \lfloor (T_n - L)/S \rfloor \quad (2.20)$$

e a função *floor* é representada por  $\lfloor - \rfloor$ . Quando o deslocamento é igual ao tamanho do quadro,  $S = L$  (não tem sobreposição) e uma concatenação de amostras representam um quadro onde as matrizes  $X = \hat{Z}$  coincidem.

A taxa de erro da classificação é um parâmetro utilizado para avaliar o desempenho de um classificador de sequência  $G(Z)$

$$E_s = \frac{1}{R} \sum_{r=1}^R I(G(Z_r) \neq y_r) \quad (2.21)$$

$I$  é a função indicador, que é 1 caso o argumento seja verdadeiro e 0 caso contrário.  $R$  representa a sequência de teste. No caso da FBSC,  $E_s$  depende diretamente do desempenho do classificador  $F$ .

### 2.5.1 *Front ends*

### 2.5.2 Direto da forma de onda - *Raw*

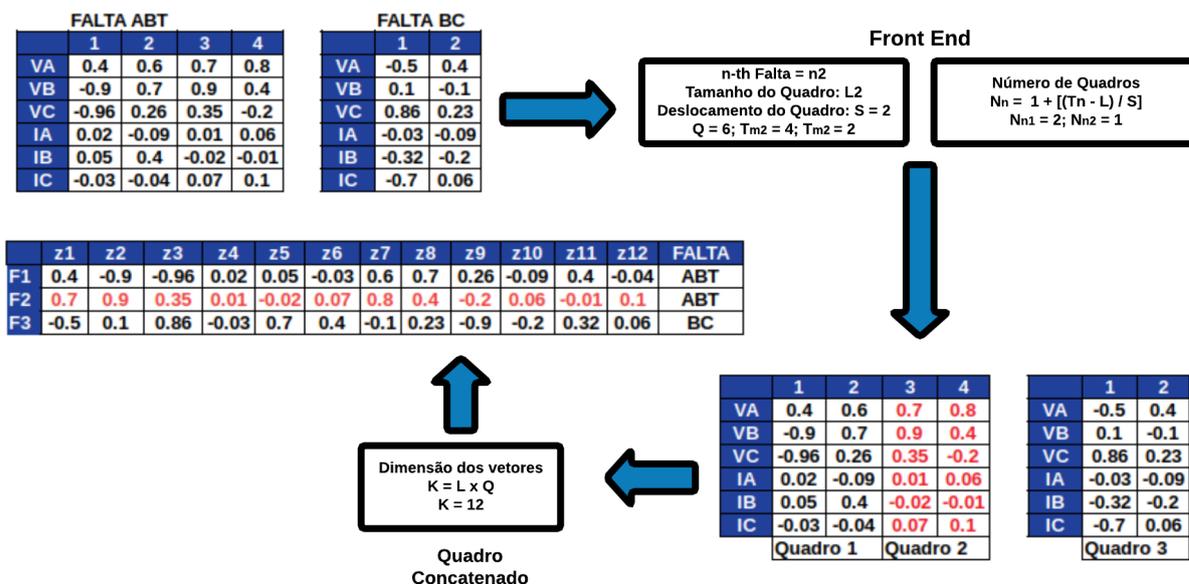
Para (MORAIS et al., 2010a) o *front end raw* é o mais simples, pois seus parâmetros de saída correspondem a valores da amostra original da falta, sem qualquer outro processamento que organize as mesmas para uma matriz  $Z$  onde ocorrerá a classificação. Sendo, assim amostras da matriz  $Z$  são organizadas de acordo com o tamanho da janela (quadro - *frame*)  $L$  e o deslocamento do mesmo representado por  $S$ . É importante ressaltar quando  $S = L$  (não há sobreposição) o quadro será uma concatenação de amostras na forma

$$\mathbf{F}_n = [\mathbf{x}_{(1+L(n-1))}, \mathbf{x}_{(2+L(n-1))}, \dots, \mathbf{x}_{(L+L(n-1))}], \quad (2.22)$$

Mais especificamente, se  $Q = 6$  (tensões e correntes), um *front end raw* poderia obter quadros  $\mathbf{F}_n$  de dimensão  $6 \times 5$  concatenando para cada amostra central seus quatro vizinhos, dois à esquerda e dois à direita. Neste caso, supondo uma falta com  $T_m = 10$  amostras e  $L = S = 5$ , os valores de  $K$  e  $N_m$  são respectivamente, 30 e 2. Desta forma,  $\hat{Z}$  e  $Z$  poderiam ter dimensões  $6 \times 10$  e  $30 \times 2$ , respectivamente. A Figura 17 ilustra a segmentação de uma falta “ABT” (4 amostras) e uma falta “BC” (2 amostras) em vetores de parâmetros com 2 e 1 quadro, respectivamente. Neste

exemplo,  $L = 2$  e  $S = 2$  (não há sobreposição) o que fornece 3 vetores  $z$ , cada um de dimensão  $K = 12$ .

Figura 17 – Demonstração do *front end* Raw, organizando os vetores de padrão  $z$ . Neste exemplo,  $L = 2$  e  $S = 2$  (não há sobreposição) e a matriz obtida pela aplicação do *front end* possui dimensão  $K = 12$  com quadros da falta “ABT” e um quadro falta “BC”.



Fonte: Adaptado de (COSTA et al., 2017).

### 2.5.3 *Front end* RMS

Várias normas de QEE classificam os fenômenos de acordo com as características como o tempo, em que o valor médio quadrático (RMS - Root Mean Square) da tensão extrapolou um limiar. O RMS é um dos *front ends* mais usados no processo de parametrização de amostras, permitindo uma maior interpretabilidade quando comparado a outros *front ends*, além de permitir uma maior estimativa aproximada da amplitude da frequência fundamental de uma forma de onda (MORAIS et al., 2010b).

Este *front end* consiste em calcular o valor RMS janelado para cada uma das formas de onda  $Q$ . Considerando o tamanho da janela  $L$  e o deslocamento  $S$ , o  $n$ -ésimo valor RMS  $z[n], n = 1, \dots, N$  de uma forma de onda  $x[t], t = 1, \dots, T$  é dado por

$$z[n] = \sqrt{\frac{1}{L} \sum_{l=1}^L (x[l + (n-1)S])^2}, \quad (2.23)$$

onde  $N$  e  $T$  são apresentados na Equação.( 2.20).

Onde:

- $N$  é calculado de acordo em Equação 2.20
- $T$  é o número de amostras.
- $L$  é o tamanho do quadro.
- $S$  é o deslocamento do quadro.

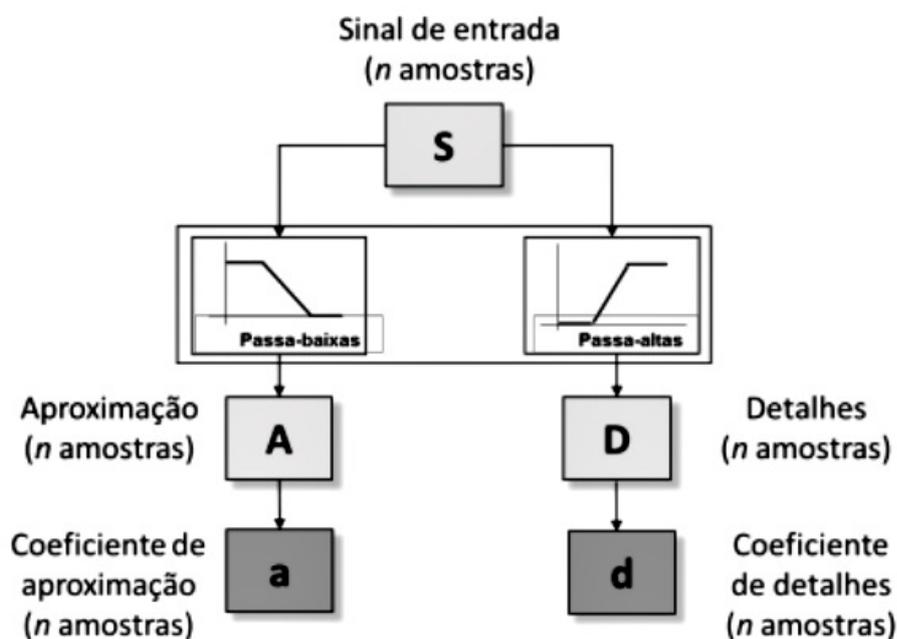
### 2.5.4 *Front ends wavelets*

São *front ends* classificados como multi-resolução (um sinal pode ser analisado em diferentes frequências com diferentes resoluções), que utilizam a técnica de transformada *wavelet* para fornecer informações do sinal no domínio do tempo e frequência simultaneamente, sendo bastante utilizados para prover estudos em sistemas elétricos pois, são capazes de identificar características únicas das diferentes faltas localizadas nos vários níveis de decomposição do sinal para o processo de classificação de faltas em linha de transmissão (SILVA; DANTAS; SOUZA, 2006; SILVA; SOUZA; BRITA, 2006; VETTERLI; KOVACEVIC, 1995).

O sinal analisado é dividido em componentes de alta escala e baixa escala. As primeiras são chamadas de “aproximações”, já que correspondem ao conteúdo de baixa frequência do sinal. As variações rápidas do sinal são chamadas de “detalhes”. Para exemplificar, no seu nível mais básico, o processo de filtragem que gera a divisão mencionada pode ser entendido através da ilustração da Figura 18, onde são utilizados dois filtros para decompor o sinal  $S$ , obtidos através da sua convolução repetidas vezes. O primeiro é um filtro passa-baixa que representa a geração do conteúdo de baixa frequência ou aqui chamado de coeficiente de aproximação  $a$  e

um segundo filtro passa-alta com o conteúdo de alta-frequência com os detalhes representados pelos coeficientes de detalhe **d**.

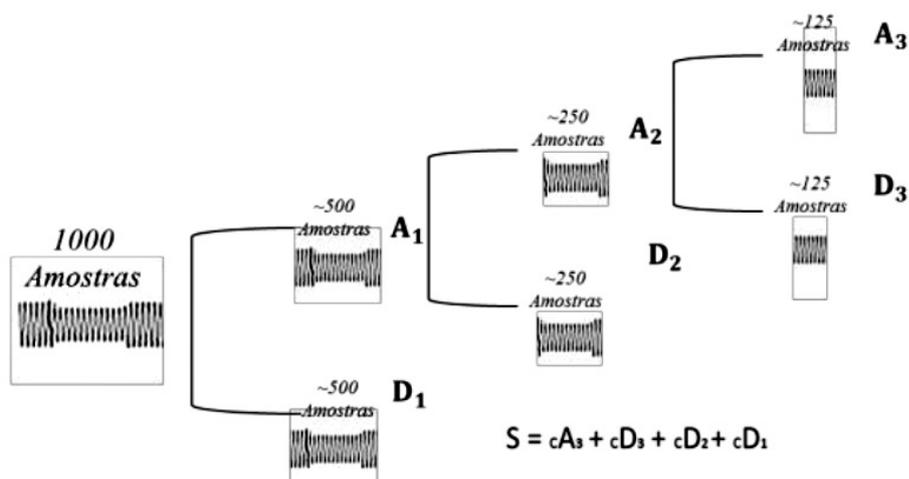
**Figura 18 – Filtragem de um estágio para geração de aproximações (A) e detalhes (D) de um sinal (S).**



Fonte: (HOMCI et al., 2016)

Após a decomposição do sinal, a primeira por si só é então decomposta em aproximações e detalhes de segundo nível, e o processo se repete, como ilustrado na Figura 19. Para  $n$  níveis, a *wavelet* oferece  $n + 1$  possibilidades de decomposição do sinal.

**Figura 19 – Exemplo de decomposição de 3 níveis.**

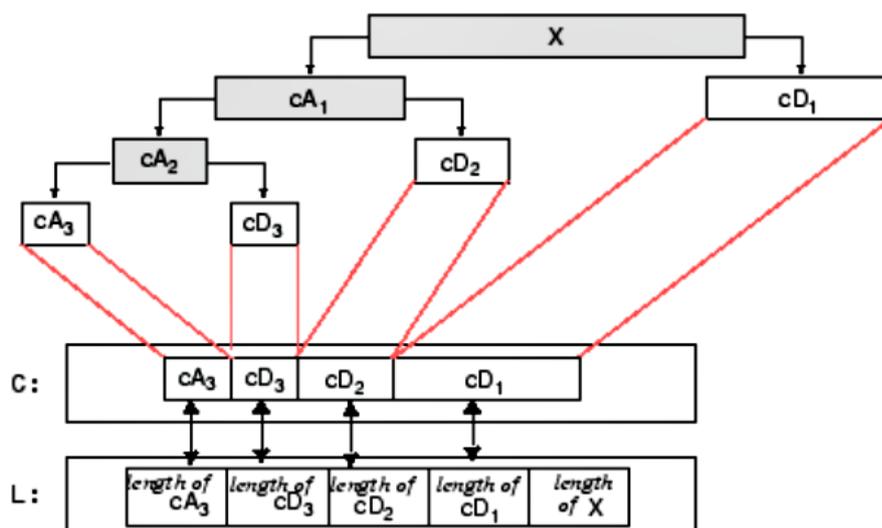


Fonte: (HOMCI et al., 2016)

A Figura 20 ilustra como ocorre o processo de decomposição *wavelet* utilizando a função *wavedec* do MATLAB, para 3 níveis de decomposição. Essa função retorna dois vetores

C e L, onde C é um vetor que armazena os coeficientes de decomposição (cD3, cD2 e cD1) e o último de aproximação (cA3), e L contém os tamanhos de cada coeficiente de C e o tamanho do sinal original.

Figura 20 – Decomposição wavelet a partir da função wavedec do MATLAB.



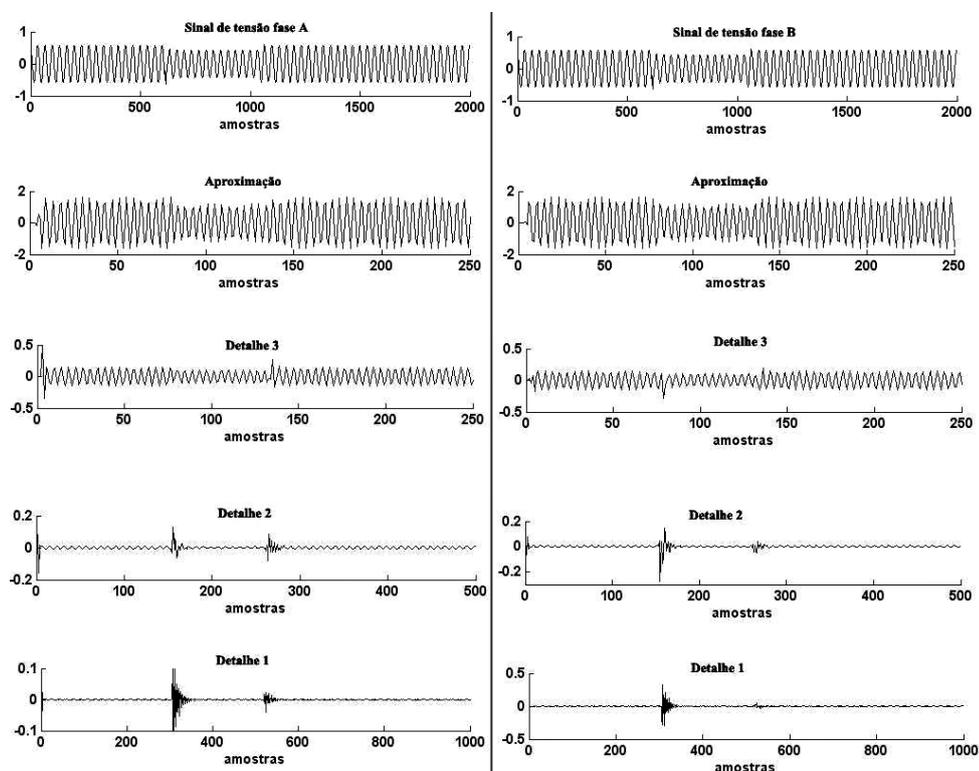
Fonte: (COSTA et al., 2017)

Assume-se uma decomposição wavelet diádica, que tem  $\gamma$  estágios de filtragem e decimação (VETTERLI; KOVACEVIC, 1995) e transforma cada uma das formas de onda  $Q$  em  $\gamma + 1$  formas de onda. Mais especificamente, a  $q$ -ésima forma de onda é decomposta em aproximação  $a^q$  e detalhes  $d_1^q, d_2^q, \dots, d_{\gamma}^q$  para  $q = 1, \dots, Q$ . Por simplicidade, a dependência em relação a  $q$  será omitida. A Figura 21 apresenta um exemplo de aplicação da transformada wavelet aos sinais de tensão das fases A e B de uma falta AB.

Alguns trabalhos na literatura usam somente um dos coeficientes de detalhe (AGUILERA; ORDUNA; RATTA, 2006; SAWATPIPAT P.; TAYJASANANT, 2010; SINGH; PANIGRAHI; MAHESHWARI, 2011). Neste caso, a decomposição *wavelet* discreta no tempo é equivalente a uma operação de filtragem digital simples. Além disso, não compensa realizar a decomposição *wavelet* por completo para depois descartar todos os conjuntos de coeficientes e no final armazenar somente um. Neste caso, é mais eficiente implementar diretamente uma filtragem digital com a banda passante de interesse.

Neste trabalho, o front end wavelet concatena todos os coeficientes e organiza-os em uma Z. Este leva em consideração que para  $\gamma > 1$  os coeficientes possuem diferentes frequências de amostragem. Para isso, em vez de usar um único L, o usuário especifica um valor  $L_{\min}$  para as formas de onda com a menor frequência de amostragem  $f_s$  ( $\mathbf{a}$  e  $\mathbf{d}_{\gamma}$ ) e um valor maior  $L_i = 2^{\gamma-i} L_{\min}$  é automaticamente adotado para outros detalhes  $i = 1, \dots, \gamma - 1$ . Por exemplo, assumindo uma aplicação *wavelet* com 3 níveis de decomposição ( $\gamma = 3$ ), o tamanho dos quadros para  $\mathbf{d}_1$  e  $\mathbf{d}_2$  são, respectivamente,  $4L_{\min}$  e  $2L_{\min}$ . Um raciocínio similar é aplicado para o deslocamento do quadro  $4L_{\min}$  e  $2L_{\min}$ , onde  $S_{\min}$  é outro parâmetro definido pelo usuário.

**Figura 21 – Exemplo de uma decomposição *wavelet* aplicada ao sinais de tensão das fases A e B de uma falta AB simulada no intervalo de 1s. A *wavelet* mãe é uma *Daubechies 4* com três níveis de decomposição ( $\gamma = 3$ ).**



Fonte: (MORAIS et al., 2010a)

Em suma, após aplicação da decomposição *wavelet*, os coeficientes são então organizados em um quadro  $F$  de dimensão  $QL$ , onde  $L_i = 2^{\gamma-i}L_{\min}$ . Neste caso, o número de parâmetros da waveletconcat é  $K = 2^{\gamma}L_{\min}Q$  e o número de quadros é dado por

$$N = 1 + (T - L_{\min})/S_{\min} \quad (2.24)$$

onde  $T_a$  é o número de elementos em  $\mathbf{a}$ . Este *front end* é chamado na literatura de waveletconcat (MORAIS et al., 2010a).

Outra alternativa para organizar os coeficientes wavelet é calculando a energia média de cada coeficiente. Este *front end* é chamado de waveletenergy, e semelhante ao waveletconcat, este tem que tratar com sinais de diferentes frequências de amostragem. A principal diferença entre esses *front ends* é que, em vez de concatenar todos os coeficientes, waveletenergy representa  $\mathbf{X}$  por meio da energia  $E$  em cada banda de frequência obtida pela decomposição wavelet. A energia é estimada no estilo "média móvel" (*moving average*), ou seja, calcula-se a energia para curtos intervalos ou quadros (mas deve-se evitar confusão com o já utilizado termo "quadro") (COSTA et al., 2017). Por exemplo, assumindo  $L_{\min} = S_{\min} = 1$  e novamente uma matriz  $\mathbf{X}$  de dimensão  $6 \times 1000$ , temos uma matriz  $6 \times 4$ , onde a primeira linha contém a energia média do coeficiente wavelet  $\mathbf{a}$  calculado a partir da tensão da fase A, logo para o caso da waveletenergy  $K = 24$  e  $N = 125$ .

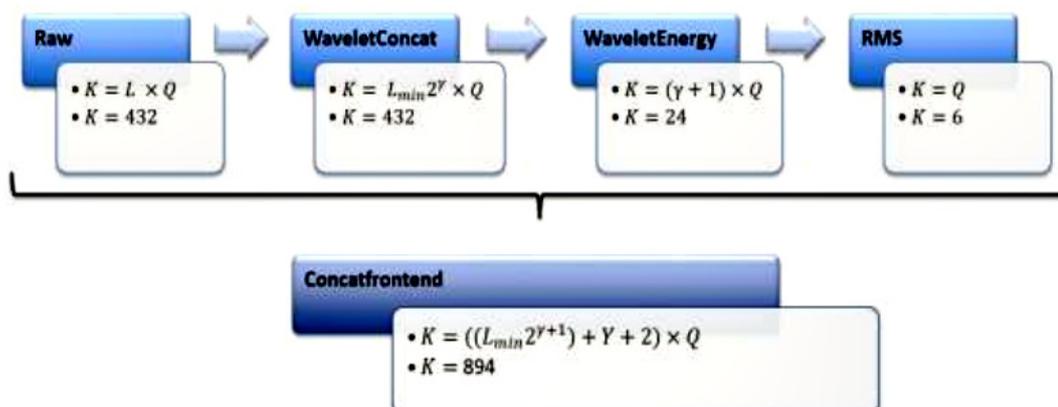
### 2.5.5 Concatenação de *fronts ends*

Nas subseções anteriores foram citados os *fronts ends* utilizados para comparação neste trabalho, assim como todas suas características e parâmetros que incidem diretamente no resultados de desempenho da classificação de faltas em linhas de transmissão, com o objetivo de alcançar o melhor desempenho.

(HOMCI et al., 2016) sugere a concatenação dos *fronts ends* raw, waveletconcat, waveletenergy e RMS, para gerar um único conjunto de sequências denominado de *concatfrontend*. O processo é realizado em duas etapas, a primeira é realizado o processo de concatenação por sinal, onde os *front ends* são calculados para cada sinal (tensão e corrente; fases A, B e C) e agrupados na ordem definida previamente. Para tal, o tamanho do quadro  $L$  e o deslocamento do quadro  $S$  devem ser calculados, onde  $L = L_{min}2^\gamma$  e  $S = S_{min}2^\gamma$ , com os parâmetros de janelamento e decomposição *wavelet* definidos, é possível gerar todos os *front ends* para cada sinal e concatená-los numa matriz na seguinte ordem: primeiramente as tensões das fases A, B e C; em seguida as correntes das fases A, B e C.

A Figura 22 ilustra a concatenação dos *front ends*, utilizando os seguintes parâmetros, tamanho da janela  $L_{min} = 9$  e o deslocamento do quadro  $S_{min} = 4$ , bem como o número de elementos  $Q = 6$  e o nível de decomposição *wavelet* = 3, observa-se que dependendo do *frontend*, o número de parâmetros  $K$  cresce muito rápido, gerando matrizes de alta dimensionalidade.

**Figura 22 – Processo de concatenação dos *front ends*, mostrando o número de parâmetros  $K$  para cada *front end*, considerando  $L_{min} = 9$  e  $S_{min} = 4$ .**



Fonte: (HOMCI et al., 2016)

O *concatFrontEnd* gera uma matriz com um grande número de parâmetros, ocasionando um elevado custo computacional quando submetido a um classificador, Como consequência desse processo, temos o surgimento de um problema que várias aplicações possuem que é geração de uma sequência com alto grau de dimensionalidade (PRIDDY; KELLER, 2005). Por esse motivo (HOMCI et al., 2016) adota técnicas de seleção de parâmetros baseada em filtros, *wrapper* e algoritmos híbridos para reduzir a dimensão dos dados de entrada selecionando os parâmetros mais relevantes, antes do estágio de reconhecimento de padrões.

## 2.6 Medidas de desempenho adotadas para os classificadores

### 2.6.1 Taxa de Erro

As medidas de desempenho servem para avaliar o desempenho de um classificador  $f$ . Entre inúmeras métricas de avaliação de um classificador pode-se citar a taxa de acerto ( $ac(f)$ ), taxa de erro ou classificações incorretas ( $err(f)$ ).

Considerando um conjunto de dados contendo  $n$  objetos, a taxa de erro apresentada na Equação 2.25 equivale à proporção de exemplos desse conjunto classificados incorretamente por  $f$  e é obtida pela comparação da classe conhecida de  $x_i$ ,  $y_i$ , com a classe predita,  $f(x_i)$ . Esse tipo de medida equivale ao uso da função de custo 0-1 relacionando os rótulos dos objetos às predições obtidas (HALL et al., 2009).

$$err(f) = \frac{1}{n} \sum_{i=1}^n I(y_i \neq f(x_i)) \quad (2.25)$$

A taxa de erro varia entre 0 e 1, e valores próximos a 0 são índices melhores nas predições dos classificadores.

Em (FACELI, 2011) os autores afirmam que a taxa de acerto (acurácia), apresentada na Equação 2.26, serve como complemento da taxa de erro.

$$ac(f) = 1 - err(f) \quad (2.26)$$

A taxa de acerto também varia entre 0 a 1, entretanto ao contrário da taxa de erro, valores próximos a 1 apresentam melhores predições nas classificações.

### 2.6.2 Custo computacional

Um algoritmo em termos computacionais pode ser definido como uma descrição de um padrão de comportamento para realizar alguma tarefa expresso em termos de um conjunto finito de tempo e ações, assim como em uma ótica matemática se constitui como um conjunto de processos (e símbolos que os representam) para efetuar um cálculo. Nesse sentido, os mesmos podem repetir passos (fazer iterações) ou necessitar de decisões (tais como comparações ou lógica) até que a tarefa seja completada. Um algoritmo implementado resolve um problema para determinada entrada produzindo uma resposta correta (CARMEM, 2002).

Um algoritmo corretamente executado não irá resolver um problema se estiver implementado incorretamente ou se não for apropriado ao problema. Em sistemas computacionais uma forma de mensurar seu desempenho é utilizar métodos de análise para determinar sua eficiência em termos de número de operações e tempo de execução, ou seja, seu custo computacional (VELOSO, 2002).

Para (CARMEM, 2002) o custo computacional de um algoritmo é uma medida de desempenho que serve para mensurar a sua eficiência para resolver domínios de problemas específicos. Sendo assim, depois da implementação de um algoritmo pode-se analisar seu desempenho para verificar a sua eficiência e ainda ter a preocupação de projetar algoritmos eficientes desde a sua concepção para o modelo proposto. Para mensurar o desempenho de um algoritmo é necessário definir alguma medida que expresse a eficiência. Costuma-se medir um algoritmo através de sua complexidade algorítmica em termos de tempo de execução ou como o espaço usado(ou memória) então:

- Para o tempo, pode-se considerar o tempo absoluto (em minutos, segundos, etc.). Medir o tempo absoluto irá depender do hardware da máquina.
- Em espaço de memória, pode-se contar o número de operações consideradas relevantes realizadas pelo algoritmo e expressa-se esse número como uma função de  $n$  operações. Essas operações podem ser comparações, operações aritméticas, movimento de dados, etc.

Para classificação de faltas em linhas de transmissão de energia proposto neste trabalho optou-se em adotar o tempo de execução como medida de desempenho para mensurar a eficiência da arquitetura FBSC e dos algoritmos HMM e KN-DTW em uma máquina com processador i7 com 16G de memória RAM sobre o sistema operacional Linux da distribuição Ubuntu, versão 18.04 - licença GPL (General Public license).

### 2.6.3 Testes de significância estatística

Testes estatístico correspondem a uma regra decisória que permite aceitar ou não uma hipótese com base nos resultados de amostras. Os autores de (SCUDINO, 2008) colocam que o teste estatístico é um mecanismo de significância precisa para amostras de predição, pois contém recursos de cálculos estatísticos mais direcionado para tarefas de classificação. Essa técnica que foi adotada no estudo para comparar os resultados de classificação entre dois classificadores com o qual se testou as seguintes hipóteses abaixo:

- $H_0 : \bar{X}_0 = \bar{X}_1$  não existe diferença significativa entre as médias das taxas de erro dos classificadores 1 e 2.
- $H_1 : \bar{X}_0 \neq \bar{X}_1$  existe diferença significativa entre as médias das taxas de erro dos classificadores 1 e 2.

A partir da Equação 2.27.

$$t = \frac{(\bar{X}_1 - \bar{X}_2)}{S_{x_1x_2} \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \quad (2.27)$$

Em que  $S_{x_1x_2}$  é:

$$S_{x_1x_2} = \sqrt{\frac{(n_1 - 1)S_{x_1}^2 + (n_2 - 1)S_{x_2}^2}{n_1 + n_2 - 2}} \quad (2.28)$$

Sendo o grau de liberdade para esses casos igual a:

$$df = n_1 + n_2 - 2 \quad (2.29)$$

onde:

- $\bar{X}_1$  corresponde a média da taxa de erro do classificador 1.
- $\bar{X}_2$  corresponde a média da taxa de erro do classificador 2.
- $n_i$  é o número de folds (experimentos) para cada classificador.
- $S_{x_i}$  corresponde ao desvio padrão da taxa de erro do classificador  $i$ :

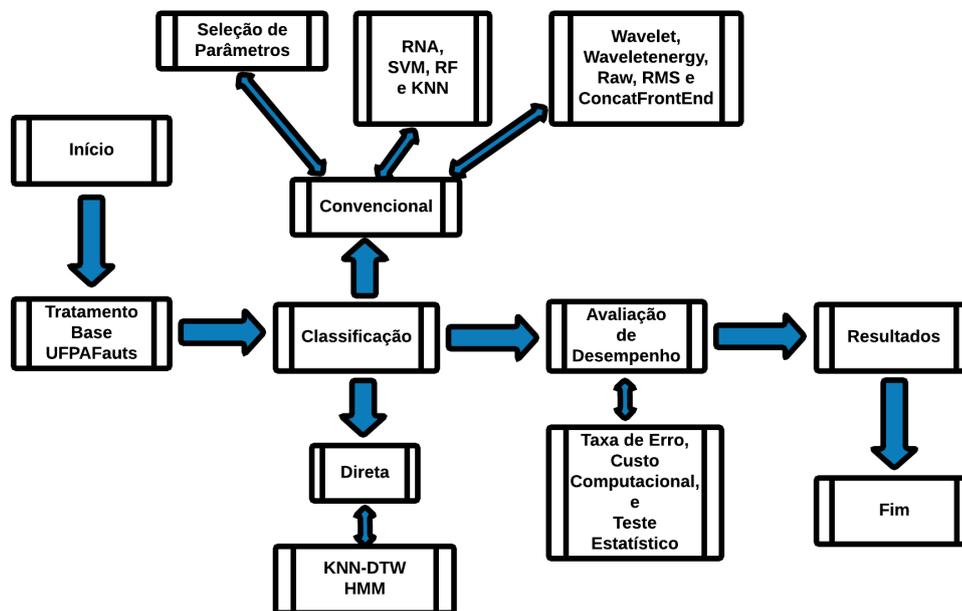
A ideia básica é que exista uma hipótese nula  $H_0$  e outra alternativa  $H_1$  para depois confrontar os resultados obtidos em valor tabelado  $t_{tab}$  da distribuição estatística de acordo com os graus de liberdade  $df$ . O valor calculado  $t_{calc}$  é extraído na Equação 2.27 (diferença das taxas de erro entre dois classificadores) e o valor de  $t_{tab}$  é obtido conforme a tabela t-Student (DANCEY; REIDY, 2006).

### 3 METODOLOGIA DO ESTUDO

Este trabalho concentra-se na avaliação do desempenho dos algoritmos KNN-DTW e HMM para classificação de faltas em linhas de transmissão de energia e sua comparação com classificadores baseados na arquitetura FBSC.

A metodologia do estudo é dividida em quatro etapas, conforme Figura 23 e usa em seus experimentos a base de dados UFPAFaults. Esta base é devidamente rotulada e composta por sequências correspondentes que representam classes de faltas do tipo curto-circuito.

Figura 23 – Metodologia do Estudo.



Na primeira etapa foram extraídos os atributos mais relevantes da base de dados UFPAFaults que foram posteriormente utilizados no desenvolvimento das arquiteturas propostas para classificação de faltas. Na segunda etapa foi desenvolvido o classificador de faltas baseado no KNN-DTW. Na terceira etapa, para comparação com os algoritmos propostos KNN-DTW e HMM, foram desenvolvidos alguns classificadores baseados na arquitetura FBSC, conforme a metodologia apresentada em (MORAIS, 2011). Na quarta etapa a foi desenvolvido o classificador HMM e realizada então sua comparação com os classificadores KNN-DTW e a arquitetura FBSC. Para análise de desempenho dos classificadores foi utilizada a medida do erro assim como testes de significância estatística.

#### 3.1 Base de dados UFPAFaults

O acesso na literatura a resultados utilizando base de dados com eventos de QEE não é uma tarefa trivial, visto que a maioria delas são proprietárias e fatores como tempo, custo e confidencialidade dos dados fazem com que as bases não sejam disponibilizadas. Incluí-se nesse

contexto a falta de disponibilização de bases rotuladas para atender aplicações relacionadas à classificação de faltas em linhas de transmissão, foco deste trabalho. Considerando esta dificuldade, o grupo de pesquisa do Laboratório de Processamento de Sinais (LaPS) da Universidade Federal do Pará, criou e disponibilizou a base de dados UFPAFaults que é uma base rotulada com faltas do tipo curto-circuito em linhas de transmissão e pode ser encontrado para estudos em <https://github.com/jeanarouche/HMM-KNN-DTW-FaultClassification>. Ao longo do tempo a base foi sendo incrementada e a versão mais atual possui 27500 simulações, organizadas em cinco conjuntos, variando de 100 a 1000 simulações de faltas. Todas as simulações são formas de onda de tensão e corrente representando faltas do tipo curto-circuito em linhas de transmissão. Considerou-se que as mesmas podiam ocorrer com a mesma probabilidade em uma linha ou em qualquer uma das linhas do circuito considerado.

A base de dados UFPAFaults foi gerada utilizando dois softwares: *ATP (Alternative Transient Program)* (SCOTT; LIU, 1992) e *AmazonTP* (PIRES, 2009). O ATP é um simulador bastante conhecido na área de sistemas de potência para execução de análises e simulações digitais de eventos. Basicamente, ele gera um modelo do circuito elétrico a ser simulado, apresentando simulação de tensão, corrente e potência elétrica. O papel do AmazonTP é invocar o ATP repetidas vezes, manipulando os modelos gerados conforme a necessidade. Os modelos foram manipulados através do fechamento de chaves do circuito, a fim de se obter faltas simuladas.

As simulações tiveram duração de 1 segundo e são apresentados em dois tipos de arquivos. O primeiro é um TXT que possui informações gerais sobre todas as simulações, como numeração, tipo de falta, quando a falta começa e termina, entre outros. O segundo arquivo está no formato ASCII, e é o que contém as formas de onda em si, passando o intervalo de tempo e valores de tensão e corrente.

Na Figura 24 é possível verificar o cabeçalho e como os dados estão organizados por coluna nos arquivos ASCII. Este arquivo é organizado da seguinte maneira: a primeira coluna indica o intervalo de tempo, as colunas 2, 3 e 4 correspondem aos valores de tensão e as colunas 5, 6 e 7 indicam os valores de corrente.

Em síntese, por exemplo, para 1000 simulações deve existir um arquivo contendo as informações gerais sobre as faltas de cada simulação e 1000 arquivos com informações das formas de onda, considerando uma frequência de amostragem de 40 KHz retratando a realidade de uma LT. A base apresenta 11 tipos de faltas com o mesmo peso uniformemente distribuídas: AT, BT, CT, AB, AC, BC, ABC, ABT, ACT, BCT e ABCT. Vale destacar que ABC e ABCT são consideradas a mesma classe neste trabalho. A Figura 25 apresenta um exemplo de falta AB da base de dados.

É importante destacar que as amostras disponibilizadas na base passaram por um processo de pré-processamento que tem como objetivo aplicar técnicas para permitir que os dados sejam representados de uma forma mais adequada, sem alterar sua organização ao longo do tempo.

Figura 24 – Arquivo ASCII representando as informações associadas disponibilizada na base UF-PAFaults.

Intervalo do tempo	Valores de Tensão (A, B e C)			Valores de Corrente (A, B e C)		
0.0	180246.	-18403.	-.162E6	-8.359	273.245	-264.89
.25E-4	179457.	-16541.	-.163E6	-11.287	274.628	-263.34
.5E-4	178652.	-14677.	-.164E6	-14.213	275.988	-261.77
.75E-4	177832.	-12812.	-.165E6	-17.139	277.322	-260.18
.1E-3	176996.	-10946.	-.166E6	-20.062	278.633	-258.57
.125E-3	176144.	-9079.	-.167E6	-22.984	279.918	-256.93
.15E-3	175277.	-7211.1	-.168E6	-25.905	281.178	-255.27
.175E-3	174394.	-5342.5	-.169E6	-28.822	282.414	-253.59
.2E-3	173495.	-3473.5	-.17E6	-31.738	283.624	-251.89
.225E-3	172581.	-1604.2	-.171E6	-34.65	284.81	-250.16
.25E-3	171652.	265.309	-.172E6	-37.559	285.97	-248.41
.275E-3	170708.	2134.76	-.173E6	-40.465	287.104	-246.64

Fonte: (COSTA et al., 2017)

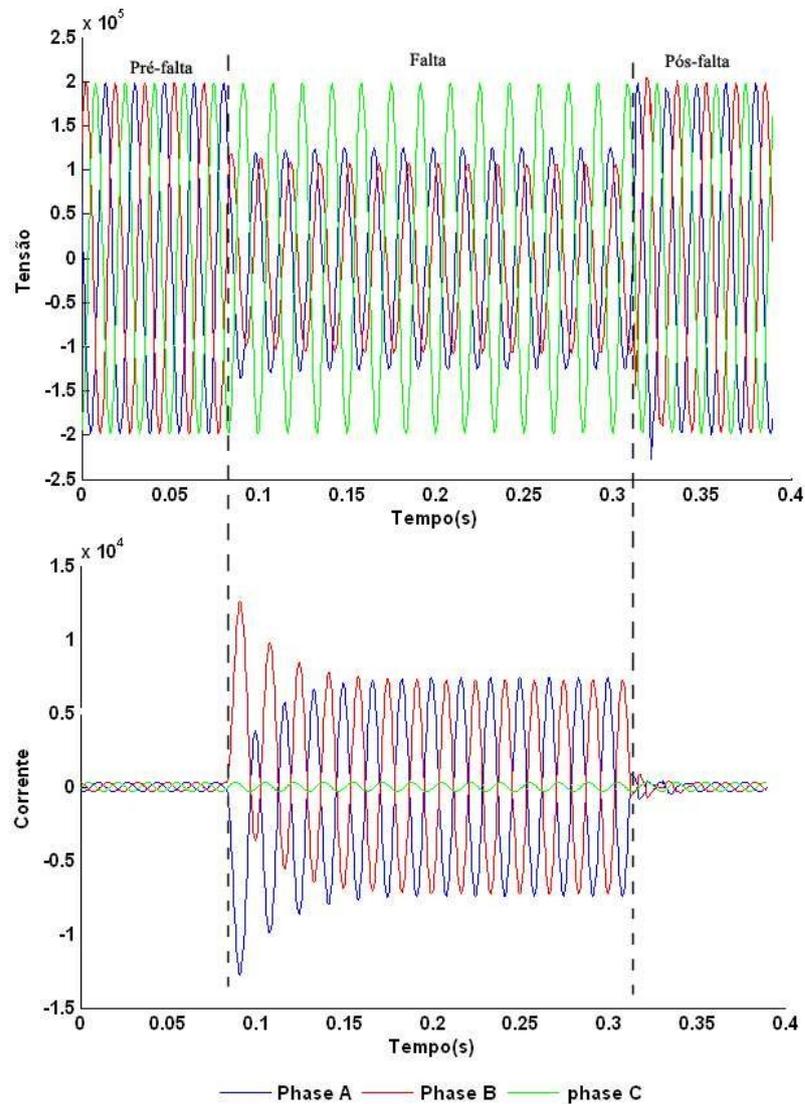
Este processo faz parte da parametrização da base que ocorre em três etapas: primeiramente realizou-se o pré-processamento que consiste em reamostrar os dados para uma frequência de amostragem que neste caso foi de 2 kHz e normalizar os dados para intervalo entre -1 e 1.

A reamostragem foi necessária devido as formas de onda geradas pelas simulações ATP terem um período de amostragem igual a 0.25 microssegundos o que corresponde a uma frequência de amostragem de 40 kHz gerando uma grande quantidade de dados. Assim, no pré-processamento para reduzir este número de amostras do sinal, usou-se a subamostragem com o objetivo de reduzir a quantidade de amostras para diminuir o custo computacional despendido ao se aplicar algoritmos de mineração de dados na base.

A Figura 26 apresenta um exemplo de sinal da base de dados já com reamostragem e normalização.

### 3.2 Configurações Gerais dos Experimentos para os algoritmos KNN-DTW, HMM e para a arquitetura FBSC

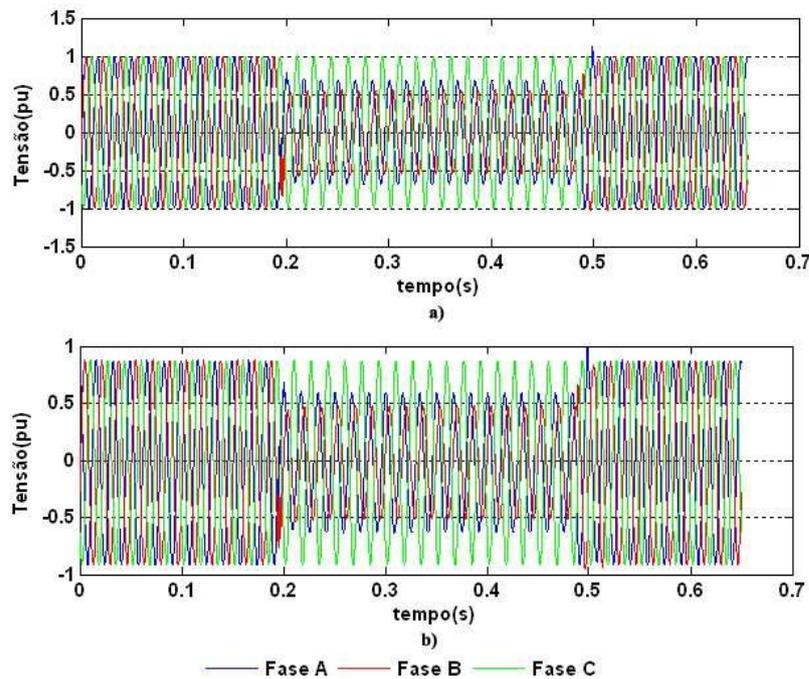
De forma geral para o desenvolvimento dos classificadores propostos usando KNN-DTW e HMM e o classificador baseado na arquitetura FBSC foram utilizados em todos os experimentos 10 conjuntos de treinamento com quantidade diferentes de amostras distribuídas

**Figura 25 – Formas de onda no momento de uma falta AB.**

**Fonte: (COSTA et al., 2017)**

em 100, 200, 300, 400, 500, 600, 700, 800, 900 e 1000 e uma base de teste com 1000 amostras. Esta divisão da base de treino servirá para analisar a importância da quantidade de amostras da base de treino para obtenção de melhores resultados dos classificadores. Todos os treinamentos para as diferentes arquiteturas, foram realizados em 5 experimentos, variando-se os valores dos parâmetros dos classificadores, sendo todos os classificadores testados individualmente com a base de teste de 1000 amostras. Os experimentos foram realizados em uma máquina com processador intel i7 e 16 GB de memória RAM com o sistema operacional linux, distribuição Ubuntu versão 18.04 sob licença GPL. Para cada técnica de classificação existem suas particularidades tais como:

- Classificação baseada no KNN-DTW: extrai conjuntos da bases de dados em forma específica da base UFPAFaults para ser utilizada com o algoritmo sem a necessidade de

**Figura 26 – Sinal com reamostragem e normalização.**

Fonte: (COSTA et al., 2017)

*front ends*. A base de dados usada por este classificador vem na forma como apresentado na secção 3.1 para o caso do uso do *front end raw* que é o *front end* que trabalha diretamente com os dados sem passar por processo de extração de características;

- Classificação baseada em HMM: extrai conjuntos de bases de dados em forma específica da base UFPAFaults para ser utilizada com o algoritmo HMM sem a necessidade de *front end*. A base de dados usada por este classificador vem na forma como apresentado na secção 3.1 para o caso do uso do *front end raw* que é o *front end* que trabalha diretamente com os dados sem passar por processo de extração de características;
- Classificação baseada na arquitetura FBSC: extrai conjuntos de bases de dados em forma específica da base UFPAFaults, utilizando a arquitetura FBSC auxiliada pelos *front end* Wavelet, Waveletenergy, Raw, RMS e ConcatFrontEnd para parametrizar as entradas para os classificadores convencionais RNN, SVM, KNN e RF. Além disso, foi utilizado a SAM para escolha dos melhores valores dos parâmetros desses classificadores;

Para desenvolvimento dos classificadores utilizando a arquitetura FBSC foi utilizado o software *Waikato Environment for Knowledge Analysis (WEKA)* que é reconhecido pela comunidade científica como um sistema de referência em Aprendizado de Máquina (AM) e mineração de dados. O sistema é formado por um conjunto de implementações de algoritmos de diversas técnicas de Aprendizado de Máquina e foi implementado na linguagem de programação Java, tornando-o acessível nas principais plataformas computacionais (WITTEN; FRANK, 2005; HALL et al., 2009).

O WEKA inclui algoritmos de regressão, classificação, agrupamento, regras de associação e seleção de parâmetros (atributos). Atualmente está na versão 3.6.10, sendo organizado em três módulos de operação. O primeiro, "*simple Command Line Interface*" (CLI), a interação do usuário com WEKA ocorre através de linhas de comando. O "*Explorer*" é considerado o principal módulo, pois executa a interface gráfica para execução dos algoritmos de AM suportados pelo WEKA. E o terceiro, é o módulo "*Experimenter*" no qual o usuário, também por meio de interface gráfica, executa testes estatísticos em diferentes algoritmos simultaneamente a fim de avaliar os resultados obtidos.

Antes de utilizar o pacote WEKA, os dados devem ser convertidos para um dos formatos de arquivo suportados pelo WEKA. Neste trabalho o formato adotado é próprio do WEKA denominado de arff (Attribute-Relation File Format). O arquivo no formato arff é um arquivo ASCII composto de três partes.

Para desenvolvimento dos classificadores baseados no KNN-DTW e HMM foram desenvolvidos scripts próprios utilizando o Java.

### 3.2.1 Configurações específicas para os experimentos com o algoritmo KNN-DTW

Os experimentos para o classificador KNN-DTW, como já mencionado, trabalha diretamente com as amostras do sinal, sem passar por um processo de extração de características e portanto, não utiliza *front ends*. Para o desenvolvimento do classificador as seguintes etapas foram realizadas:

- Foi desenvolvido um script em linguagem Java para formatar os arquivos originais da base de dados UFPAFaults em novos arquivos com extensão "txt". Neste processo as faltas ABCT da base de dados foram rotuladas para ABC. O arquivo gerado foi utilizado para dar entrada direta no classificador KNN-DTW.
- Foi desenvolvido um algoritmo para implementação da técnica KNN-DTW em linguagem Java conforme as especificações do algoritmo disposta em (PETITJEAN et al., 2016) no software NetBeans, Versão 8.2 sob Licença GPL.
- Foram executados os experimentos para desenvolvimento do classificador KNN-DTW para classificação de faltas do tipo curto-circuito em linhas de transmissão de energia utilizando o software implementado na etapa anterior.

### 3.2.2 Configurações específicas para os experimentos com a arquitetura FBSC

Os experimentos com a arquitetura FBSC, utiliza *front ends* para extrair características da base de dados UFPAFaults que servirão como entradas para os classificadores convencionais.

Portanto, depois de definido as configurações gerais dos experimentos o estudo seguiu um fluxo específico para a arquitetura FBSC com os seguintes passos:

- Desenvolvimento dos *front ends* Raw, RMS, Waveletconcat, Waveletenergy e ConcatFrontEnd para extração de características da base UFPAFaults (arquivo ASCII) o que gerou matrizes de tamanho fixo (novos arquivos com extensão txt);
- Desenvolvimento de um script em linguagem Java para formatar os arquivos da base de dados com extensão txt, gerados por cada *front end*, para padrão do software WEKA, no qual gerou novos arquivos com extensão arff. Vale ressaltar que esse processo também rotulou as amostras das faltas ABCT da base de dados UFPAFaults para faltas ABC;
- Implementação de um script em Java para fazer a seleção automática do modelo para melhores ajustes nos parâmetros dos classificadores utilizados no estudo;
- Realização da réplica dos experimentos de (HOMCI et al., 2016) utilizando os classificadores convencionais RNA, SVM-RBF, KNN e RF para classificação de faltas do tipo curto-circuito em linhas de transmissão de energia utilizando o software WEKA, sob licença GPL.

### 3.2.3 Configurações específicas para os experimentos com o algoritmo HMM

Os experimentos para o classificador HMM, também como o KNN-DTW, trabalha diretamente com amostras diretas da base de dados, sem necessidade de um *front-end*. O estudo seguiu um fluxo específico para o HMM com as seguintes etapas:

- Desenvolvimento de um script em linguagem Java para formatar os arquivos originais da base de dados UFPAFaults em novos arquivos com extensão "txt". Neste processo as faltas ABCT da base de dados foram rotuladas para ABC . O arquivo gerado foi utilizado para dar entrada direta no classificador HMM;
- Implementação do algoritmo HMM em linguagem R no software R STUDIO, versão 1.1.442 sob licença GPL, que foi devidamente testado e validado conforme as especificações dispostas em (FRONDANA, 2012).
- Realização dos experimentos para desenvolvimento do classificador HMM para classificação de faltas do tipo curto-circuito em linhas de transmissão de energia utilizando o software implementado na etapa anterior.
- Aplicação de testes de significância estatísticas nos resultados obtidos para o classificadores desenvolvidos usando KNN-DTW, HMM e os classificadores convencionais da arquitetura FBSC para provar a igualdade ou diferença de seus desempenhos.

## 4 RESULTADOS E ANÁLISES

Este capítulo apresenta os resultados e análises dos experimentos realizados para desenvolvimento dos classificadores propostos neste trabalho para um cenário off-line.

### 4.1 Resultados dos Experimentos do algoritmo KNN-DTW

Nesta seção serão apresentados os resultados dos experimentos realizados para desenvolvimento do classificador baseado no KNN-DTW. A Tabela 3 apresenta os resultados de todos os experimentos realizados variando-se o parâmetro K (quantidade de vizinhos) do KNN-DTW entre os valores de 1 a 9, considerando apenas valores ímpares.

**Tabela 3 – Taxa de erro (%) do KNN-DTW para a base de teste, de acordo com número de vizinhos e o número de amostras de treino.**

K-Vizinhos/Nº-Amostras Treino	100	200	300	400	500	600	700	800	900	1000
1	71,3%	33,6	14,9	15,1	12,6	14,5	12,9	11,6	<b>8,7</b>	9,1
3	73,1%	33,9	15,7	15,3	14,3	15,3	13,7	15,2	11,2	10,7
5	74,6%	35,1	17,2	18,4	15,1	17,1	14,6	16,7	10,3	11,1
7	77,0%	36,4	18,3	17,3	17,2	17,3	14,7	17,5	11,5	11,5
9	71,8%	34,5	16,1	18,8	18,7	18,0	14,1	14,2	11,7	12,3

Observa-se na Tabela 3 que o classificador KNN-DTW teve o seu melhor desempenho no parâmetro K=1 e no conjunto de treinamento de 900 amostras. É possível perceber através dos resultados obtidos para todos os experimentos a importância de uma base de dados representativa das faltas a serem classificadas para o caso do KNN-DTW. A medida que se aumentou o número de padrões de treino, a taxa de erro em cima da base de teste diminuiu consideravelmente.

Considerando o custo computacional utilizando o KNN-DTW, o conjunto de treino inicial de 100 amostras apresentou um tempo de processamento próximo de 2 horas, sendo que a medida que as amostras de treino foram aumentando o tempo de processamento esboçou um comportamento próximo a um crescimento linear, atingindo um valor um pouco inferior a 30 horas para o classificador com um conjunto de 1000 amostras. A Figura 27 ilustra o tempo de execução dos experimentos do algoritmo KNN-DTW em cada base de treino.

### 4.2 Resultados dos Experimentos da arquitetura FBSC

Esta seção apresenta os resultados dos experimentos replicados usando a arquitetura FBSC e o conjunto de dados da base UFPAFaults. Os front ends adotados foram Waveletconcat, Waveletenergy, Raw, RMS e ConcatFrontEnd. Enquanto que os classificadores convencionais empregados foram RNA, RF, KNN e SVM, todos do conjunto de pacotes do software WEKA.

A Tabela 4 apresenta a variação dos valores de parâmetros usados nos experimentos para cada tipo de classificador convencional utilizado.

Figura 27 – Tempo dos experimentos do algoritmo KNN-DTW.

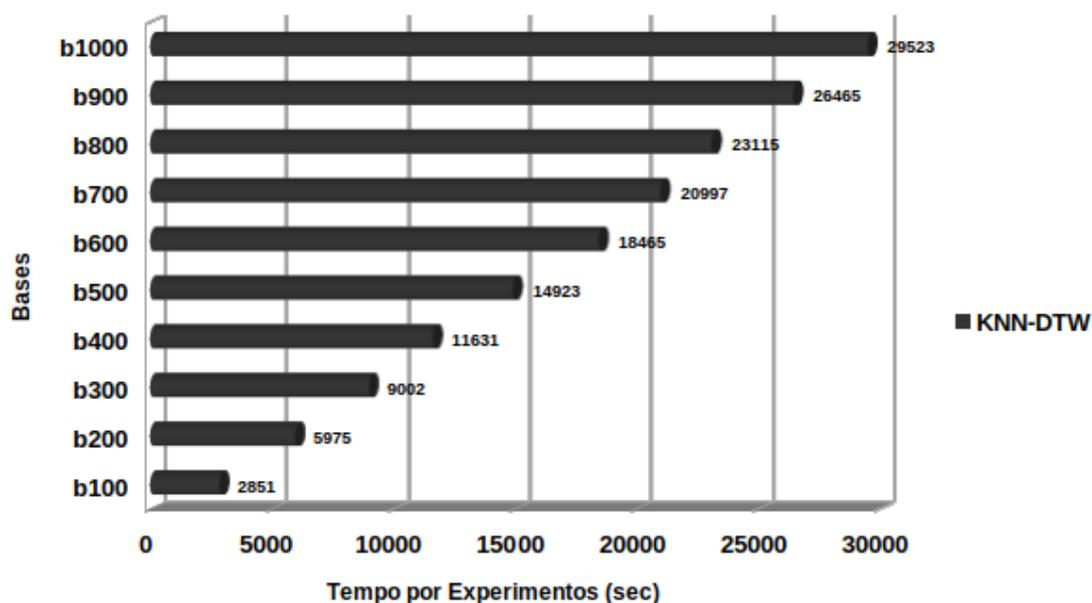


Tabela 4 – Resultado do Grid de Seleção do Modelo dos Classificadores Convencionais.

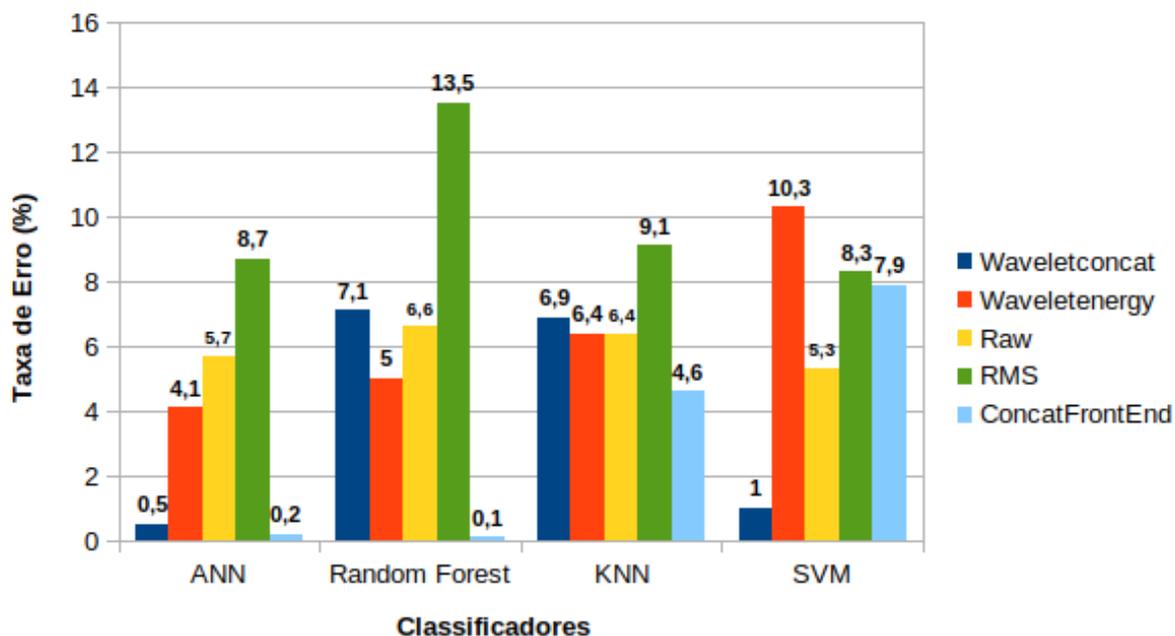
Classificador	Parâmetros	Valores Grid
RNA	H	10, 20, 40, 80 e <b>160</b>
	L	0,1, 0,5, e <b>0,9</b>
	M	0,1, 0,2 e <b>0,4</b>
RF	I	<b>100</b> , 200, 300, ..., and 1000
SVM	G	0.1, 1, <b>100</b>
	C	<b>0.1</b> , 1, 10 e 100
KNN	K	<b>1</b> , 3, 5, 7 e 9

Como pode se observar na Tabela 4 as replicações dos experimentos realizados para cada classificador teve uma quantidade razoável de variação nos valores de seus parâmetros até atingir um valor ótimo. Para RNA no parâmetro *H*, 160 neurônios na camada oculta foi o necessário para melhor convergência da rede. Por outro lado, RF precisou de 100 árvores em *I* para obter melhor performance. Enquanto que SVM utilizou 100 e 0,01 em *G* e *C* respectivamente para chegar ao mesmo objetivo. E por fim no KNN, um vizinho mais próximo no parâmetro *K* foi suficiente para atingir melhor valor de classificação .

A Figura 28 apresenta os melhores resultados obtidos na base de teste considerando todos front ends e classificadores convencionais utilizados na arquitetura FBSC.

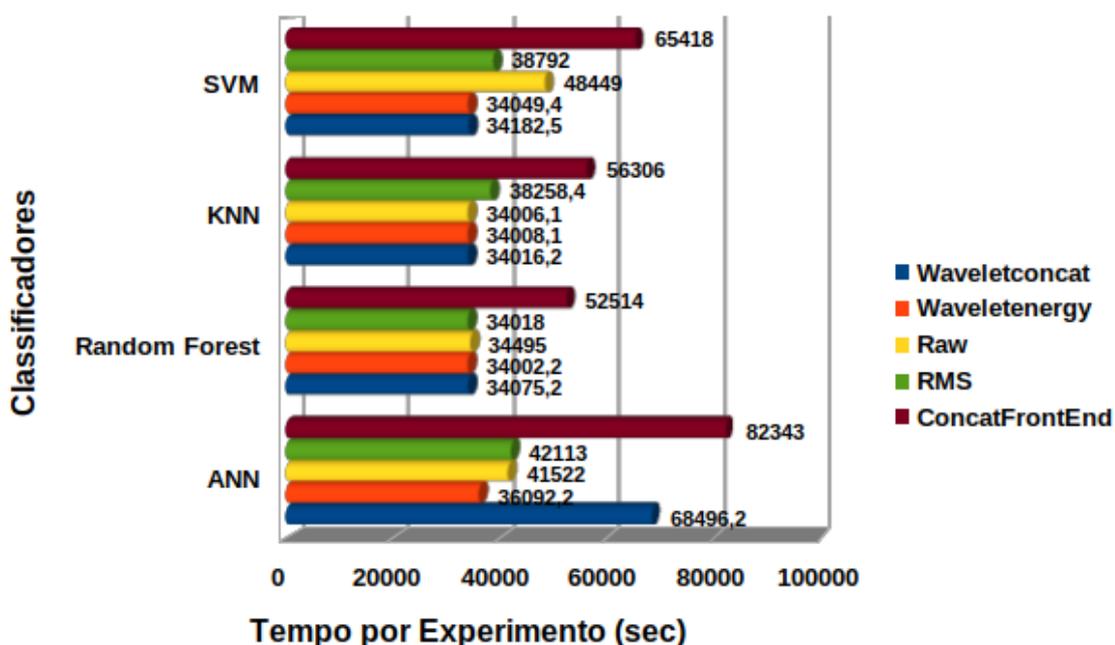
Outro fator avaliado no estudo para arquitetura FBSC foi o custo computacional na replicação dos experimentos. A classificação nessa arquitetura requer um tempo elevado na execução dos experimentos. A Figura 29 exibe o tempo de execução em segundos para todos algoritmos convencionais associados a seus respectivos front ends. O tempo de execução consi-

Figura 28 – Melhores Resultados dos Experimentos de Classificação da Arquitetura FBSC.



dera para cada arquitetura o treinamento para todos os casos de variação de amostras da base de treino.

Figura 29 – Tempo de execução para os classificadores FBSC.



Em conformidade com as observações da Tabela 11 o tempo de execução para o desenvolvimento dos classificadores baseados em FBSC é relativamente alto para classificação de

faltas em linhas de transmissão. O front end com custo computacional maior foi o ConcatFrontEnd e Waveletconcat associados ao classificador convencional RNA. Isso certamente deve-se ao fato de os mesmos possuírem características em conjunto de outros front ends. Enquanto que Waveletenergy, Raw e RMS possuem estruturas mais simples, e apresentam-se com custo computacional inferior associados a maioria dos classificadores.

### 4.3 Resultado dos Experimentos do algoritmo HMM

Nesta seção são apresentados os resultados dos experimentos do algoritmo HMM utilizando o conjunto de dados da base UFPAFaults. Para os processos de treinamento e testes foi utilizado o algoritmo de Baum-Welch. A Tabela 5 exibe os valores de cada parâmetro do algoritmo HMM empregados no estudo. A tabela também apresenta o resultado da taxa de erro na base de teste em cada experimento por base de dados de treino, e a média geral da taxa de erro entre todos os resultados.

De acordo com a Tabela 5 observa-se que o algoritmo HMM teve um bom desempenho considerando todas as bases de treino, sendo que o melhor resultado ocorreu para a base de treino de 500 amostras onde obteve uma taxa de erro de 0,02, sendo que a probabilidade de estado para esta base variou entre 0,47 e 0,72, a probabilidade de transição ficou entre 0,21 à 0,68, a média na faixa de -0,36 à -0,43, e a variância entre 0,18 à 0,32. Esse resultado indica que o balanceamento e quantidade de amostras dispostas para cada classe nesta base pode ter influenciado no resultado obtido. Estas características também são percebidas nas outras bases que em média teve um desempenho próximo ou igual a base de 500 amostras.

Os resultados dos experimentos contidos na Tabela 5 também indicam algumas vantagens do algoritmo HMM para classificação de faltas em linhas de transmissão de energia onde percebe-se sua capacidade diferenciada de reconhecer padrões de forma mais precisa em relação aos outros algoritmos testados. Além da capacidade de tratar diretamente informações sem emprego de *front ends*, a sua característica probabilística favorece os treinamentos quando se considera a quantidade de dados, tendo obtido bons resultados para todas as bases de dados utilizadas, obtendo resultados satisfatórios mesmo para uma base de dados pequena para treino (100).

Além do desempenho geral do classificador, outro fator avaliado no estudo foi o desempenho do classificador considerando cada classe nos experimentos realizados, onde foi possível obter uma ideia mais precisa do aprendizado do algoritmo HMM na classificação de faltas. A Tabela 6 apresenta a máxima verossimilhança ( $M_v$ ) obtida, o resultado da menor taxa de erro das classes entre os experimentos ( $E_i$ ), a média da taxa de erro da classe entre os experimentos ( $AE_i$ ) e a média geral da taxa de erro dos experimentos de cada classe entre todos os conjuntos de dados ( $G_{Ac}$ ).

Tabela 5 – Valores Utilizados dos Parâmetros do algoritmo HMM, Taxa de Erro (na base de teste) e Média da Taxa de Erro para as bases de treino de 100 à 1000.

Bases											
Parâmetros/Erro	Exp	100	200	300	400	500	600	700	800	900	1000
Probabilidade de Estado	1	0.45	0.27	0.38	0.42	0.57	0.38	0.54	0.48	0.49	0.48
	2	0.61	0.75	0.36	0.68	0.72	0.66	0.86	0.36	0.56	0.77
	3	0.36	0.41	0.86	0.20	0.71	0.36	0.19	0.53	0.52	0.53
	4	0.22	0.09	0.53	0.24	0.48	0.56	0.84	0.62	0.52	0.53
	5	0.54	0.85	0.36	0.44	0.47	0.64	0.82	0.38	0.64	0.54
Probabilidade de Transição	1	0.39	0.17	0.22	0.03	0.31	0.41	0.35	0.20	0.06	0.56
	2	0.63	0.72	0.24	0.61	0.68	0.68	0.91	0.83	0.56	0.70
	3	0.64	0.12	0.87	0.01	0.61	0.14	0.09	0.70	0.31	0.68
	4	0.36	0.52	0.48	0.18	0.21	0.42	0.84	0.41	0.36	0.48
	5	0.25	0.88	0.003	0.01	0.52	0.45	0.82	0.03	0.65	0.59
Média ( $\mu$ )	1	-1.47	-1.55	-0.99	-1.09	-0.36	-1.42	-1.38	-0.56	-0.94	-0.90
	2	-0.95	-1.50	-1.23	-0.80	-0.64	-0.03	-0.61	-0.23	-1.46	-0.25
	3	-0.97	-1.44	-0.26	-0.06	-1.23	-0.11	-1.08	-1.13	-1.21	-1.52
	4	-1.30	-1.12	-0.07	-0.96	-1.27	-0.25	-1.57	-1.47	-0.42	-0.80
	5	-0.71	-0.23	-0.56	-1.17	-1.43	-1.02	-1.28	-0.79	-0.79	-0.06
Variância( $\lambda$ )	1	0.34	0.48	0.43	0.46	0.32	0.43	0.58	0.33	0.46	0.32
	2	0.10	0.46	0.52	0.31	0.41	0.08	0.23	0.56	0.62	0.19
	3	0.08	0.17	0.04	0.43	0.18	0.18	0.45	0.48	0.53	0.09
	4	0.52	0.10	0.52	0.57	0.32	0.09	0.20	0.56	0.27	0.34
	5	0.29	0.19	0.17	0.24	0.27	0.47	0.05	0.42	0.38	0.34
Taxa de Erro	1	0.03	0.04	0.03	0.05	<b>0.02</b>	0.04	0.04	0.04	0.03	0.04
	2	0.03	0.02	0.03	0.04	<b>0.04</b>	0.03	0.05	0.04	0.03	0.04
	3	0.03	0.04	0.04	0.04	<b>0.03</b>	0.04	0.03	0.04	0.04	0.04
	4	0.03	0.03	0.03	0.05	<b>0.02</b>	0.04	0.03	0.04	0.03	0.05
	5	0.04	0.02	0.03	0.05	<b>0.02</b>	0.04	0.03	0.04	0.03	0.04
Média da Taxa de Erro	=	0.03	0.03	0.03	0.04	<b>0.03</b>	0.04	0.04	0.04	0.03	0.04

Como pode ser observado na Tabela 6 o desempenho da classificação de faltas entre as bases de dados utilizando o algoritmo HMM teve destaque para as classes BC e ABG que obtiveram em média a menor taxa de erro tendo sido obtido o valor de 0,02% para ambas classes. Para essas duas classes a máxima verossimilhança variou entre 13,16 à 16,42, e 13,16 à 16,14 respectivamente.

Em relação a classe AG, esta foi a que obteve em média a maior taxa de erro (0,06%) com máxima verossimilhança variando entre 12,63 à 16,80. As outras classes restantes obtiveram desempenho em relação a taxa de erro de 0,04% com máxima verossimilhança distintas das classes AG, BC e ABG. Percebe-se que semelhante aos resultados obtidos por base, a classificação

Tabela 6 –  $E_i$ ,  $M_v$ ,  $AE_i$  e  $G_{Ac}$  nas bases de 100 à 1000.

Bases												
Class	=	100	200	300	400	500	600	700	800	900	1000	$G_{Ac}$
AG	$E_i$	0,00	0,00	0,00	0,12	0,00	0,02	0,00	0,07	0,03	0,06	0,06
	$AE_i$	0,04	0,03	0,02	0,13	0,01	0,07	0,02	0,09	0,04	0,13	
	$M_v$	16,80	12,63	14,48	13,80	16,27	16,77	15,42	16,46	14,15	16,30	
BG	$E_i$	0,06	0,00	0,00	0,07	0,00	0,00	0,01	0,00	0,00	0,00	0,03
	$AE_i$	0,07	0,05	0,04	0,06	0,00	0,01	0,08	0,01	0,00	0,01	
	$M_v$	16,23	16,44	15,82	13,61	14,97	16,23	15,67	16,43	14,66	14,61	
CG	$E_i$	0,06	0,00	0,08	0,06	0,00	0,08	0,06	0,06	0,00	0,06	0,04
	$AE_i$	0,03	0,01	0,00	0,06	0,02	0,06	0,06	0,07	0,00	0,06	
	$M_v$	16,89	17,11	14,06	14,08	14,66	14,64	16,56	16,80	14,83	14,47	
AB	$E_i$	0,04	0,06	0,00	0,01	0,08	0,01	0,05	0,06	0,04	0,02	0,04
	$AE_i$	0,03	0,01	0,00	0,06	0,02	0,06	0,06	0,07	0,00	0,06	
	$M_v$	14,12	15,25	14,25	16,93	13,92	15,63	16,02	14,59	15,62	15,45	
AC	$E_i$	0,01	0,01	0,00	0,05	0,04	0,02	0,02	0,05	0,06	0,00	0,04
	$AE_i$	0,01	0,02	0,07	0,05	0,04	0,05	0,02	0,07	0,06	0,01	
	$M_v$	14,12	15,25	14,25	16,93	13,92	15,63	16,02	14,59	15,62	15,45	
BC	$E_i$	0,00	0,01	0,06	0,02	0,00	0,00	0,00	0,00	0,00	0,00	<b>0,02</b>
	$AE_i$	0,01	0,03	0,05	0,04	0,01	0,01	0,02	0,02	0,01	0,01	
	$M_v$	14,23	16,42	14,97	15,10	15,85	15,19	13,16	15,63	16,32	14,71	
ABC	$E_i$	0,02	0,00	0,06	0,00	0,00	0,03	0,00	0,05	0,06	0,06	0,04
	$AE_i$	0,04	0,01	0,04	0,00	0,02	0,05	0,01	0,05	0,06	0,07	
	$M_v$	13,97	14,27	13,99	15,95	14,86	14,63	16,84	15,48	14,63	14,67	
ABG	$E_i$	0,00	0,00	0,01	0,02	0,00	0,01	0,06	0,00	0,00	0,01	<b>0,02</b>
	$AE_i$	0,03	0,01	0,02	0,03	0,00	0,03	0,02	0,01	0,02	0,01	
	$M_v$	16,23	16,44	15,82	13,61	14,97	16,23	15,67	16,43	14,66	14,61	
ACG	$E_i$	0,06	0,08	0,00	0,05	0,03	0,00	0,05	0,00	0,07	0,01	0,04
	$AE_i$	0,07	0,06	0,00	0,04	0,05	0,00	0,07	0,00	0,07	0,02	
	$M_v$	14,01	16,18	12,51	17,81	15,69	14,00	14,89	15,18	18,09	17,80	
BCG	$E_i$	0,01	0,00	0,002	0,02	0,06	0,08	0,02	0,06	0,06	0,06	0,04
	$AE_i$	0,03	0,01	0,00	0,04	0,05	0,07	0,02	0,05	0,06	0,04	
	$M_v$	16,30	15,73	13,33	16,77	13,81	14,53	15,25	15,47	16,16	16,95	

por classe indica também que o balanceamento e quantidade de amostras disposta para cada classe nesses conjuntos de dados pode ter influenciado nos resultados de classificação.

Assim como nas outras arquiteturas testadas também foi observado o custo computacional para o algoritmo HMM nos experimentos realizados para classificação de faltas. Devido o algoritmo HMM tratar diretamente as amostras de uma série temporal, isto favoreceu para diminuição da quantidade de operações, influenciando para um menor custo computacional. A Tabela 7 apresenta o tempo dos experimentos para o caso do classificador HMM.

De acordo com a Tabela 7 observa-se que algoritmo HMM apresenta-se com um custo computacional bastante aceitável, onde carrega somente uma vez todas as bases de dados com

**Tabela 7 – Tempo de execução para o classificador HMM.**

<b>Base</b>	<b>Tempo por Experimento (sec)</b>
Carregamento das Bases	274
100	233
200	256
300	278
400	237
500	322
600	349
700	371
800	389
900	416
1000	435

tempo de 274 segundos. E posteriormente vai realizando os experimentos em cada base com quantidade de amostras diferentes. Todo o processo leva um tempo total de 3564 segundos para realizar todos os experimentos para desenvolvimento dos classificadores..

## 4.4 Comparação dos Resultados entre a arquitetura FBSC, o algoritmo KNN-DTW e o algoritmo HMM

A Tabela 8 apresenta os melhores resultados da taxa de erro dos algoritmos HMM, KNN-DTW e das arquiteturas FBSC. Como pode ser visto, dos resultados da taxa de erro, o algoritmo HMM apresenta no geral um desempenho superior, com taxa de erro de 0,03%. Logo após o HMM, os classificadores com estrutura FBSC com front-end concatFrontEnd e classificador RNA e RF foram os que obtiveram a menor taxa de erro. Todos os outros classificadores apresentaram taxa de erro bem acima destes 3 modelos.

Em relação ao custo computacional a Tabela 8 também exibe o tempo de processamento para desenvolvimento dos classificadores testados. Observa-se que o algoritmo HMM necessitou de 3564 segundos para realizar os experimentos considerando todas as bases de dados de treino testadas. Este tempo é consideravelmente menor do que o tempo utilizado para desenvolvimento dos outros classificadores. O KNN-DTW foi entre todos algoritmos o com maior tempo de processamento.

**Tabela 8 – Melhores Resultados dos Experimentos de Classificação e o Tempo de Execução em Segundos do Custo Computacional da arquitetura FBSC, KNN-DTW e HMM.**

Classificador	Front End	Taxa de Erro (%)	Tempo por Experimento (sec)
RNA	Waveletconcat	0,5	68496
	Waveletenergy	4,1	36092
	Raw	5,7	41522
	RMS	8,7	42113
	concatFrontEnd	<b>0.2</b>	82343
RF	Waveletconcat	7,1	34075
	Waveletenergy	5,0	34049
	Raw	6,6	48449
	RMS	13,5	38792
	concatFrontEnd	<b>0.1</b>	65418
SVM	Waveletconcat	1,0	34182
	Waveletenergy	10.3	34049
	Raw	5,3	48449
	RMS	8,3	38792
	concatFrontEnd	7.9	65418
KNN	Waveletconcat	6,9	34016
	Waveletenergy	6,4	34008
	Raw	6,4	34006
	RMS	9,1	38258
	concatFrontEnd	4,6	56306
KNN-DTW	—————	<b>8,7</b>	96295
HMM	—————	<b>0.03</b>	<b>3564</b>

Para testar a semelhança entre os algoritmos propostos foi aplicado o teste de significância estatística *t-student*. A Tabela 9 apresenta o resultado da aplicação do teste entre os

algoritmos HMM e KNN-DTW e os classificadores convencionais da arquitetura FBSC associados ao *front end* concatFrontEnd. O teste estatístico teve como valor tabulado  $t_{tab}$  igual a 2,31 e grau de liberdade condicionado a referência 8 após a aplicação do cálculo da Equação 2.28 com significância de  $\alpha = 5\%$ , no qual corresponde a comparação entre dois classificadores com cinco experimentos para cada um. A diferença da taxa de erro entre HMM e RNA ficou em 0,17%, HMM e KNN em 4,70%, HMM e RF em 0,07%, HMM e SVM em 7,87%, HMM e KNN-DTW em 8,67%, KNN-DTW e RNA em 8,5%, KNN-DTW e KNN em 4,1%, KNN-DTW e RF em 8,6% e KNN-DTW e SVM em 0,8%. Então para esse cenário o HMM, RNA e RF possuem estatisticamente desempenho semelhantes, apesar de valores diferentes em seus resultados.

**Tabela 9 – Resultados da Comparação de Teste Estatístico entre a arquitetura FBSC com concat-FrontEnd associada aos Classificadores RNA, SVN, KNN e RF e os algoritmos HMM e KNN-DTW com Significância de  $\alpha = 5\%$ .**

Classificador	Graus de Liberdade	$T_{cal}$	$T_{tab}$	t-Student	Média
KNN-DTW->RNA	8	8,5	2.31	$T_{cal} > T_{tab}$	Diferente
KNN-DTW->KNN	8	4.1	2.31	$T_{cal} > T_{tab}$	Diferente
KNN-DTW->RF	8	8,6	2.31	$T_{cal} > T_{tab}$	Diferente
KNN-DTW->SVM	8	0,8	2.31	$T_{cal} < T_{tab}$	Igual
HMM->RNA	8	0.17	2.31	$T_{cal} < T_{tab}$	Igual
HMM->KNN	8	4.70	2.31	$T_{cal} > T_{tab}$	Diferente
HMM->RF	8	0.07	2.31	$T_{cal} < T_{tab}$	Igual
HMM->SVM	8	7.87	2.31	$T_{cal} > T_{tab}$	Diferente
HMM->KNN-DTW	8	8,67	2.31	$T_{cal} > T_{tab}$	Diferente

Dos resultados obtidos tem-se então que considerando tanto o valor da taxa de erro quanto o custo computacional, entre todos os classificadores testados, o classificador baseado no HMM foi o que demonstrou maior potencial para o problema de classificação de faltas do tipo curto-circuito em linhas de transmissão.

## 5 CONSIDERAÇÕES FINAIS

Ao longo deste trabalho investigou-se uma importante classe de eventos que compromete a QEE fornecida pelos SEPs: as faltas do tipo curto-circuito em linhas de transmissão de energia. A classificação dessas faltas em um cenário *off-line*, isto é, em um cenário pós-falta é caracterizada como um problema de classificação de sequências, uma tarefa de classificação especial onde os dados de entrada são representados por uma matriz que possui tamanho variável. Neste contexto, duas abordagens para classificação de faltas foram apresentadas e testadas: uma utilizando o algoritmo K-vizinho mais próximo tendo como medida de similaridade o alinhamento temporal dinâmico (DTW) e outra utilizando HMM.

Para comparação de resultados foram também desenvolvidos e testados alguns classificadores baseados na arquitetura FBSC apresentada em (HOMCI et al., 2016). A arquitetura utiliza algoritmos convencionais, ou seja, que lidam com entradas de tamanho fixo tais como redes neurais e máquinas de vetores de suporte na tarefa de classificação de faltas. Para isso adota uma etapa de pré-processamento ou *front end* que converte ou organiza as amostras das sequências de tamanho variável em segmentos (quadros) de tamanho fixo que podem ser processados pelos classificadores convencionais. Acontece que esta arquitetura de classificação, conforme visto neste trabalho, possui elevado grau de parametrização tanto dos front ends quanto dos algoritmos de AM, tornando o processo de escolha do melhor modelo um procedimento custoso e não trivial.

Nesse contexto, como alternativa à arquitetura FBSC, foi proposto para classificação de faltas os algoritmos HMM e KNN-DTW, que diferente da arquitetura FBSC consegue lidar diretamente com amostras de tamanho diferentes, tendo como principal característica a baixa parametrização para encontrar seu melhor modelo.

Nos experimentos realizados foi utilizada a base de dados UFPAFaults composta por simulações, baseada em um modelo de SEP real de faltas do tipo curto-circuito em linhas de transmissão. Os classificadores convencionais adotados na arquitetura FBSC foram todos pertencentes ao pacote de mineração de dados WEKA e os *front ends* testados são os mais utilizados na literatura. O algoritmo HMM foi implementado em linguagem R no *software* R STUDIO e o algoritmo KNN-DTW em linguagem Java no *software* NetBeans, sendo esses *softwares* bem utilizados na comunidade científica.

O desempenho do algoritmo HMM em geral superou o desempenho dos classificadores baseados no algoritmo KNN-DTW e na arquitetura FBSC que utiliza os *front ends* Waveletconcat, Waveletenergy, Raw, RMS e concatFrontEnd associados aos classificadores convencionais RNN, RF, KNN, SVM. O algoritmo HMM obteve taxas de erro menores em relação ao algoritmo KNN-DTW e os classificadores convencionais.

Com a aplicação do teste de significância estatística t-Student observou-se que os classificadores RNA e RF, associados ao *front end* concatFrontEnd, apresentam desempenho

equivalente ao algoritmo HMM considerando a significância  $\alpha = 5\%$ .

Em termos de custo computacional o algoritmo HMM é superior em cerca de 90% no seu tempo de processamento em relação a todos classificadores da arquitetura FBSC e 99,6% ao algoritmo KNN-DTW nas mesmas condições, se configurando como um classificador potencial na classificação de faltas em linhas de transmissão.

Como trabalhos futuros podemos propor a avaliação de outros tipos de faltas em sistemas de potência com o uso do algoritmo HMM e também o desenvolvimento de sistemas classificadores utilizando bases de dados reais de faltas em linhas de transmissão.

## REFERÊNCIAS

- AGUILERA, C.; ORDUNA, E.; RATTA, G. Fault detection, classification and faulted phase selection approach based on high-frequency voltage signals applied to a series-compensated line. **Proceedings - Generation, Transmission and Distribution**, IEEE, v. 153, p. 469–475, 2006.
- ALMEIDA, A. R. et al. Ica feature extraction for the location and classification of faults in high-voltage transmission lines. **Electric Power Systems Research**, Science Direct, v. 1, n. 1, p. 254–263, 2017.
- AROUCHE J. C., G. C. A. R. H. M. S. M. B. F.; MORAIS, J. M. Transmission line fault classification using hidden markov models. **IEEE Access**, Science Direct, v. 7, p. 113499–113510, 2019.
- BAKKEN, D. E. et al. Smart generation and transmission with coherent, real-time data. **Proceedings of the IEEE**, IEEE, v. 99, n. 6, p. 928–951, 2011.
- BIOLOGY, C. Gentwarper: Mining of gene expression time series with dynamic time warping techniques. In: \_\_\_\_\_. Home Page Computacional Biology, 2019. Disponível em: <<https://www.psb.ugent.be/cbd/papers/gentxwarper/index.htm>>.
- BISWAS K. KUMAR, A. G. S.; NAYAK, P. K. Fault detection and classification for tesc compensated transmission lines using wavelet energy. **4th International Conference on Recent Advances in Information Technology (RAIT)**, IEEE Xplore, 2018.
- BOLLEN, M. H. et al. Bridging the gap between signal and power. **IEEE SIGNAL PROCESSING MAGAZINE**, 2009.
- BRAGA, A. P.; CARVALHO, A. C. P. L.; LUDERMIR, T. B. N. **Redes Artificiais: teoria e aplicações**. [S.l.]: LTC - Livros Técnicos e Científico, 2000. v. 2.
- CAPPÉ, E. M. O.; RYDÉN, T. Inference in hidden markov models. **Springer Series in Statistics**, Springer, 2005.
- CARIRI, B. Chesf aumenta confiabilidade de atendimento para os estados do ceará e rio grande do norte. In: \_\_\_\_\_. Home Page Blog Carir, 2019. Disponível em: <<https://www.blogcariri.com.br/2012/11/chesf-aumenta-confiabilidade-de.html>>.
- CARMEM, T. H. Algoritmos –teoria e prática. **Campus Editora**, Campus, 2002.
- CHATTERJEEA, B.; DEBNATHB, S. Cross correlation aided fuzzy based relaying scheme for fault classification in transmission lines. **Elsevier Engineering Science and Technology**, Science Direct, 2019.
- COSTA, B. G. et al. Fault classification on transmission lines using knn-dtw. **17th International Conference on Computational Science and Its Applications (ICCSA 2017)**, Springer, v. 1, p. 174–187, 2017.
- DANCEY, C. P.; REIDY, J. **Estatística sem Matemática para Psicologia: usando SPSS para Windows**. [S.l.]: Artmed, 2006.
- FACELI, K. **Inteligência artificial: uma abordagem de aprendizado de máquina**. [S.l.]: Grupo Gen-LTC, 2011.

- FARSHAD, M. Detection and classification of internal faults in bipolar hvdc transmission lines based on k-means data description method. **Electrical Power and Energy**, Elsevier, 2019.
- FATHABADI, H. Novel filter based ann approach for short-circuit faults detection, classification and location in power transmission lines. **International Journal of Electrical Power Energy Systems**, Science Direct, v. 74, n. Supplement, p. 374–383, 2016.
- FAULKENBERRY, L. M.; COFFER, W. Electrical power distribution and transmission. **IEEE Press Series on Power Engineering**, 1996.
- FONTES, C. Pattern recognition in multivariate time series – a case study applied to fault detection in a gas turbine. **Engineering Applications of Artificial Intelligence**, v. 1, n. 1, 2015.
- FRONDANA, I. M. **Identificação de Biopotenciais via Cadeias de Markov Ocultas**. Dissertação (Mestrado) — Universidade de Brasília, 2012.
- GIORGINO, T. Computing and visualizing dynamic time warping alignments in r. **Journal of Statistical Software, Articles**, v. 31, n. 7, p. 1–24, 2009.
- HALL, M. et al. The weka data mining software: an update. **ACM SIGKDD explorations newsletter**, ACM, v. 11, n. 1, p. 10–18, 2009.
- HAYKIN, S. S. et al. **Neural networks and learning machines**. [S.l.]: Pearson Education Upper Saddle River, 2009. v. 3.
- HOMCI, M. et al. A new strategy based on feature selection for fault classification in transmission lines. **15th Ibero-American Conference on AI (IBERAMIA-2016)**, Springer, v. 2, p. 376–387, Nov 2016.
- JU, Y. W. W.; ROSSO, A. D. Average wavelet energy-based method for fault classification in transmission lines. **IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)**, IEEE Xplore, 2019.
- KEOGH, E.; RATANAMAHATANA, C. A. Exact indexing of dynamic time warping. **Knowledge and Information Systems**, v. 7, n. 3, p. 358–386, 2005.
- KEZUNOVIC, M.; RIKALO, I. Detect and classify faults using neural nets. **Computer Applications in Power, IEEE**, IEEE, v. 9, n. 4, p. 42–47, 1996.
- LAWRANCE, R.; RABINER, A. Tutorial on hidden markov models and selected application in speech recognition. **Proc. Of IEEE**, IEEE Xplore, v. 77, p. 257–285, 1989.
- MARMARELIS, V. Z. Appendix ii: Gaussian white noise. In: \_\_\_\_\_. **Nonlinear Dynamic Modeling of Physiological Systems**. John Wiley Sons, Inc., 2004. p. 499–501. ISBN 9780471679370. Disponível em: <<http://dx.doi.org/10.1002/9780471679370.app2>>.
- MOHAMMADI, H.; DEGHANI, M. Pmu based voltage security assessment of power systems exploiting principal component analysis and decision trees. **International Journal of Electrical Power & Energy Systems**, Elsevier, v. 64, p. 655–663, 2015.
- MORAIS, J. et al. A framework for evaluating automatic classification of underlying causes of disturbances and its application to short-circuit faults. **Power Delivery, IEEE Transactions on**, IEEE, v. 25, n. 4, p. 2083–2094, 2010.

MORAIS, J. et al. A framework for evaluating automatic classification of underlying causes of disturbances and its application to short-circuit faults. **IEEE Transactions on Power Delivery**, v. 25, n. 4, p. 2083–2094, Oct 2010.

MORAIS, J. M. **Avaliação de desempenho de classificadores de faltas em sistemas elétricos de potência**. Tese (Doutorado) — Universidade Federal do Pará, 2011.

OLESKOVICZ, M. **Aplicação de redes neurais artificiais na proteção de distância**. Tese (Doutorado) — Universidade São Paulo, 2001.

OLIVEIRA, A. R. **Redes Neurais Artificiais Aplicadas na Detecção, Classificação e Localização de Defeitos em Linhas de Transmissão**. Tese (Doutorado) — Universidade Federal de Juiz de Fora, 2005.

OSHIRO, T. **Uma abordagem para construção de uma única árvore a partir do random forest para classificação de bases de expressão gênica**. Tese (Doutorado) — Universidade de São Paulo, 2013.

PETITJEAN, F. et al. Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. **Knowledge and Information Systems**, v. 47, n. 1, p. 1–26, Apr 2016.

PIRES, Y. P. **Mineração de dados aplicada a Sistemas Elétricos: Classificação de faltas de curto-circuito em linhas de transmissão**. Tese (Doutorado) — Universidade Federal do Pará, 2009.

PRIDDY, K. L.; KELLER, P. E. The curse of dimensionality. **SPIE - The International Society for Optical Engineering, Artificial Neural Networks**, v. 1, p. 1–180, 2005.

RAMESH, N.; MOHAN, B. J. Fault classification in power systems using emd and svm. **Ain Shams Engineering Journal**, Science Direct, v. 8, n. 2, p. 103–111, 2017.

RAY D. P. MISHRA, K. D. P.; MISHRA, P. Fault detection and classification of a transmission line using discrete wavelet transform artificial neural network. **IEEE - International Conference on Information Technology (ICIT)**, IEEE Xplore, 2017.

SAHA I., M. J. Artificial neural networks in hardware: A survey of two decades of progress. **Neurocomputing**, 2010.

SAWATPIPAT P.; TAYJASANANT, T. Fault classification for thailand's transmission lines based on discrete wavelet transform. **ECTI-CON2010: The 2010 ECTI International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology**, v. 1, p. 636–640, 2010.

SCOTT, W.; LIU, T. Alternative transients program atp rule book. Canadian/American EMTP User Group, v. 1, 1992.

SCUDINO, P. A. **A Utilização de alguns testes estatísticos para análise da variabilidade do preço do Mel nos municípios Angra dos Reis e Mangaratiba, Estado do Rio de Janeiro**. Tese (Doutorado) — Universidade Federal Rural do Rio de Janeiro, 2008.

SHARMA O. P. MAHELA, M. K. G.; KUMAR, N. Detection and classification of transmission line faults using stockwell transform and rule based decision tree. **International Conference on Smart Electric Drives and Power System (ICSEDPS)**, IEEE Xplore, 2018.

- SILVA, D. F.; BATISTA, G. E. A. P. A. Speeding up all-pairwise dynamic time warping matrix calculation. **2016 SIAM International Conference on Data Mining**, v. 1, n. 1, p. 837–845, 2016.
- SILVA, K. M.; DANTAS, K.; SOUZA, B. A. Haar wavelet-based method for fast fault classification in transmission lines. **PES Transmission Distribution Conference and Exposition: Latin America**, IEEE, v. 1, p. 1–5, 2006.
- SILVA, K. M.; SOUZA, B. A.; BRITA, N. S. D. Fault detection and classification in transmission lines based on wavelet transform and ann. **Transactions on Power Delivery**, IEEE, v. 21, p. 2058–2063, 2006.
- SILVA K. M., S. B. A. B. N. S. D. D. K. M. C.; COSTA F. B., S. S. S. B. Detecção e classificação de faltas a partir da análise de registros oscilográficos via redes neurais artificiais e transformada wavelet. **Controle Automação Sociedade Brasileira de Automatica**, 2007.
- SINGH, M.; PANIGRAHI, B. K.; MAHESHWARI, R. P. Transmission line fault detection and classification. **2011 International Conference on Emerging Trends in Electrical and Computer Technology**, v. 1, p. 15–22, 2011.
- SOLAR. Em quais situações devemos utilizar um transformador nos sistemas de energia solar fotovoltaico? In: \_\_\_\_\_. Home Page Computacional Biology, 2019. Disponível em: <<https://vocesolar.com.br/em-quais-situacoes-devemos-utilizar-um-transformador-nos-sistemas-de-energia-solar-fotovoltaico/>>.
- SUDHA, G. A. Comparison between different approaches for fault classification in transmission lines. **Institution of Engineering and Technology**, IET Conference Proceedings, v. 5, p. 398–403, 2007.
- VAPNIK, V. **The nature of statistical learning theory**. [S.l.]: Springer Science & Business Media, 2013.
- VELOSO, L. V. T. e P. A. S. Algoritmos –teoria e prática. **Editora Sagra-Luzzato**, Sagra-Luzzato, 2002.
- VETTERLI, I.; KOVACEVIC, J. **Wavelets and subband coding**. [S.l.]: Prentice Hall, 1995.
- WITTEN, I. H.; FRANK, E. **Data Mining: Practical machine learning tools and techniques**. [S.l.]: Morgan Kaufmann, 2005.
- XI, X. et al. Fast time series classification using numerosity reduction. **Twenty-Third International Conference on Machine Learning**, v. 1, n. 1, p. 1033–1040, 2006.
- XING, Z.; PEI, J.; KEOGH, E. A brief survey on sequence classification. **ACM SIGKDD Explorations Newsletter**, ACM, v. 12, p. 40–48, 2010.
- YADAV, A.; DASH, Y. An overview of transmission line protection by artificial neural network: Fault detection, fault classification, fault location, and fault direction discrimination. Chao-Ton Su, Article ID 230382, 20 pages, 2014.
- YURTMAN, A.; BARSHAN, B. Detection and evaluation of physical therapy exercises by dynamic time warping using wearable motion sensor units. **Information Sciences and Systems**, 2013.

ZHANG, N.; KEZUNOVIC, M. A real time fault analysis tool for monitoring operation of transmission line protective relay. **Electric Power Systems Research**, Electric Power Systems Research, v. 7, p. 361–370, 2007.

ZHONGMIN, Y.; BIN, W. A review on induction motor online fault diagnosis. **Proceedings IPEMC 2000**, Third International Power Electronics and Motion Control Conference (IEEE Cat. No.00EX435, v. 3, p. 1353–1358, 2000.

ZHOU, Q. et al. Structure damage detection based on random forest recursive feature elimination. **Mechanical Systems and Signal Processing**, Elsevier, v. 46, n. 1, p. 82–90, 2014.

## A APÊNDICE A

Os códigos dos algoritmos HMM e KNN-DTW, script shell (Construção da base) e a base de dados UFPAFaults encontram-se disponibilizados no endereço eletrônico:

<https://github.com/jeanarouche/HMM-KNN-DTW-FaultClassification>.

Isso é devido os mesmos possuírem uma grande extensão de codificação e também muitos arquivos de implementação dos algoritmos HMM e KNN-DTW. O repositório está composto com a estrutura descrita abaixo:

- Base de dados UFPAFaults com 11 arquivos de extensão ".txt";
- Código do algoritmo HMM com 1 arquivo de extensão ".R";
- Código do algoritmo KNN-DTW com 1 arquivo compactado de extensão ".ZIP";
- Código da construção das bases em script shell com 1 arquivo de extensão ".SH";

## B APÊNDICE B

Tabela 10 – Melhores Resultados dos Experimentos de Classificação da Arquitetura FBSC

Classificador	Front End	Taxa de Erro (%)
RNA	Waveletconcat	0.5
	Waveletenergy	4.1
	Raw	5.7
	RMS	8.7
	concatFrontEnd	<b>0.2</b>
RF	Waveletconcat	7.1
	Waveletenergy	5.0
	Raw	6.6
	RMS	13.5
	concatFrontEnd	<b>0.1</b>
SVM	Waveletconcat	1.0
	Waveletenergy	10.3
	Raw	5.3
	RMS	8.3
	concatFrontEnd	7.9
KNN	Waveletconcat	6.9
	Waveletenergy	6.4
	Raw	6.4
	RMS	9.1
	concatFrontEnd	4.6

## C APÊNDICE C

Tabela 11 – Tempo de execução para os classificadores FBSC.

<b>Classificador</b>	<b>Front End</b>	<b>Tempo por Experimento (sec)</b>
RNA	Waveletconcat	68496
	Waveletenergy	36092
	Raw	41522
	RMS	42113
	concatFrontEnd	82343
RF	Waveletconcat	34075
	Waveletenergy	34049
	Raw	48449
	RMS	38792
	concatFrontEnd	65418
SVM	Waveletconcat	34182
	Waveletenergy	34049
	Raw	48449
	RMS	38792
	concatFrontEnd	65418
KNN	Waveletconcat	34016
	Waveletenergy	34008
	Raw	34006
	RMS	38258
	concatFrontEnd	56306