UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Identificação e classificação de curtos-circuitos em sistemas elétricos de potência utilizando redes neurais e máquina de vetores de suporte.

Kenneth Roosevelt Sampaio Mendonça

Monografia - MBA em Inteligência Artificial e Big Data

ICMC USP
SÃO CARLOS

Kenneth Roosevelt Sampaio Mendonça

Identificação e classificação de curtos-circuitos em sistemas elétricos de potência utilizando redes neurais e máquina de vetores de suporte.

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Me. Willian Dener de Oliveira

Versão original

São Carlos

2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

S192i

Sampaio Mendonça, Kenneth Roosevelt
Identificação e classificação de curtos-circuitos em sistemas elétricos de potência utilizando redes neurais e máquina de vetores de suporte. / Kenneth Roosevelt Sampaio Mendonça; orientador Willian Dener de Oliveira. -- São Carlos, 2023.
85 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2023.

1. Curto-Circuito. 2. Sistema Elétrico de Potência - SEP. 3. Máquina de Vetores de Suporte - SVM. 4. Redes Neurais Artificiais - RNA. 5. Classificação Multiclasse. I. Dener de Oliveira, Willian, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Kenneth Roosevelt Sampaio Mendonça

Identification and classification of short circuits in electrical power systems using neural networks and support vector machines.

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Advisor: Prof. Me. Willian Dener de Oliveira

Original version

São Carlos

2023

Este trabalho é dedicado a todos aqueles que consideram a dúvida, a arte e o conhecimento parte de suas vidas.

AGRADECIMENTOS

Agradeço ao Prof. Me. Willian Dener de Oliveira pelo apoio, paciência, sugestões, correções e pelo tempo dedicado ao desenvolvimento deste trabalho de conclusão de curso.

Meu alegre agradecimento a Profa. Dra. Solange O. Rezende, Prof. Dr. Thiago A. S. Pardo, Profa. Dra. Roseli Ap. Francelin Romero, Profa. Dra. Cristina Dutra de Aguiar e Prof. Dr. Ricardo M. Marcacini. O aprendizado por meio da visão de vocês é motivador.

Agradeço amigo e mestre Eng. Prof. Me. Nilo Sérgio Soares Ribeiro pela revisão da parte teórica de Sistemas Elétricos de Potência.

"Não-mais-querer, não-mais-estimar, não–mais-criar: ó, que esse grande cansaço esteja para sempre longe de mim!"

Friedrich Nietzsche, Ecce Homo

RESUMO

Mendonça, K.R.S. **Identificação e classificação de curtos-circuitos em sistemas elétricos de potência utilizando redes neurais e máquina de vetores de suporte.**. 2023. 85p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

Durante a operação de Sistemas Elétricos de Potência várias contingências sistêmicas ocorrem, sendo que estas contigências podem desligar parte ou toda a rede. Umas destas contingências sistêmicas são os já amplamene analisados e estudados curtos-circuitos. Os curtos-circuitos são classificados em trifásico aterrado, trifásico isolado, bifásifo aterrado, bifásico isolado e monofásico. Cada tipo de curto-circuito apresenta um comportamento específico com relação às suas tensões e correntes, tornando-os distiguíveis em um processo de classificação multiclasse. Este trabalho avalia a capacidade de várias configurações de redes neurais e máquina de vetores de suporte com o objetivo de identificar a presença ou não de curto-circuito, bem como classifica-las nos tipos propostos para o fenômeno. Nas simulações dos modelos foram encontrados configurações que obtiveram acurácia, precisão, revocação e valor AUC com todas as predições corretamente classificadas no dataset de teste.

Palavras-chave: Curto-circuito. Sistema Elétrico de Potência - SEP. Máquina de Vetores de Suporte - SVM. Redes Neurais Artificiais - RNA. Classificação Multiclasse.

ABSTRACT

Mendonça, K.R.S. **Identification and classification of short circuits in electrical power systems using neural networks and support vector machines.**. 2023. 85p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

During the operation of Electrical Power Systems several systemic contingencies occur and these contingencies can shut down part or all of the network. One of these systemic contingencies are the already widely analyzed and studied short circuits. The short circuits are classified as three-phase grounded, three-phase isolated, two-phase grounded, isolated two-phase and single-phase. Each type of short circuit has a specific behavior in relation to their voltages and currents, making them distinguishable in a process of multiclass classification. This work evaluate the capacity of several configurations of neural networks and support vector machines with the aim of identifying the presence or absence of a short circuit, as well as classifying them into the proposed types for the phenomenon. In model simulations, configurations were found that obtained accuracy, precision, recall and AUC value with all predictions correctly classified in the test dataset.

Keywords: Short circuit. Electrical Power System. Vector Machine Support - SVM. Artificial Neural Networks - ANN. Multiclass Classification.

SUMÁRIO

1 INTRODUÇÃO

A observação da natureza dos fenômenos físicos, a dúvida e a criatividade fazem parte da evolução da humanidade. A análise da natureza dos fenômenos físicos culminaram com a percepção de um ente comum a todos os eventos observáveis: a Energia. A Energia pode se apresentar de várias formas: Energia Potencial, Energia Cinética, Energia Mecânica, Energia Química, Energia Luminosa, Energia Elétrica, dentre outras (REIS, 2003).

A Energia exerce uma função crucial para a humanidade: ao lado de transportes, telecomunicações, águas/saneamento, etc, estrutura o mundo moderno fomentando a incorporação do ser humano ao modelo de desenvolvimento vigente (REIS, 2003).

O crescente consumo de Energia, em suas diversas formas, fomentou o desenvolvimento de tecnologias para sua produção, bem como tecnologias para o seu transporte aos grandes centros de carga (STEVENSON, 1986). Para se alcançar excelência no atendimento ao mercado é preciso assegurar o fornecimento contínuo de Energia Elétrica. Com este objetivo, a estrutura de trasmissão opera interligada formando redes complexas que neste trabalho serão denominadas de Sistema Elétrico (KINDERMANN, 1997). Assim, o Sistema Elétrico pode ser descrito como uma rede de conexões que pode ser dividida em três partes principais: as unidades geradoras, o sistema de transmissão e o sistema de distribuição (STEVENSON, 1986). No caso brasileiro, o Sistema Elétrico é denominado como Sistema Interligado Nacional (SIN) sendo esse, em sua maioria, conectado. Os casos em que não ocorrem conexão com a rede é conhecido como Sistema Isolado.

Com o enfoque tanto em Operação quanto em Planejamento de curto, médio e longo prazo, o SIN deve ser monitorado constantemente. Este monitoramento visa analisar o comportamento do sistema frente à contingências e alterações da rede, diagnosticar e prever o resultado das medidas adotadas, planejar ampliações e mudanças de configuração do sistema (KINDERMANN, 1997). Este acompanhamento será realizado através de sistemas de monitoramento em conjunto com estudos elétricos.

Para estudos de proteção e operação de Sistemas Elétricos as correntes de curto-circuito serão avaliadas. No caso da Operação do SIN, faz-se necessária a rápida e eficaz identificação de Curtos-Circuitos na rede, bem como a identificação do Tipo de Curto-Circuito para a definição das ações a serem tomadas após o ocorrência de uma contingência.

1.1 Motivação

Este trabalho é motivado pela proposta de tornar ainda mais eficiênte o processo de Operação de Sistemas Elétricos dentro do SIN por meio da detecção e identifição de curtos-circuitos em tempo real.

(UDDIN *et al.*, 2021) propõe que vários sistemas de detecção de faltas em Linhas de Transmissão foram desenvolvidos, mas que estes sistemas demandam muito tempo computacional, possuem dados incompletos, demanda o valor do ângulo de início da ocorrência e a variação da resistência da falta. Em seu trabalho, o foco é a melhoria da performance do classificador existente e reduzir a complexidade computacional. Para isto, são propostos quatro métodos para a Classificação dos Curtos-Circuitos: 1º) Máquina de Vetores de Suporte - SVM - combinado com Particle Swarm Optimization, 2º) Somente SVM, 3º) SVM combinada com Transformada Wavelet e 4º) SVM combinada com Transformada Wavelet e GridSearchCV.

No trabalho de (MUNDRA; SHAH; CHATTERJEE, 2022) é descrita a utilização de Relés de linha de uso Duplo (Dual use line relays - DULR) para o monitoramento da rede, detecção de curto-circuito e inicialização do disjuntor. O DULR é um tipo especial de Relé que possui a capacidade de entregar a medida do sincrofasor. Um sicrofasor é uma unidade de medição fasorial usado para estimar a magnitude e o ângulo de fase de uma grandeza fasorial elétrica na rede usando uma fonte de tempo comum para sincronização. A partir da combinação dos dados do sincrofasor com um algoritmo de Extreme Gradient Boosting (XGBoost) foram explorados a identificação de curto-circuito em Sistemas de Potência.

A utilização de vários algoritmos de Machine Learning é abordado em (HASSAN; NIZAMI, 2022) para o problema de curto-circuito. As faltas investigadas neste estudo incluem os tipos bifásica isolada (L-L), trifásica isolada (L-L-L), monofásica (L-G), bifásica aterrada (L-L-G) e a trifásica aterrada (L-L-L-G). São avaliados neste trabalho 6 algoritmos de Aprendizado de Máquina que são Decision Tree, Máquina de Vetores de Suporte, K-Nearest Neighbor (knn), Random Forest, XGBoost (XGB) e Naive Bayes para o reconhecimento e classificação da contingência no sistema elétrico.

1.2 Questão de Problema

A questão colocada será se é possível detectar e classificar curtos-circuitos em Sistemas de Potência por meio de várias configurações de Redes Neurais Artificiais e Máquina de Vetores de Suporte.

1.3 Objetivos

Este trabalho avaliará a detecção e classificação de curtos-circuitos em Sistemas Elétricos de Potência utilizando-se Redes Neurais Artificiais e Máquina de Vetores de Suporte.

1.4 Organização do Trabalho

Sobre a organização desse trabalho, o Capítulo 2 contém o referencial teórico detalhando os conceitos de Curto Circuito em Sistemas Elétricos de Potência, Redes Neurais, Máquina de Vetores de Suporte e as métricas de avaliação dos modelos propostos. No Capítulo 3 serão apresentados e caracterizados o dataset selecionado para treinamento e as configurações específicas de cada algoritmo de Aprendizado de Máquina. O Capítulo 4 abrange a apresentação dos resultados para cada teste realizado em conjunto com a avaliação dos algoritmos de classificação. O Capítulo 5 traz uma discussão sobre os resultados encontrados e possibilidades futuras.

2 DESENVOLVIMENTO

Neste capítulo serão apresentados os referenciais teóricos utilizados na detecção e identificação de curtos-circuitos em Sistemas Elétricos de Potência.

2.1 O Sistema Elétrico de Potência

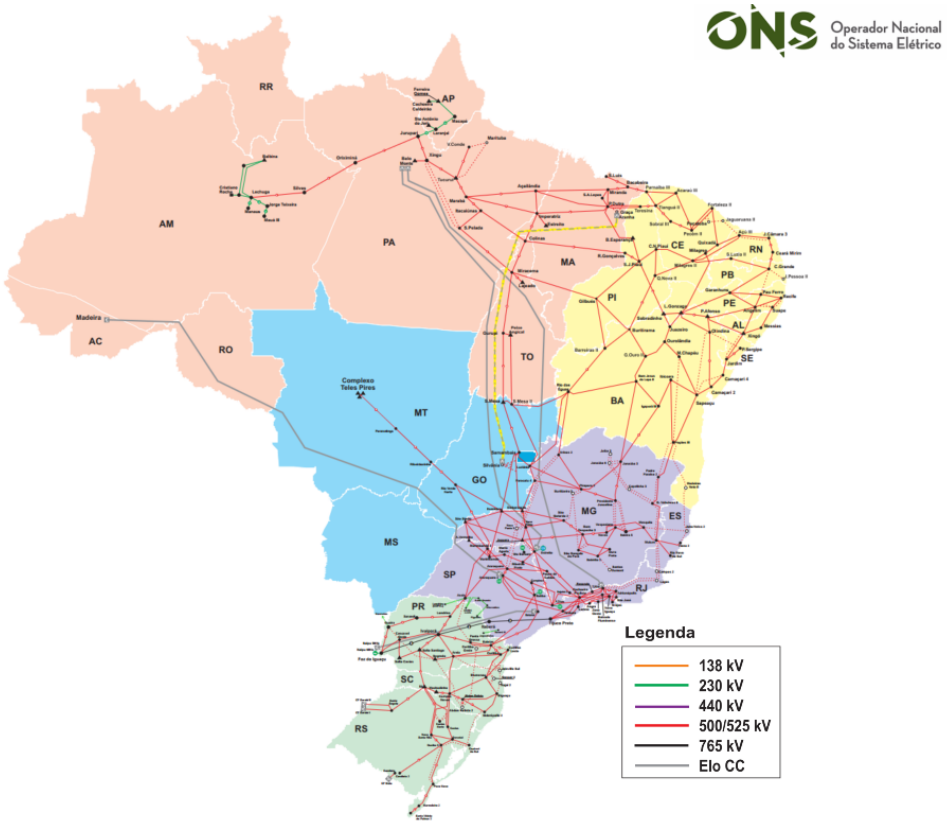
O Sistema Elétrico de Potência - SEP - pode ser dividido em geração, transmissão, sub-transmissão e sistemas de distribuição. Esses SEPs são organizados para conectar eletricamente regiões em um país, regiões estas que são conectadas através de Linhas de Transmissão - LT - para formar sistemas nacionais e também internacionais. Cada área é interconectada com outra nos termos de alguns parâmetros contratados: geração e programação de geração, geração e o fluxo de potência nas linhas de transmissão, operações de contingência (DAS, 2010).

No Brasil, esta conexão entre geração e carga é entitulada de Sistema Interligado Nacional - SIN - e é gerido pelo Operador Nacional do Sistema - ONS. O SIN é um sistema que possui 216759 km de extensão, linhas estas operando em tensões de 230, 345, 440, 500/525, 600, 750 e 800 KV (ONS, 2023b). As Figuras 1 e 2 apresentam uma representação do SIN para o horizonte de operação de 2027 e a atual distribuição de fontes de geração de energia elétrica no Brasil ano de 2023.

Devido a exigência de assegurar a continuidade de suprimento de energia elétrica aos diversos mercados nacionais, a forma como a rede se comporta deve ser acompanhada constantemente. Este acompanhamento vai desde a operação do sistema, os planejamentos de curto, medio e longo prazo ao acompanhamento de contingências que retiram de funcionamento equipamentos do SIN. Para se obter um histórico constantemente atualizado, avaliar o comportamento quanto a ocorrência das contingências e alteraçãoes, avaliar e prever o resultado das medidas a serem implementadas, planejar ampliaçãoes e alterações de configuração, o SIN precisa ser rigorosamente representado através de uma modelagem adequada ao tipo de análise que se for realizar (KINDERMANN, 1997).

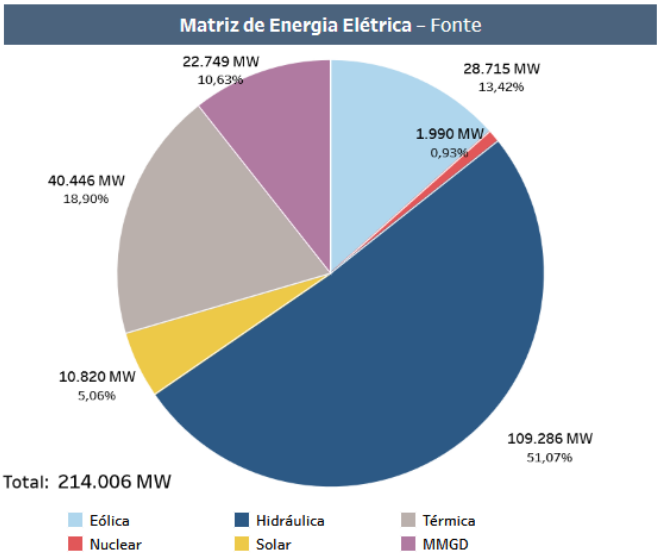
A Figura 3 apresenta um pequeno sistema elétrico de potência onde T_1 e T_2 são transformadores (BORGES, 2005).

Figura 1 – Representação do SIN horizonte 2027.



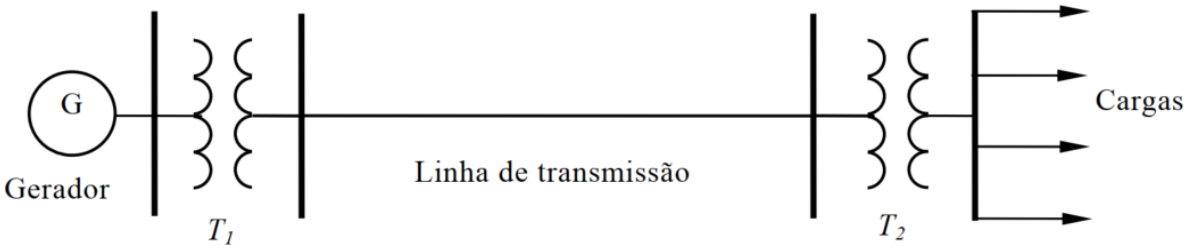
(ONS, 2023a)

Figura 2 – Matriz Energetica Brasileira ONS 2023.



(ONS, 2023b)

Figura 3 – Modelo Unifilar de um Pequeno Sistema Elétrico de Potência.



(BORGES, 2005)

2.1.1 O Sistema Trifásico

Cerca de 95% da energia elétrica do mundo é gerada na modelagem trifásica. Geradores Elétricos em plantas de geração convertem a energia mecânica produzida nas turbinas, movidos por água descendo de uma barragem ou por vapor circulante, em energia elétrica. As plantas de geração são comumente distantes dos centros de consumo e como resultado as LTs são construídas para transmitir em altas tensões, reduzindo assim as perdas na transmissão - alta tensão é traduzida em baixas correntes (IRWIN; NELMS, 2007).

A análise dos sistemas polifásicos, mais especificamente dos sistemas trifásicos, fazem parte dos estudos dos circuitos de corrente alternada em regime permanente. Os sistemas trifásicos são estudados devido a algumas vantagens. Esta forma de avaliação é mais eficiente e econômica para gerar e transmitir energia elétrica em uma modelagem polifásica do que em um sistema monofásico. Como resultado, a maior parte da energia elétrica é transmitida em circuitos polifásicos. Em vários países do mundo os sistemas trabalham com a frequência de 60hz, sendo que em alguns lugares a frequência pode ser de 50hz (IRWIN; NELMS, 2007).

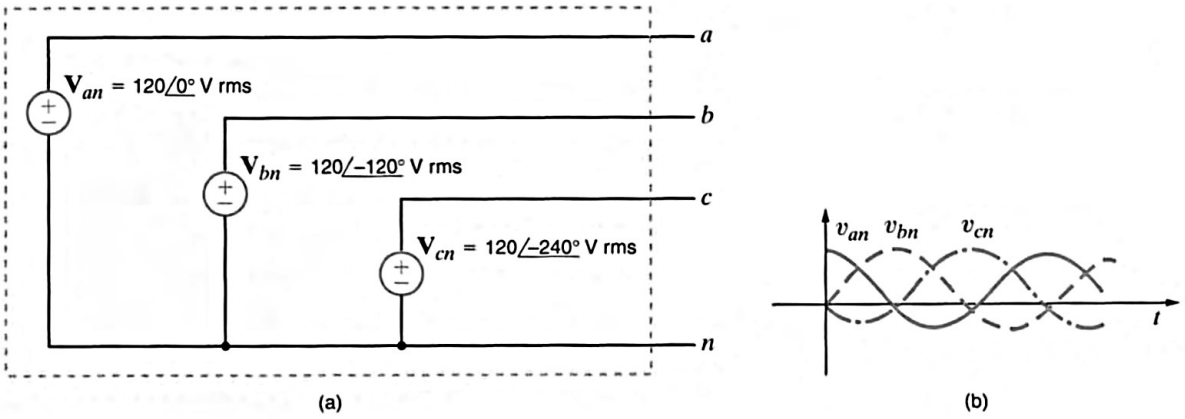
Como o nome implica, os circuitos trifásicos são aqueles que a função de força é um sistema trifásico de tensões. Se as três tensões senoidais tem a mesma magnitude e frequência e cada tensão possui 120º de diferença angular de uma fase a outra, as tensões são classificadas como balanceadas. Se as cargas são tal como as correntes produzidas pelas tensões, então as cargas também são balanceadas. O circuito completo é conhecido como *circuito trifásico balanceado*. Na Expressão 2.1 são apresentadas as expressões fasoriais de tensão para o modelo trifásico balanceado (IRWIN; NELMS, 2007).

$$\begin{aligned} \hat{V}_{an} &= V_{an}\angle 0^\circ \ [V] \\ \hat{V}_{bn} &= V_{bn}\angle 120^\circ \ [V] \\ \hat{V}_{cn} &= V_{cn}\angle 240^\circ \ [V] \end{aligned} \tag{2.1}$$

Onde, para o sistema equilibrado, $\hat{V}_{an}$, $\hat{V}_{bn}$ e $\hat{V}_{cn}$ são as representações fasoriais das tensões e $V_{an} = V_{bn} = V_{cn}$ representam o módulo das tensões.

Na Figura 4(a) e (b) um modelo para uma fonte de tensão balanceada e um gráfico da forma senoidal da tensão balanceada (IRWIN; NELMS, 2007).

Figura 4 – Tensões Trifásicas Balanceadas.



(IRWIN; NELMS, 2007)

O duplo subescrito apresentado na notação dos fasores, como em $\hat{V}_{an} = V_{an}\angle 0°$, significa que $\hat{V}_{an}$ é a tensão para o ponto a em relação ao ponto n (tensão de fase); esta apresentação em duplo subescrito também será utilizada para a representação das correntes. Assim, $\hat{I}_{an}$ será usada para se referirir a corrente que atravessa o circuito no intervalo de a a n. No entanto, esta forma de apresentar esta corrente deve se atentar ao exato caminho no qual a corrente circula. Os fasores de tensão apresentados na forma da Expressão 2.1 podem ser representados no domínio do tempo como (IRWIN; NELMS, 2007):

$$\begin{aligned} v_{an} &= V_m cos wt\ [V] \\ v_{bn} &= V_m cos(wt - 120°)\ [V] \\ v_{cn} &= V_m cos(wt - 240°)\ [V] \end{aligned} \tag{2.2}$$

Onde v_{an} é a tensão instantânea; $V_m = V_{an} = V_{bn} = V_{cn}$ é o módulo ou valor máximo da tensão de fase; wt é a velocidade angular em rad/s e t é o tempo em segundos. Cabe ressaltar que w depende diretamente da frequência do sistema de acordo com a expressão $w = 2\pi$f (IRWIN; NELMS, 2007).

Dada uma carga balanceada, a expressão para a corrente elétrica na fonte de tensão é dada por (IRWIN; NELMS, 2007):

$$\begin{aligned} i_a &= I_m cos(wt - \theta)\ [A] \\ i_b &= I_m cos(wt - \theta - 120°)\ [A] \\ i_c &= I_m cos(wt - \theta - 240°)\ [A] \end{aligned} \tag{2.3}$$

Onde i_a é a corrente instantânea; I_m é o módulo ou o valor máximo da corrente; wt é a velocidade angular em rad/s, θ é o ângulo da impedância z e t é o tempo em segundos. Cabe ressaltar que w depende diretamente da frequência do sistema de acordo com a expressão $w = 2\pi$f (IRWIN; NELMS, 2007).

A potência instantânea produzida pelo sistema trifásico será a soma das potências de cada fase, Dado que (IRWIN; NELMS, 2007):

$$p(t) = V_m I_m cos(wt + \theta) \tag{2.4}$$

A potência trifásica será:

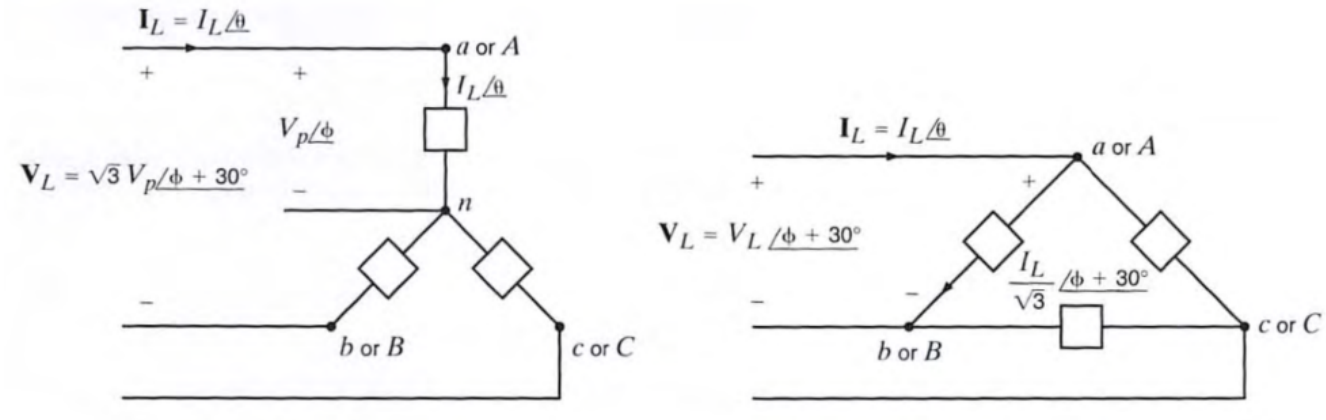
$$p_{3F}(t) = p_a(t) + p_b(t) + p_c(t) \tag{2.5}$$

Com algum desenvolvimento matemático, encontra-se a expressão:

$$p_{3F}(t) = 3\frac{V_m I_m}{2} cos\theta\ [\text{W}] \tag{2.6}$$

A Figura 5 traz os modelos de conexão de carga em estrela (Y) ou em delta($\triangle$).

Figura 5 – Relações entre Tensão e Corrente para Y e $\triangle$.



(IRWIN; NELMS, 2007)

Valor por Unidade

Normalmente nas formulações matemáticas, cálculos, etc as grandezas trabalhadas possuem implicitamente como base o valor 1. Quando se quer usar um valor unitário pré-estabelecido $\neq 1$, todos os valores dessas grandezas ficam referenciados em relação ao número pré-fixado. A formulação utilizando esta medida é entitulada por resolução por unidade - pu (KINDERMANN, 1997).

Para o caso da Engenharia Elétrica, a utilização da representação do SEP em pu apresenta várias vantagens na simplificação da modelagem e computação do sistema (KINDERMANN, 1997).

VALOR POR UNIDADE(pu): é o quociente do valor da grandeza e o valor selecionado de base dessa grandeza.

$$\text{valor pu} = \frac{\text{valor real da grandeza}}{\text{valor base da grandeza}} \tag{2.7}$$

Cada ponto do sistema elétrico é descrito por quatro grandezas: a tensão elétrica (V), a corrente elétrica (I), a potência aparente (S) e a impedância (Z). Uma vez conhecidas duas destas grandezas, as outras duas também ficam conhecidas. No Sistema de Potência é comum se escolher como bases Tensão(V_{base}) e a Potência Aparente(S_{base}), e a partir dessas duas bases são calculadas a corrente e a impedância para o nível de tensão correspondente (KINDERMANN, 1997).

Um sistema trifásico (3ϕ) de potência envolve cargas e transformadores ligados em delta ($\triangle$) e estrela (Y). Para análises de curto-circuito, para proteção de sistemas elétricos, são realizados aplicando-se componentes simétricas, que são equilibradas. Como decorrência deste fato, a análise poderá ser feita em apenas uma fase(1ϕ). A seguir são expressas as algumas transformações em pu para um sistema elétrico (KINDERMANN, 1997).

$S_{base} \rightarrow$ Potência aparente base do sistema trifásico. É a soma das potências aparentes base de cada base.

$$S_{b(3\phi)} = 3S_{b(1\phi)} \tag{2.8}$$

$V_{base} \rightarrow$ Tensão base de linha à linha, ou $\sqrt{3}$ vezes a tensão de base do sistema Y equivalente.

$$V_{base} = \sqrt{3}V_{b_f} \tag{2.9}$$

$V_{b_f} \rightarrow$ Tensão base de fase.

$I_{base} \rightarrow$ A corrente de base é a corrente de linha do sistema trifásico original da fase do Y equivalente.

$$S_{base} = \sqrt{3} V_{base} I_{base} \tag{2.10}$$

$$I_{base} = \frac{S_{base}}{\sqrt{3} V_{base}} \tag{2.11}$$

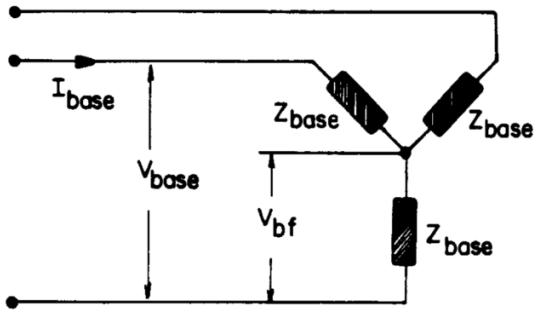
$Z_{base} \rightarrow$ A impedância de base de um sistema trifásico é sempre a impedância da fase do sistema trifásico em Y equivalente.

$$Z_{base} = \frac{V_{b_f}}{I_{b_f}} \tag{2.12}$$

$$Z_{base} = \frac{V_{base}^2}{S_{base}} \tag{2.13}$$

A Figura 6 localiza algumas destas grandezas descritas acima.

Figura 6 – Modelo em Y equivalente.



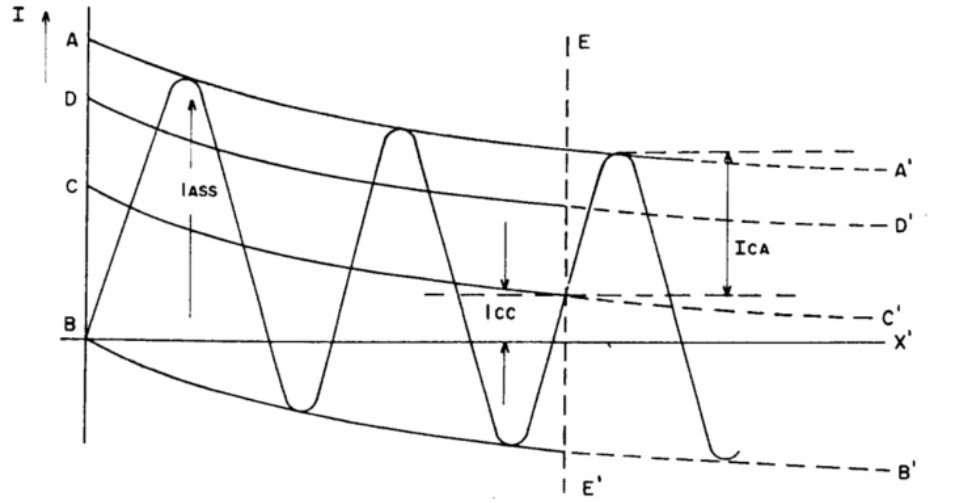
(KINDERMANN, 1997)

2.1.2 Análise de Faltas no Sistema Elétrico de Potência

Uma falta em um SEP é alguma falha que ocasione interferência com o fluxo normal de corrente elétrica. Grande parte das faltas em LTs de 115kV ou superiores é provocada por descargas atmosféricas que culminam no centelhamento dos isoladores das linhas de transmissão. A experiência mostra que de 70 a 80% das faltas em linhas de transmissão são entre uma fase e a terra. As faltas trifásicas tem a menor frequência de ocorrência, contabilizando 5% das ocorrências. Outros tipos de faltas em LTs são as faltas entre duas fases isoladas(sem envolvimento da terra) e as faltas envolvendo a terra (STEVENSON, 1986).

A corrente circulante nas diversas partes do SEP imediatamente após a ocorrência de uma falta (conhecida como corrente subtransitória e ocorre no tempo de microsegundos após a falta) difere daquela que circula poucos ciclos depois (ressaltando que um ciclo no SEP está relacionado a frequência de trabalho do sistema que é de 60hz = 60ciclos/s). A corrente nesse segundo momento é conhecida como corrente transitória. As duas correntes descritas em muito diferem da corrente circulante no sistema em uma situação normal de operação. Essa última descrição é conhecida como corrente em regime permantente. O cálculo de faltas compreende-se como a determinação dessas correntes para os diversos tipos de falta e em vários pontos do sistema (STEVENSON, 1986).

Figura 7 – Instantes Transitórios e Sub-Transiórios para a Corrente de Falta.



(D'AJUZ *et al.*, 1985)

A Figura 7 ilustra a corrente assimétrica $Iass$ e a corrente simétrica Ica. A corrente assimétrica é uma outra denominação para a corrente subtransitória e a corrente simétrica é a corrente de falta em regime permanente. A corrente Icc apresentada faz referência a componente contínua da corrente de curto-circuito durante a passagem pelos três estágios descritos para a contingência (D'AJUZ *et al.*, 1985).

O Teorema de Fortescue

O teorema de Fortescue entitulado "*Método de componentes simétricas aplicado à solução de circuitos polifásicos*", estabelece que um equacionamento com n fasores desequilibrados pode ser analisado em n sistemas de fasores equilibrados. Estes fasores transformados são conhecidos como componentes simétricas. A expressão analítica geral para um sistema desequilibrado com n fases é (KINDERMANN, 1997):

$$
\begin{aligned}
\dot{V}_a &= \dot{V}_{a_0} + \dot{V}_{a_1} + \dot{V}_{a_2} + ... + \dot{V}_{a_{(n-1)}} \\
\dot{V}_b &= \dot{V}_{b_0} + \dot{V}_{b_1} + \dot{V}_{b_2} + ... + \dot{V}_{b_{(n-1)}} \\
\dot{V}_c &= \dot{V}_{c_0} + \dot{V}_{c_1} + \dot{V}_{c_2} + ... + \dot{V}_{c_{(n-1)}} \\
&. \\
&. \\
&. \\
\dot{V}_n &= \dot{V}_{n_0} + \dot{V}_{n_1} + \dot{V}_{n_2} + ... + \dot{V}_{n_{(n-1)}}
\end{aligned} \tag{2.14}
$$

O sistema desequilibrado inicial de sequência de fase $a, b, c, ..., n$ é descrito pelos seus n fasores $\dot{V}_a, \dot{V}_b, \dot{V}_c, ...\dot{V}_n$ que trabalham na frequência polifásica do sistema. Cada um dos fasores, passando pela transformação proposta pelo equacionamento das expressões contidas em 2.14, será decomposto em n fasores denominados componentes de sequência de $0, 1, 2, 3, ..., (n-1)$. Desta forma será obtido um sistema de n sistemas equilibrados que representam o sistema desequilibrado (KINDERMANN, 1997).

Portanto, obtém-se os sistemas de (KINDERMANN, 1997):

1. **sequência zero** - são o sistema de fasores $\dot{V}_{a_0}, \dot{V}_{b_0}, \dot{V}_{c_0}, ..., \dot{V}_{n_0}$ de mesmo módulo e ângulo, girando no mesmo sentido e velocidade síncrona do sistema original de n fases.

2. **sequência 1** - são o sistema de fasores $\dot{V}_{a_1}, \dot{V}_{b_1}, \dot{V}_{c_1}, ..., \dot{V}_{n_1}$ de mesmo módulo, ângulo com defasamento de $\frac{2\pi}{n}$, girando no mesmo sentido e velocidade síncrona do sistema polifásico original.

3. **sequência 2** - são o sistema de fasores $\dot{V}_{a_2}, \dot{V}_{b_2}, \dot{V}_{c_2}, ..., \dot{V}_{n_2}$ de mesmo módulo, ângulo com defasamento de $2\left(\frac{2\pi}{n}\right)$, girando no mesmo sentido e velocidade síncrona do sistema polifásico original.

4. **sequência k-ésima** - são o sistema de fasores $\dot{V}_{a_k}, \dot{V}_{b_k}, \dot{V}_{c_k}, ..., \dot{V}_{n_k}$ de mesmo módulo, ângulo com defasamento de $k\left(\frac{2\pi}{n}\right)$, girando no mesmo sentido e velocidade síncrona do sistema polifásico original.

Somente os sistemas polifásicos com n igual a um número ímpar obterá sempre um sistema de sequência em perfeita simetria de fasores (KINDERMANN, 1997).

O Teorema de Fortescue em um sistema trifásico ficará assim equacionado: "Um sistema de 3ϕ de três fasores desequilibrados pode ser decomposto em três sistemas 3ϕ de três fasores balanceados chamados de componentes simétricas de sequência zero, positiva (1) e negativa(2)".

Do exposto, o Teorema de Fortescue será visualizado em uma representação analítica. Aplicando-se a transformação proposta pelo teorema, obtém-se o seguinte equacionamento (KINDERMANN, 1997):

$$\begin{aligned} \dot{V}_a &= \dot{V}_{a_0} + \dot{V}_{a_1} + \dot{V}_{a_2} \\ \dot{V}_b &= \dot{V}_{b_0} + \dot{V}_{b_1} + \dot{V}_{b_2} \\ \underbrace{\dot{V}_c} &= \underbrace{\dot{V}_{c_0}} + \underbrace{\dot{V}_{c_1}} + \underbrace{\dot{V}_{c_2}} \end{aligned} \tag{2.15}$$

Usando como referência os colchetes embaixo da última linha, e sequindo da coluna da esquerda para a coluna da direita, tem-se (KINDERMANN, 1997):

1. Sistema trifásico desequilibrado.
2. Sistema trifásico equilibrado de sequência zero.
3. Sistema trifásico equilibrado de sequência positiva ou 1.
4. Sistema trifásico equilibrado de sequência negativa ou 2.

Analogamente, como foi explicado no item referente a Circuitos Trifásicos, os equacionamentos para tensão e corrente - em um circuito equilibrado - estão defasados de 120° de uma fase para a outra; assim tem-se também um defasamento para as componentes simétricas de 120°. A partir deste ponto, o defasamento ângular das fases de cada sequência será tratado como $a = 1\angle 120°$. Como os sistemas trifásicos de sequência são equilibrados, pode-se realizar a avaliação em relação a uma fase a e assume-se que o módulo das componentes sejam iguais $V_{a_1} = V_{b_1} = V_{c_1}$ (KINDERMANN, 1997):

$$\begin{aligned} &\dot{V}_{a1} \\ \dot{V}_{b1} &= 1\angle -120°\dot{V}_{a_1} = a^2\dot{V}_{a_1} \\ \dot{V}_{c1} &= 1\angle -240°\dot{V}_{a_1} = a\dot{V}_{a_1} \end{aligned} \tag{2.16}$$

Assim, a Equação 2.15 pode ser re-escrita (KINDERMANN, 1997):

$$\begin{aligned} \dot{V}_a &= \dot{V}_{a_0} + \dot{V}_{a_1} + \dot{V}_{a_2} \\ \dot{V}_b &= \dot{V}_{a_0} + a^2\dot{V}_{a_1} + a\dot{V}_{a_2} \\ \dot{V}_c &= \dot{V}_{a_0} + a\dot{V}_{a_1} + a^2\dot{V}_{a_2} \end{aligned} \tag{2.17}$$

Na forma matricial (KINDERMANN, 1997):

$$\begin{bmatrix} \dot{V}_a \\ \dot{V}_b \\ \dot{V}_c \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \begin{bmatrix} \dot{V}_{a_0} \\ \dot{V}_{a_1} \\ \dot{V}_{a_2} \end{bmatrix} \tag{2.18}$$

Representando a matriz por T (KINDERMANN, 1997):

$$T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix} \tag{2.19}$$

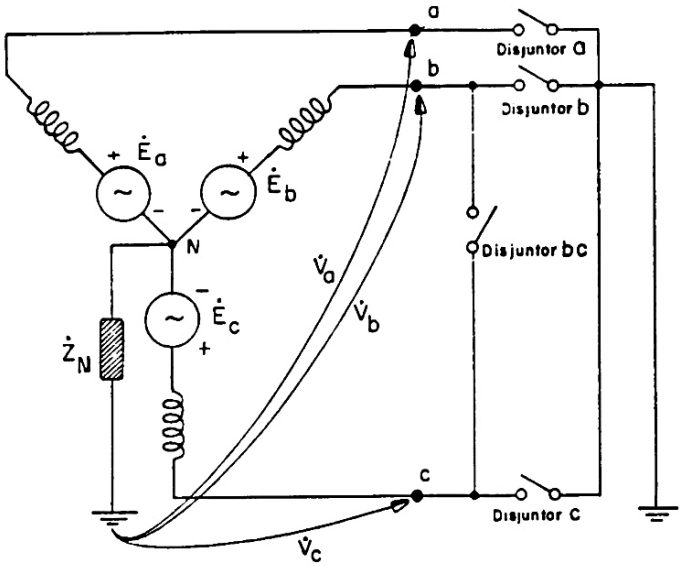
Curto-circuito no Gerador Síncrono

O SEP possui diversos equipamentos em sua composição. Estes equipamentos são as LTs, Reatores de Linha e de Barra, Reatores limitadores de corrente, Capacitores Shunt e Série, Geradores, etc (D'AJUZ *et al.*, 1985). Neste trabalho, os geradores serão utilizados para apresentar o equacionamento referente a cada um dos tipos de curto-circuito expostos anteriormente. Este equacionamento deve ser expandido para todo o SEP com as devidas adequações. Não serão desenvolvidos detalhamentos matemáticos, sendo somente apresentados os diagramas e as expressões que representam cada fenômeno.

A Figura 8 traz um modelo de representação de uma máquina síncrona e as conexões a serem alteradas para representar a ocorrência de cada tipo de curto-circuito analisado. Supõe-se que o gerador síncrono estará funcionando com o rotor girando à velocidade síncrona, exitado de modo a gerar tensões nominais nas bobinas da armadura e sem carga conectada aos terminais (KINDERMANN, 1997). Cada fechamento de disjuntor, descritos a seguir, configurará o modelo para o curto-circuito desejado:

- Para o curto-circuito 3ϕ, fecham-se os disjuntores a,b e c;
- Para o curto-circuito 1ϕ, fecham-se os disjuntores a;
- Para o curto-circuito 2ϕ, fecham-se os disjuntores b e c.

Figura 8 – Representação do Gerador Síncrono.



(KINDERMANN, 1997)

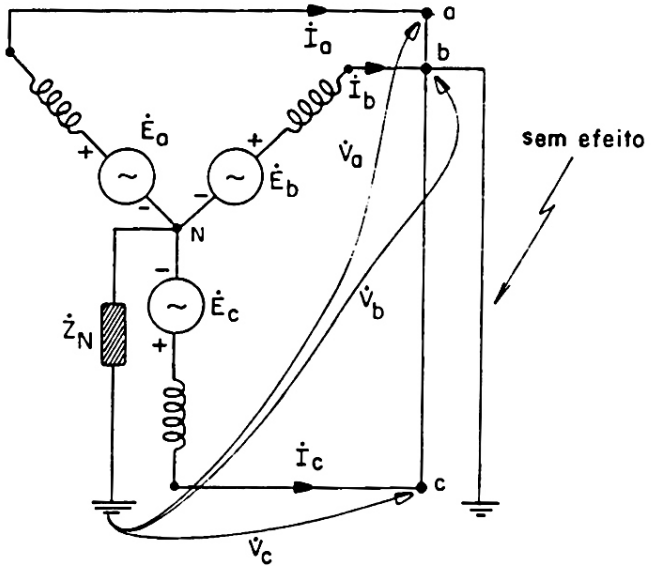
Os curtos-circuitos que serão apresentadas são os seguintes: curto-circuito trifásico 3ϕ isolado, curto-circuito bifásico 2ϕ isolado, curto-circuito bifásico à terra 2ϕ-t e curto-circuito monofásico à terra 1ϕ-t(KINDERMANN, 1997).

1. **Curto-circuito trifásico isolado** (3ϕ)

   O gerador síncrono é modelado para ser perfeitamente equilibrado, ou seja, as bobinas são iguais e simetricamente distribuídas espacialmente na armadura (estator). Sendo assim, as correntes de curto apresentarão apenas componentes de sequência positiva. Este curto ocorrerá fechando-se os três disjuntores a, b e c na Figura 8, obtendo-se assim a configuração apresentada na Figura 9 (KINDERMANN, 1997). Como esse curto-circuito é equilibrado, não haverá corrente circulando na conexão com a terra.

   Para esta falta, o comportamento da tensão e da corrente de sequência positiva no nó b são descritos nas duas equações a seguir:

Figura 9 – Curto-Circuito Trifásico Isolado.



(KINDERMANN, 1997)

As características do defeito impõem a seguinte condição ao modelo:

$$\dot{V}_a = \dot{V}_b = \dot{V}_c = 0 \tag{2.20}$$

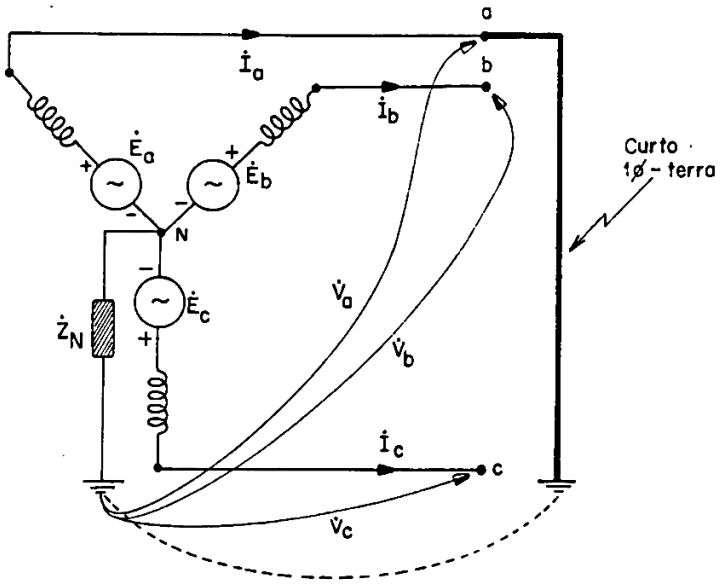
Resultando em:

$$\dot{I}_{a1} = \frac{\dot{E}_a}{jX_1} \tag{2.21}$$

2. **Curto-circuito monofásico à terra(1ϕ-t)**

Esta falta será obtida fechando-se o disjuntor a na Figura 8, chegando-se assim a configuração apresentada na Figura 10 (KINDERMANN, 1997).

Figura 10 – Curto-Circuito Monofásico à Terra.



(KINDERMANN, 1997)

As características do defeito impõem as seguintes condições ao modelo:

$$\dot{V}_a = 0 \tag{2.22}$$

$$\dot{I}_b = \dot{I}_c = 0 \tag{2.23}$$

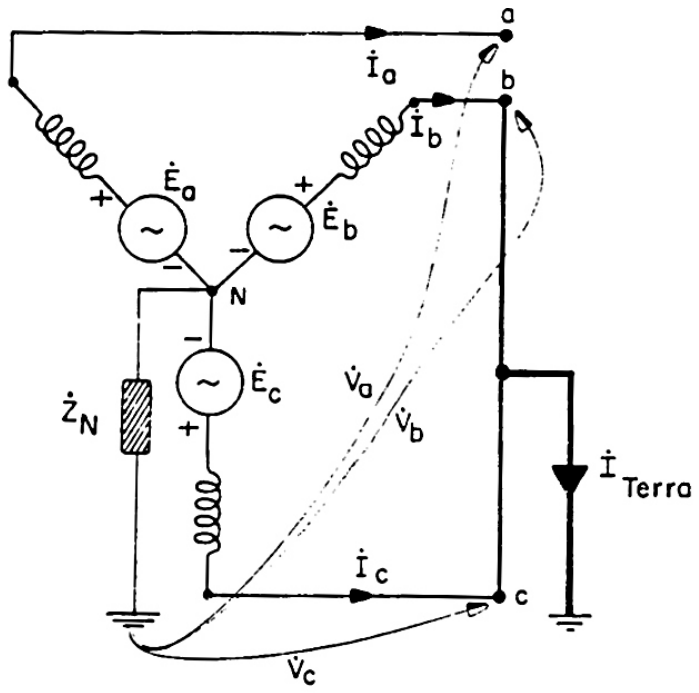
Resultando em:

$$\dot{E}_{a1} = (jX_1 + jX_2 + jX_3 + 3\dot{Z}_n)\dot{I}_{a1} \tag{2.24}$$

3. **Curto-circuito bifásico à terra(2ϕ-t)**

Esta falta será obtida fechando-se os disjuntores b e c na Figura 8 com a adição de uma conexão entre as fases envolvidas para à terra, chegando-se assim a configuração apresentada na Figura 11 (KINDERMANN, 1997).

Figura 11 – Curto-Circuito Bifásico à Terra.



(KINDERMANN, 1997)

As características do defeito impõem as seguintes condições ao modelo:

$$\dot{V}_b = \dot{V}_c = 0 \tag{2.25}$$

$$\dot{I}_a = 0 \tag{2.26}$$

Resultando em:

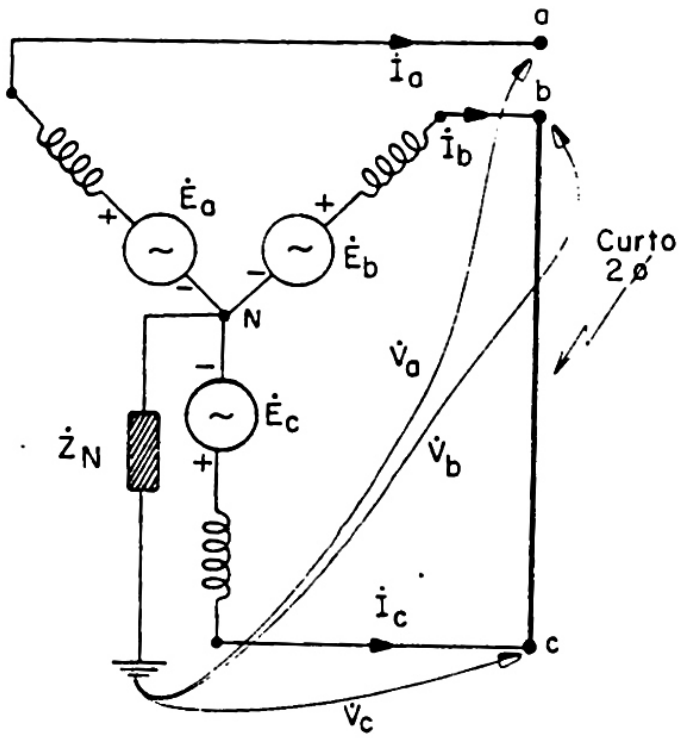
$$\dot{I}_{terra} = 3\dot{I}_{a0} = \dot{I}_b + \dot{I}_c \tag{2.27}$$

$$\dot{V}_{bN} = \dot{V}_{cN} = -\dot{V}_{N-terra} = 3\dot{Z}_N\dot{I}_0 = \dot{Z}_N(\dot{I}_b + \dot{I}_c) \tag{2.28}$$

4. **Curto-circuito bifásico isolado (2ϕ)**

Esta falta será obtida fechando-se os três disjuntores b e c na Figura 8, chegando-se assim a configuração apresentada na Figura 12 (KINDERMANN, 1997).

Figura 12 – Curto-Circuito Bifásico Isolado.



(KINDERMANN, 1997)

As características do defeito impõem as seguintes condições ao modelo:

$$\dot{V}_b = \dot{V}_c \tag{2.29}$$

$$\dot{I}_a = 0 \tag{2.30}$$

$$\dot{I}_b + \dot{I}_c = 0 \tag{2.31}$$

Resultando em:

$$\dot{V}_a = -2\dot{V}_b \tag{2.32}$$

2.2 Inteligência Artificial

Algumas necessidades são atendidas na computação por meio de algoritmos que descrevem o passo a passo da solução de um problema. Mas como passar para um algoritmo uma necessidade de reconhecimento de um rosto ou de voz que fazem parte do dia a dia das pessoas? O que deverá ser considerado para um reconhecimento eficiente? Esta atividade é facilmente realizada por todos dentro das suas rotinas, e fazem isto através de reconhecimento de padrões. Este reconhecimento é possível quando uma pessoa reconhece determinadas características de um rosto ou em uma fala a partir de sua exposição à vários tipos de rosto e voz (FACELI; LORENA; CARVALHO, 2011).

Médicos experientes conseguem, a partir de um conjunto de sintomas e resultados de exames clínicos, identificar se um paciente está com problemas cardíacos. Este diagnóstico é obtido pela análise dos sintomas, pela análise de exames e pela experiência adquirida por este médico no decorrer dos seus anos de trabalho. Como se poderia descrever um programa que, a partir de informações dos sintomas e resultados de exames clínicos, apresente um diagnóstico tão bom quanto o de um médico experiente (FACELI; LORENA; CARVALHO, 2011)?

Mesmo com a dificuldade de programar tais soluções em um computador, deve-se levar em conta também a necessidade de lidar, de forma eficiente, com uma grande quantidade de informações e se considerar a quantidade de vezes que estas atividades serão realizadas por dia. Pensando nestes itens, pode-se imaginar que algumas atividades serão muito difíceis ou até impossíveis de se realizar por pessoas (FACELI; LORENA; CARVALHO, 2011). Técnicas de Inteligência Artificial (IA), em particular de Aprendizado de Máquina (AM), têm sido aplicadas a determinados problemas reais com bons resultados.

2.2.1 Inteligência Artificial e Aprendizado de Máquina

Há pouco tempo, a IA era considerada como uma área teórica, com aplicações somente em alguns pequenos problemas diferentes, curiosos e desafiadores. A maior parte das soluções do dia a dia que recorriam a computação era resolvida por códigos em alguma linguagem de programação (FACELI; LORENA; CARVALHO, 2011).

A partir da década de 70, ocorreu uma grande propagação do uso de técnicas de IA para a resolução de problemas práticos. Na maior parte dos casos, estes problemas eram trabalhados computacionalmente por meio da interação com os profissionais especialistas em cada domínio e a definição de linhas, frequentemente por regras lógicas, em um programa de computador. O processo de aquisição de conhecimento, nestes casos, envolvia entrevistas com o objetivo de acessar as regras que estes profissionais utilizavam para a tomada de decisão. Este processo possuia e possui várias limitações. Como exemplo, tem-se a subjetividade(intuição) e pouca cooperação(medo de demissão), etc (FACELI;

LORENA; CARVALHO, 2011).

Atualmente, com o desenvolvimento do conhecimento, consequente crescimento da complexidade dos problemas computacionais e do volume de dados gerados a serem analisados, tornou-se clara a necessidade de ferramentas de solução mais sofisticadas e autônomas – sem intervenção humana e necessidade de especialistas (FACELI; LORENA; CARVALHO, 2011).

Dentro da necessidade de soluções para problemas complexos de maneira autônoma, a Inteligência Artificial surge trazendo soluções e vários questionamentos. A definição de Inteligência Artificial já mostra uma dificuldade dentro da própria definição de inteligência. Muitas perguntas controversas são feitas sobre inteligência e as respostas dependem muito da bibliografia e do ponto de vista adotados (FRANCO, 2014).

Outro ponto controverso é a qual área de conhecimento se encaixaria a IA: Filosofia? Matemática? Engenharia? Biologia? Ciência da Computação? Uma resposta bem aceita nos dias de hoje é a de que o campo da Inteligência Artificial é uma ciência multidisciplinar. É considerada essencialemente parte da Ciência da Computação, uma vez que o computador foi o escolhido para implementar ou implantar a inteligência (FRANCO, 2014).

As definições de IA se diferem em duas características principais: o pensamento/raciocínio e o comportamento. Nos últimos anos a IA utiliza-se das teorias existentes como base. O conceito mais aceito atualmente para definição de IA é de um agente inteligente, no qual abordagens simbólicas e conexionistas podem resolver os problemas de forma colaborativa através de um sistema computacional (FRANCO, 2014).

Um sistema capaz de aprender é essencial para se obter um comportamento inteligente. Quesitos como memorizar, observar e explorar situações para compreender fatos, melhorar habilidades de aplicação por meio de práticas e organizar o novo conteúdo em representações apropriadas cabem nas atividades consideradas relacionadas ao aprendizado (FACELI; LORENA; CARVALHO, 2011).

A Inteligência Artificial possui diversas subdivisões, sendo uma delas o Aprendizado de Máquina - AM. Para Mitchell (1997) a AM é definida como (FACELI; LORENA; CARVALHO, 2011):

“A capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência.”

Em AM, computadores são programados para aprender com eventos anteriores. Com este intuito, empregam o princípio de inferência denominado Indução. A Teoria de Inferência Indutiva Universal de Solomonoff é uma teoria de predição baseada em observações lógicas, onde a única suposição é de que o ambiente segue alguma distribuição probabilística desconhecida mas computável. Para (ANGLUIN; SMITH, 1983), a inferência indutiva pode ser usada para fornecer modelos abstratos do processo de investigação

científica ou do processo pelo qual uma criança adquire sua língua nativa.

Do exposto, algoritmos de AM analisam, extraem padrões e aprendem para induzir uma função ou hipótese capaz de resolver um problema a partir de informações que representam exigências do problema. Este conjunto de exigências (instâncias) representam o conjunto de dados (ANGLUIN; SMITH, 1983).

A seguir serão apresentadas duas técnicas de AM - Redes Neurais Artificiais - RNA - e Máquina de Vetores de Suporte - SVM - com o objetivo de implementar a detecção de curto-circuito em Sistemas de Potência.

2.2.2 Redes Neurais Artificiais

Modelos de RNA tem sua inspiração no cérebro. Existem cientistas cognitivos e neurocientistas focados na comprensão do funcionamento do cérebro. Nesta direção, esses cientistas pretendem construir modelos de redes neurais naturais do cérebro e fazer estudos de simulação (ALPAYDIN, 2020). Para a Ciência da Computação, para a Engenharia, etc, o objetivo não é entender o cérebro em si, mas construir modelos/máquinas úteis. Nestes ramos do conhecimento o interesse em redes neurais artificiais está na hipótese de que estas podem ajudar a construir melhores sistemas computacionais (ALPAYDIN, 2020).

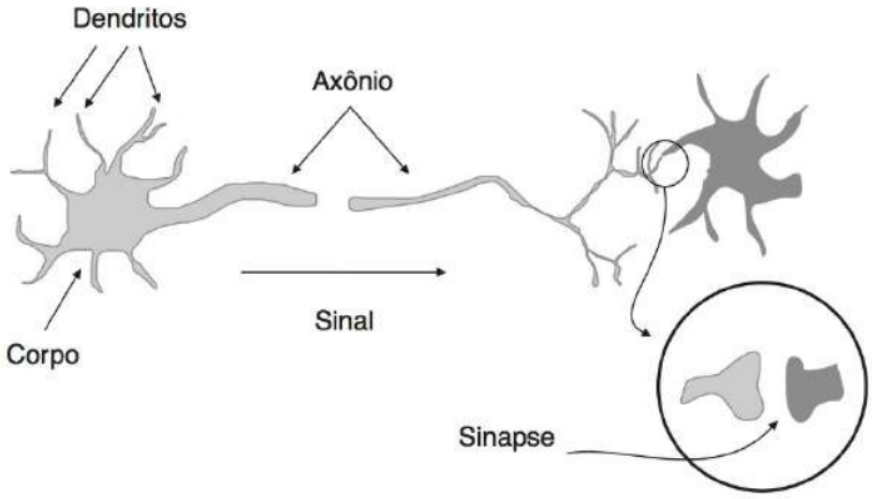
Sistema Nervoso

O sistema nervoso, que possui como um dos seus componentes o cérebro, é um agrupamento complexo de células encarregadas pelo funcionamento e comportamento dos seres vivos. O cérebro é observado em todos os seres vivos vetebrados e em muitos invertebrados. A menor unidade do sistema nervoso é um neurônio. Um neurônio é uma célula nervosa que se difere das demais por possuir a característica de excitabilidade e essa excitabilidade permite que estas células nervosas possam reagir a estímulos externos e internos. A reação a estimulos permite com que neurônios transmitam impulsos nervosos entre si (FACELI; LORENA; CARVALHO, 2011).

Um neurônio é composto por dentritos, corpo celular e axônio. Um esquema de neurônio é mostrado na Figura 13. Os neurônios se diferem entre si de várias formas, podendo assim assumir diferentes estruturas e não serão detalhadas neste trabalho. Os dentritos são prolongamentos dos neurônios que possuem a função de recepcionar os estímulos de outros neurônios e do ambiente; o dentrito se conecta ao corpo celular ou soma, e esse coleta as informações, as combina e processa. Caso a intensidade e frequência do sinal seja grande o suficiênte, o corpo celular pode gerar um novo sinal que é transmitido para o axônio. O axônio é um prolongamento dos neurônios ao qual é atribuida a tarefa de conduzir os sinais elétricos produzidos na soma para outro neurônio (ou local) mais afastado. A conexão entre neurônios é realizada pelas sinapses, e estas respondem de forma excitatória ou inibitória (FACELI; LORENA; CARVALHO, 2011). A Figura 13

exemplifica o exposto.

Figura 13 – Modelo Biológico Simplificado



Fonte: (FACELI; LORENA; CARVALHO, 2011)

A quantidade de neurônios no cérebro humano é grande e na ordem de 10 a 500 bilhões. Avalia-se que estes neurônios se organizam em 1000 módulos principais, cada um com 500 redes neurais. Cada neurônio pode estar conectado a centenas e até milhares de outras células neuronais. Essas redes naturais trabalham paralelamente de forma massiva, promovendo uma grande rapidez de processamento (FACELI; LORENA; CARVALHO, 2011).

2.2.2.1 Composição das RNAs

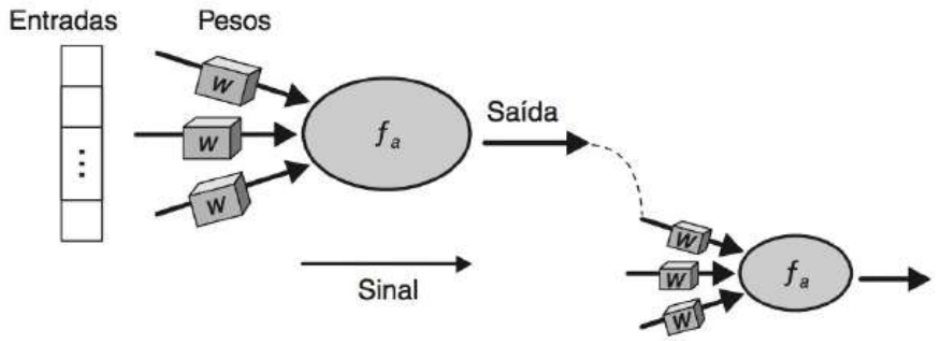
As Redes Neurais Artificiais são sistemas computacionais distribuídos formados por centros de processamento simples e densamente interconectados. Esses centros de processamento, conhecidos como neurônios artificiais, realizam funções matemáticas. Os centros ou unidades são dispostos em uma ou em várias camadas e são interligadas entre si por um grande número de conexões; as conexões são geralmente unidirecionais. A maior parte das configurações das redes neurais possuem pesos que ponderam o sinal de entrada em cada neuronio da rede. Os pesos podem ser positivos e negativos; pesos positivos são considerados excitatórios e pesos negativos são considerados inibitórios. Os pesos são ajustados no processo de aprendizado e descrevem o conhecimento adquirido pela rede (FACELI; LORENA; CARVALHO, 2011).

Arquitetura

O perceptron é o centro de processamento fundamental de uma RNA. Na Figura 14 é apresentada uma forma simplificada de um neurônio artificial. Os centros de processamento executam uma atividade simples: cada entrada do neurônio, como ocorre nos dentritos, recebe um sinal/valor de entrada; estes valores são ponderados e combinados em uma função matemática f_a, sendo que esta ponderação e combinação equivalem a uma operação de soma. O resultado deste processo é a resposta do neurônio ao sinal de entrada. A função f_a pode assumir várias formas sendo que, para serem compreendidas, algumas suposições são necessárias: suponha um objeto $\mathbf{x}$ com d atributos na forma vetorial $\mathbf{x} = [x_1, x_2, ..., x_d]^t$ e um neurônio com d pontos de entrada ponderados pelos atributos $w_1, w_2, ..., w_d$, que na forma vetorial é $\mathbf{w} = [w_1, w_2, ..., w_d]$. A entrada completa fornecida ao neurônio, v, pode ser definida pela Equação 2.33 (FACELI; LORENA; CARVALHO, 2011):

$$v = \sum_{j=1}^{d} x_j w_j \tag{2.33}$$

Figura 14 – Modelo de Rede Neural Artificial



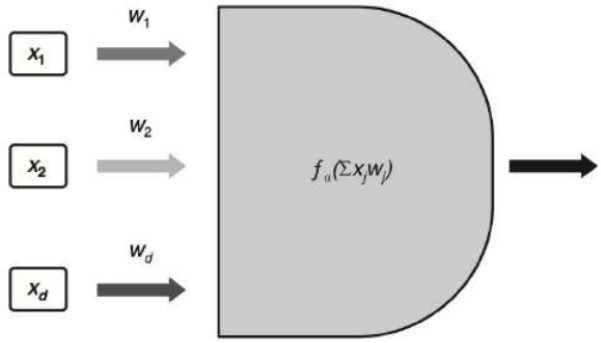
Fonte: (FACELI; LORENA; CARVALHO, 2011)

Os pesos neuronais w_j podem assumir valores maiores ou menores do que zero. O valor igual a zero indica ausência de conexão (FACELI; LORENA; CARVALHO, 2011).

Função de Ativação

A resposta de um neurônio é definida aplicando-se uma Função de Ativação - f_a - aos valores de entrada $\mathbf{x}$ (FACELI; LORENA; CARVALHO, 2011). As funções de ativação são elementos-chave nas camadas das redes neurais e são responsáveis por introduzir não-linearidade nas saídas dos neurônios para que a rede possa aprender mais do que relações lineares entre as variáveis dependentes e independentes. Estas funções são indispensáveis para fornecer capacidade representativa às redes neurais (FACURE, 2012). A Figura 15 mostra uma representação da função de ativação sendo aplicada a entrada total do neurônio.

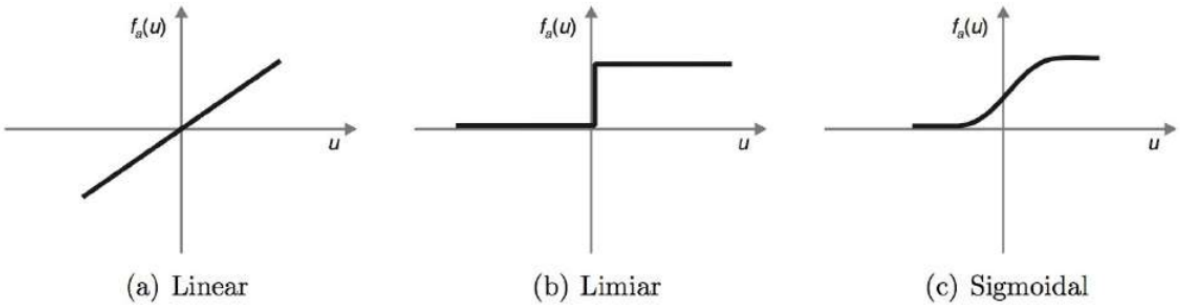
Figura 15 – Entrada total em um neurônio



Fonte: (FACELI; LORENA; CARVALHO, 2011)

A Figura 16 apresenta o formato de três funções de ativação presentes na literatura. A função linear irá retornar o valor da abscissa (na imagem representada pelo eixo u) como saída. A função limiar proposta por McCulloch e Pitts (1943) irá determinar quando a saída do neurônio será igual a 0 ou 1. No momento em que a soma dos valores recebidos são maiores que um valor pré-definido (o limiar), o neurônio torna-se ativo (+1). A função sigmoidal redefine a função limiar em uma aproximação continua e diferenciável (FACELI; LORENA; CARVALHO, 2011). Esta função de ativação será melhor detalhada na Seção 3.

Figura 16 – Funções de Ativação



(a) Linear (b) Limiar (c) Sigmoidal

Fonte: (FACELI; LORENA; CARVALHO, 2011)

Aprendizado

Existem vários algoritmos na literatura para ajuste dos parâmetros em uma RNA. O ajuste de parâmetros se refere principalmente a definição dos pesos relativos a cada conexão da rede com o objetivo de se alcançar um melhor desempenho. Estes algoritmos são conhecidos como algoritmos de treinamento e são constituídos por um grupo de regras estritamente definidas que determinam quando e como devem ser atualizados cada peso. Na literatura observam-se algoritmos de treinamento de RNAs seguindo os paradigmas de aprendizado supervisionado, não supervisionado e por reforço (FACELI; LORENA; CARVALHO, 2011).

Vários algoritmos de treinamento foram desenvolvidos para o grande número de arquiteturas criadas. O foco deste trabalho está em modelos e algoritmos do paradigma de aprendizado supervisionado.

A primeira RNA criada foi a rede perceptron desenvolvida por Rosenblatt (1958) e ela possui a forma próxima a apresentada na Figura 14. Mesmo sendo uma rede simples, com apenas uma camada de neurônios, ela apresentou uma boa acurácia preditiva em problemas de classificação diversos (FACELI; LORENA; CARVALHO, 2011).

A rede perceptron é treinada utilizando um algoritmo supervisionado de correção de erros e usa uma função de ativiação do tipo limiar. Neste algoritmo, para um objeto de entrada x_i os pesos serão alterados conforme a Equação 2.34 (FACELI; LORENA; CARVALHO, 2011):

$$w_j(t+1) = w_j(t) + \eta x_i^j (y_i - \hat{f}(x_i)) \tag{2.34}$$

Na Equação acima apresentada tem-se que: $w_j(t)$ é o peso da *j-ésima* conexão de entrada para o instante de tempo t; η é uma taxa de aprendizado; x_i^j é o valor do *j-ésimo* valor do vetor de entrada x_i; $\hat{f}(x_i)$ é a saída produzida pela RNA no instante de tempo t; y_i é a saída esperada para a rede - conhecida também como o rótulo de x_i. A taxa de aprendizado irá definir a magnitude da alteração realizada nos pesos. Valores altos para a taxa de aprendizado farão com que as variações nos pesos sejam grandes, enquanto taxas pequenas implicam em menores variaçõs nos pesos da rede (ALPAYDIN, 2020). A seguir é apresentado o algoritmo de treinamento.

Algoritmo de treinamento da rede perceptron

Entrada

– Um conjunto de n objetos de treinamento

Saida

– Rede perceptron com valores dos pesos ajustados

• Inicializar pesos da rede com valores baixos

• Repita

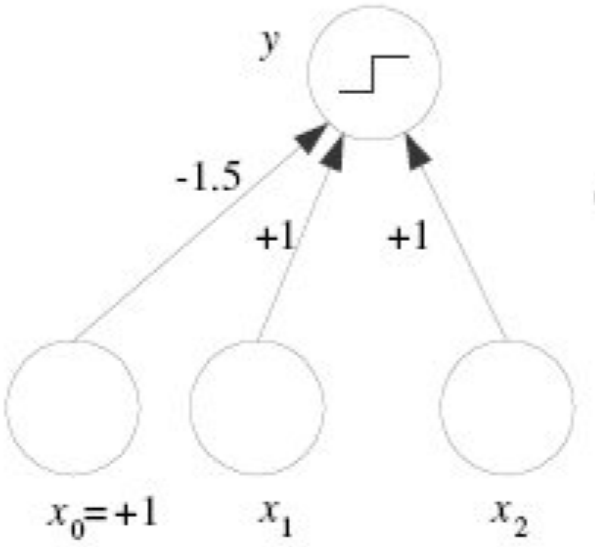
- **para cada** objeto x_i do conjunto de treinamento **faça**
  - Calcular valor da saída produzida pelo neurônio, $\hat{f}(x_i)$
  - Calcular erro $= y_i - \hat{f}(x_i)$
  - **se** erro > 0 **então**
    - Ajustar pesos do neurônio utilizando a Equação 2.34
  - **fim**
- **fim**

– **até** erro 0;

Multilayer Perceptrons

Um perceptron que possui uma única camada de pesos pode somente aproximar funções linearmente separáveis como OR e AND como dados de entrada - Figura 17.

Figura 17 – Representação de um perceptron



(ALPAYDIN, 2020)

Para funções não lineares, como problemas de classificação envolvendo $OR\ Exclusivo$ - XOR, um único perceptron não irá conseguir solucionar o problema. De maneira similar, somente um perceptron não poderá ser usado para regressão não linear. Esta limitação não se aplica para redes retroalimentadas com uma camada intermediária ou camada escondida entre as camadas de entrada e saída: uma Multilayer Perceptron (MLP). Estas redes retroalimentadas com várias camadas poderão resolver os problemas de implementação de discriminantes não lineares para classificação e de aproximação não linear de funções (ALPAYDIN, 2020).

As redes neurais se iniciam com a proposta de uma camada oculta. Esta camada oculta é generalizada para redes com multiplas camadas escondidas e culmina com as redes neurais profundas (ALPAYDIN, 2020).

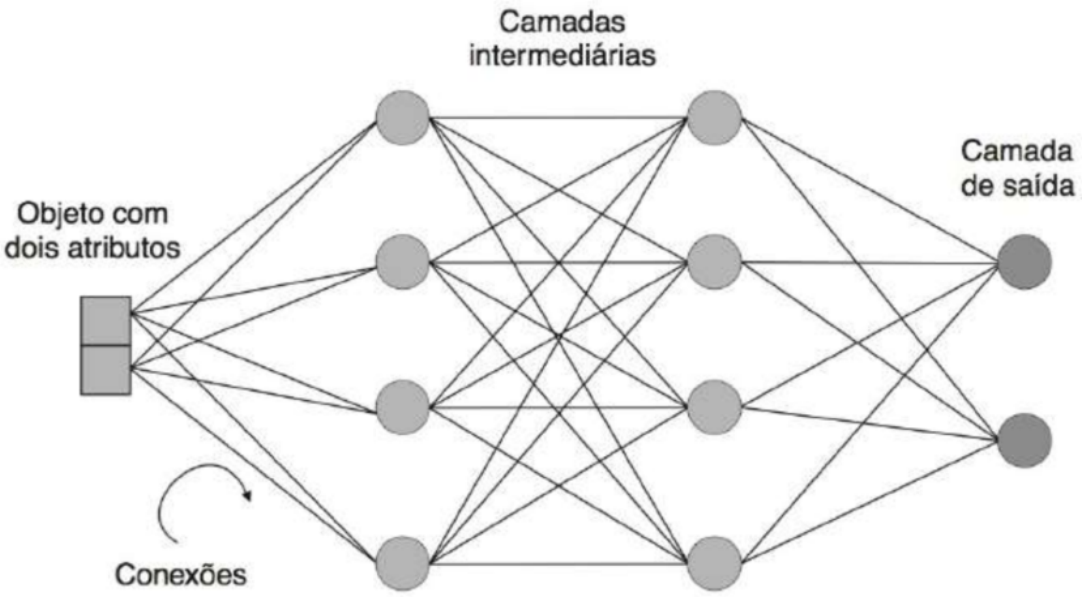
Para descrever o funcionamento de uma MLP admite-se que x alimenta a camada de entrada incluindo também um BIAS ou VIÈS (aqui considerado como sendo w_{i0}). A camada de entrada irá transmitir os dados inseridos para camadas ocultas conectadas a frente e os valores das unidades escondidas z_h serão calculados. Cada unidade oculta é um perceptron por si só e aplica uma função de ativação não linear (aqui será apresentada a função sigmóide) à sua soma ponderada (ALPAYDIN, 2020):

$$z_h = sigmoid(w_h^T x) = \frac{1}{1 + exp[-(\sum_{j=1}^{d} w_{hj} x_j + w_{h0})]}, h = 1, ..., H \tag{2.35}$$

As saídas y_i serão perceptrons na camada final recebendo como entrada a saída das camadas ocultas ponderadas pelos respectivos pesos v_{ih}. Existirá também a unidade de viés na camada escondida que será definida como z_0 e uma unidade de viés para os pesos v_{i0}. A Equação 2.36 definide a expressão para y_i. A camada de entrada não é descrita na expressão pois nenhum calculo é realizado nessa quando existe uma ou mais camadas ocultas (ALPAYDIN, 2020).

$$y_i = v_i^T z = \sum_{h=1}^{H} v_{ih} z_h + v_{i0} \tag{2.36}$$

Figura 18 – Multilayer Perceptron



(FACELI; LORENA; CARVALHO, 2011)

A Figura 18 apresenta uma MLP.

2.2.2.2 Algoritmo Back-Propagation

O algoritmo back-propagation é baseado na regra delta generalizada e é estruturado com a iteração de duas etapas. A primeira etapa é a fase para a frente (*forward*) e a segunda etapa para trás (*backward*). Na primeira etapa *forward* o objeto x_j é apresentado a rede sendo imediatamente ponderado pelos pesos respectivos w_{1j} da camada de entrada da rede. Cada neurônio da camada aplica sua função de ativação ao resultado da soma das suas entradas ponderadas. O resultado obtido após a aplicação da função de ativação será utilizado como entrada para os neurônios da próxima camada. O processo continuará por todas as camadas ocultas até a saída, onde será produzido o resultado. A diferença entre o resultado produzido pela rede e o apresentado a rede - para cada neurônio da camada de saída - indica o erro cometido nesta etapa (FACELI; LORENA; CARVALHO, 2011).

Na segunda etapa do algoritmo, chamada de *backward*, o erro calculado é utilizado para realizar a atualização de todos os pesos de entrada das camadas da rede. O ajuste será realizado da camada de saída até a primeira camada escondida intermediária. A Equação 2.37 descreve como deverá ser feito o ajuste de uma MLP através do algoritmo back-propagation (FACELI; LORENA; CARVALHO, 2011):

$$w_{jl}(t+1) = w_{jl}(t) + \eta x^j \delta_l \tag{2.37}$$

Na expressão apresentada, w_{jl} indica o peso referente a um neurônio l e a *j-ésima* conexão de entrada, δ_l indica o erro relacionado ao *l-ésimo* neurônio e x^j indica o valor de entrada recebido por um neurônio.

A diferença dos valores produzidos e os desejados, na etapa de *foward*, retorna o erro da saida da MLP. O erro para as camadas intermediárias será calculado pelo algoritmo back-propagation aproveitando o resultado do erro calculado para a camada posterior. A proposta do algoritmo é que o erro da camada em análise será a soma dos erros da camada seguinte ponderado pelos valores do peso associado à cada ramo de conexão da rede. A forma de encontrar o erro está diretamente relacionada à camada em que o neurônio se encontra. A expressão para o cálculo do erro para as camadas internas da rede está definida na Equação 2.38 (FACELI; LORENA; CARVALHO, 2011):

$$\delta_l = \begin{cases} f_a^{'} e_l & se \quad n_l \quad \epsilon \quad c_{sai} \\ f_a^{'} \sum w_{lk} \delta_k & se \quad n_l \quad \epsilon \quad c_{int} \end{cases} \tag{2.38}$$

onde o *l-ésimo* neurônio é representado por n_l, a camada de saída é representada por c_{sai}, a camada intermediária é representada por c_{int}, a derivada parcial da função de ativação é apresentada como sendo $f_a^{'}$, k é o número de neurônios na camada de saída e

e_l é o erro quadrático no neurônio de saída da rede conforme a Equação 2.39 (FACELI; LORENA; CARVALHO, 2011):

$$e_l = \frac{1}{2}\sum_{q=1}^{k}(y_q - \hat{f}_q)^2 \tag{2.39}$$

A derivada parcial $f_a^{'}$ computa a variação dos pesos, utilizando o gradiente descente da função de ativação. A derivada irá calcular a influência de cada peso no erro da rede: caso ela seja positiva, indica que está aumentando o erro da rede e o seu vaor deverá ser reduzido; caso contrário estará reduzindo o erro da rede, e o seu valor deverá ser aumentado (FACELI; LORENA; CARVALHO, 2011).

O valor de η - taxa de aprendizado - influencia fortemente o tempo de convergência da rede. Para taxas de aprendizado muito pequenas, pode se necessitar de muitos ciclos de iteração do algoritmo para produzir uma boa resposta. Do contrário, taxas de aprendizado muito altas dificultam a convergência da rede. Uma medida possível para contornar esse problema é acrescentar o conceito de *momentum* α (RUMELHART; MCCLELLAND, 1986). O momentum irá quantificar o grau de importância da variação do peso do ciclo anterior ao ciclo atual. A Equação 2.40 apresenta esta alteração na atualização dos pesos da rede (FACELI; LORENA; CARVALHO, 2011).

$$w_{jl}(t+1) = w_{jl}(t) + \eta x^j \delta_l + \alpha(w_{jl}(t) - w_{jl}(t-1)) \tag{2.40}$$

Onde x^j representa o valor do *j-ésimo* elemento de **x** ou a saída do **j-esimo** neurônio da camada anterior.

O algoritmo apresentado a seguir traz o passo a passo do da técnica de back-propagation (ALPAYDIN, 2020):

Algoritmo de treinamento back-propagation

Entrada

– Um conjunto de n objetos de treinamento

Saida

– Rede MLP com valores de peso ajustados

• Inicializar pesos da rede com valores aleatórios

• Inicializar $erro_{total} = 0$

• Repita

- **para cada** objeto x_i do conjunto de treinamento **faça**
  - **para cada** camada da rede, a partir da primeira camada intermediária **faça**
    - **para cada** neurônio n_{jl} da camada atual **faça**
      - Calcular valor da saída produzida pelo neurônio, $\hat{f}$
    - **fim**
  - **fim**
  - Calcular $erro_{parcial} = y - \hat{f}$
    - **para cada** camada da rede, a partir da camada de saída **faça**
    - **para cada** neurônio n_{jl} da camada atual **faça**
      - Ajustar os pesos do neurônio utilizando a Equação 2.37
    - **fim**
  - **fim**
  - Calcular $erro_{total} = erro_{total} + erro_{parcial}$
- **fim**

• **até** $erro_{total} < \epsilon$;

2.2.3 Máquina de Vetores de Suporte

As Máquinas de Vetores de Suporte são um ramo do Aprendizado de Máquina onde é aplicada a teoria de aprendizado estatístico de autoria de (VAPNIK, 1995), teoria esta iniciada anteriormente em (VAPNIK; CHERVONENKIS, 1971). A teoria de aprendizado estatístico define um conjunto de princípios que precisam ser executados para a obtenção de classificadores que generalizem bem (FACELI; LORENA; CARVALHO, 2011).

Teoria do Aprendizado Estatístico

Dado h um classificador e H o agrupamento de todos os possíveis classificadores gerados por um determinado algoritmo de AM. Um conjunto de treinamento $\mathbf{X}$ é utilizado durante o processo de aprendizado e é composto por n pares (x_i, y_i) com o objetivo de gerar um classificador particular $\hat{h} \in H$ (FACELI; LORENA; CARVALHO, 2011).

A Teoria de Aprendizado Estatístico (TAE) fixa condições matemáticas para auxiliar na escolha de um classificador em particular $\hat{h}$ a partir de um dataset de treinamento. As condições matemáticas definidas consideram o desempenho do classificador no conjunto

de treinamento e a sua complexidade. O objetivo aqui é alcançar desempenho satisfatório para novos dados no mesmo domínio (FACELI; LORENA; CARVALHO, 2011).

Será assumido que inicialmente os dados do domínio são gerados de forma independente e identicamente distribuídos (i.i.d.) de acordo com uma distribuição de probabilidade $P(x, y)$. Essa função de probabilidades descreverá a relação entre os objetos e o seus rótulos. O erro esperado(risco) para um classificador h para todos os dados deste domínio pode então ser mensurado pela Equação 2.41 e mede a capacidade de generalização de h. Na Equação 2.41, a expressão c(h(x),y) é uma função de custo ligada a previsão $h(x)$ à saída desejada y, onde $y_i \in -1, +1$. A função de custo comum em problemas de classificação é a 0-1. Esta função retorna 0 se **x** for classificado corretamente e 1 se não for classificado corretamente - Equação 2.42 - (FACELI; LORENA; CARVALHO, 2011).

$$R(h) = \int c(h(x), y) dP(x, y) \tag{2.41}$$

$$c(h(x), y) = \frac{1}{2}|y - h(x)| \tag{2.42}$$

Não é possível minimizar o risco esperado ***R(h)*** diretamente da expressão uma vez que normalmente a distribuição de probabilidades ***P(x,y)*** é desconhecida. O que se tem para análise são as informações do dataset, também amostrados de ***P(x,y)***. Usualmente se utiliza o princípio da indução para inferir uma função $\hat{h}$ que reduza ao máximo o erro sobre o dataset e supõe-se que este processo resulte também em um menor erro sobre novos dados. Essa é a estratégia utilizada por grande parte das técnicas de AM supervisionadas. O risco emprírico de h calcula os resultados do classificador nos dados de treinamento através da taxa de classificações incorretas obtidas de **X** e é apresentada na Equação 2.43.

$$R_{emp}(h) = \frac{1}{n} \sum_{i=1}^{n} c(h(x_i), y_i) \tag{2.43}$$

Esse processo de indução com base no dataset de treinamento apresentado estabelece o princípio de minimização do risco empírico (SMOLA; SCHOLKOPF, 2002). Assintoticamente, com $n \to \infty$, é factível fixar condições para o algoritmo de aprendizado que assegure a obtenção de classificadores cujos resultados de risco empírico convergem para o risco esperado (Muller et al, 2001). Para dados de treinamento menores normalmente não e possível determinar este tipo de garantia (FACELI; LORENA; CARVALHO, 2011).

A ideia expressa no conteúdo apresentado é a de que, ao admitir que $\hat{h}$ seja escolhida baseado em um grupo de funções H, é sempre possível alcançar uma h com o menor risco empírico. No entanto, nesse caso, o conjunto de exemplo de treinamento pode se tornar pouco informativo para o trabalho de aprendizado - o classificador induzido pode se superajustar ao exemplo. Uma estratégia para lidar com este problema é limitar a classe de

funções de onde $\hat{h}$ é adquirida. A TAE resolve esta questão considerando a complexidade da classe de funções - ou complexidade - que o algoritmo de aprendizado é capaz de induzir (SMOLA; SCHOLKOPF, 2002). Assim, a TAE fornece vários limites no risco esperado de uma função de classificação, limites esses que podem ser utilizados na escolha do classificador (FACELI; LORENA; CARVALHO, 2011).

2.2.3.1 SVMs Lineares

As SVMs são o resultado direto da aplicação das conclusões fornecidas pela TAE. A primeira concepção, bem simples, trata de problemas linearmente separáveis (BOSER; I.L.; VAPNIK, 1992). A formulação proposta por Boser foi ulteriormente ampliada e passou a definir fronteiras lineares sobre conjuntos de dados mais genéricos (CORTES; VAPNIK, 1995). A partir das proposições inicias que serão descritas, as SVMs evoluem para o tratamento de dados não lineares (FACELI; LORENA; CARVALHO, 2011).

SVMs com Margens Rígidas

As SVMs lineares com margens rígidas fixam limites lineares a partir de um conjunto de dados linearmente separáveis. Dado $\mathbf{X}$ sendo um dataset de treinamento com n objetos $x_i \in X$ e seus rótulos relacionados $y_i \in Y$, em que X compõe o espaço de entrada e $Y = -1, +1$ são as prováveis classes. X será linearmente separável se for plausível a separação dos objetos das classes +1 e -1 por um hiperplano (FACELI; LORENA; CARVALHO, 2011).

Denominam-se classificadores lineares aqueles que distinguem os dados por meio de um hiperplano. É apresentada na Equação 2.44 a expressão para um hiperplano onde $w.x$ representa o produto escalar entre os vetores $\mathbf{w}$ e $\mathbf{x}$, $w \in X$ é o vetor normal ao hiperplano proposto e $\frac{b}{||w||}$ equivale a distância do hiperplano à origem, com $b \, \epsilon \, \Re$

$$h(x) = w.x + b \tag{2.44}$$

A Equação acima pode ser usada para separar o espaço de entrada $\mathbf{X}$ em duas regiões de forma que $w.x + b > 0$ e $w.x + b < 0$. Para a obtenção das classificações aplica-se uma função sinal $g(x) = sgn(h(x))$. A Equação 2.45 ilustra a função (FACELI; LORENA; CARVALHO, 2011).

$$g(x) = sgn(h(x)) = \begin{cases} +1 & se \quad w.x + b > 0 \\ -1 & se \quad w.x + b < 0 \end{cases} \tag{2.45}$$

A partir de $h(x)$ é admissível encontrar um número infinito de hiperplanos equivalentes a partir do produto entre w e b por uma constante comum. Assim representa-se o hiperplano canônico em relação aos dados $\mathbf{X}$ como o conjunto em que w e b são escalados

de forma que os objetos mais próximos ao hiperplano $w.x + b = 0$ atendam a Equação 2.46 (Muller et al, 2001) (FACELI; LORENA; CARVALHO, 2011).

$$|w.x + b| = 1 \tag{2.46}$$

Essa forma culmina nas Inequações 2.47 e em sua versão resumida 2.48 (FACELI; LORENA; CARVALHO, 2011).

$$\begin{cases} w.x + b \geq +1 & se \quad y_i = +1 \\ w.x + b \leq -1 & se \quad y_i = -1 \end{cases} \tag{2.47}$$

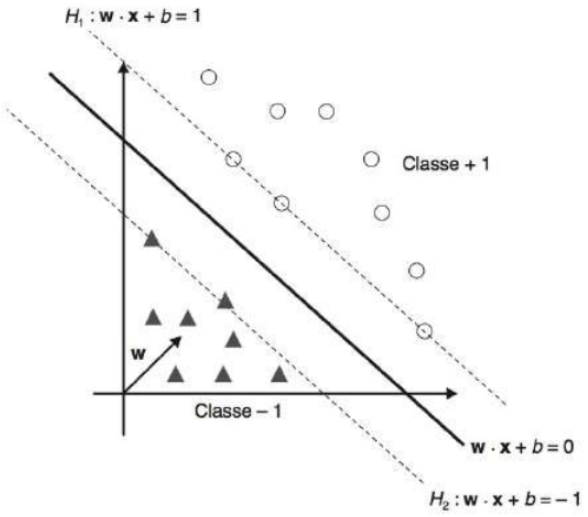
$$y_i(w.x_i + b) - 1 \geq 0, \forall (x_i, y_i) \epsilon X \tag{2.48}$$

Após alguns tratamentos algébricos simples utilizando pontos sobre os hiperplanos H_1 e H_2, representados na Figura 19, pode-se deduzir que a maximização da margem de separação dos objetos em relação a $w.x+b = 0$ pode ser encontrada a partir da minimização de $||w||$ (Campbell, 2000). Assim, utiliza-se o seguinte problema de otimização (FACELI; LORENA; CARVALHO, 2011):

$$\text{Minimizar}_{w,b} \ \frac{1}{2}||w||^2 \tag{2.49}$$

$$\text{Com as restrições: } y_i(w.x_i + b) - 1 \geq 0, \forall i = 1, ..., n \tag{2.50}$$

Figura 19 – Hiperplanos canônicos e separador.



(FACELI; LORENA; CARVALHO, 2011)

As restrições são colocadas de forma que não existam dados de treinamento entre as margens que separam as classes. Por esta razão, a SVM obtida é chamada de SVM com margens rígidas (FACELI; LORENA; CARVALHO, 2011).

Como o problema de otimização é quadrático, a sua resolução possui vasta e estabelecida teoria matemática. A função objetivo a ser minimizada é convexa e os pontos que satisfazem os limites colocados formam um conjunto convexo; o problema possui um único mínimo global. A questão em análise poderá ser resolvida com a introdução de uma função lagrangiana com as limitações da função objetivo associadas aos multiplicadores de Lagrange α_i. A expressão resultante é descrita a seguir na Equação 2.51 (FACELI; LORENA; CARVALHO, 2011).

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i (y_i(w.x_i + b) - 1) \tag{2.51}$$

A expressão lagrangeana deve ser minimizada e isto resulta maximizar a variável α_i, enquanto $\mathbf{w}$ e b serão minimizadas. A derivada de L em relação a $\mathbf{w}$ e b igualada a zero implica nas expressões abaixo apresentadas (FACELI; LORENA; CARVALHO, 2011).

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.52}$$

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{2.53}$$

Levando as expressões das Equações 2.52 e 2.53 para a Equação 2.51, chega-se ao equacionamento a seguir (FACELI; LORENA; CARVALHO, 2011):

$$\text{Maximizar}_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i.x_j) \tag{2.54}$$

$$\begin{cases} \alpha_i \geq 0, & \forall i = 1, ..., n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{2.55}$$

A partir de alguns tratamentos matemáticos, obtém-se como resultado final o classificador g(x) apresentado a seguir (FACELI; LORENA; CARVALHO, 2011):

$$g(x) = sgn(h(x) = sgn\Big(\sum_{x_i \in SV} y_i \alpha_i^* i x_i.x + b^* \Big) \tag{2.56}$$

Onde α_i^* e b^* são definidos como (FACELI; LORENA; CARVALHO, 2011):

$$\alpha_i^* (y_i(w^*.x_i + b^*) - 1) = 0, \ \forall i = 1, ..., n \tag{2.57}$$

$$b^* = \frac{1}{n_{SV}} \sum_{x_j \in SV} \left(\frac{1}{y_i} - \sum_{x_i \in SV} \alpha_i^* y_i x_i . x_j \right) \tag{2.58}$$

onde n_{SV} refere-se ao número de SVs e SV denota o conjunto de SVs.

SVMs com Margens Suaves

Em problemas reais, torna-se difícil encontrar aplicações cujos dados sejam linearmente separáveis. AS SVMs lineares, definidas na seção anterior, podem ser ampliadas para conseguir resolver problemas mais gerais. Nesta linha, aceita-se que alguns dados possam desrespeitar os limites definidos na Equação 2.48 e isto será feito introduzindo variáveis de folga ξ para todo $i = 1, ..., n$. Essas variáveis irão relaxar os limites impostos a função de otimização inicial, resultando em (SMOLA; SCHOLKOPF, 2002):

$$y_i(w.x_i + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \ \forall i = 1, ..., n \tag{2.59}$$

Insere-se um erro no conjunto de treinamento e este é indicado por um valor de ξ_i maior que 1. Assim, a soma dos ξ_i significará um limite a quantidade de erros de treinamento. (Burges,1998) reformulou o equacionamento para a função objetivo da seguinte forma:

$$\text{Minimizar}_{w,b,\xi} \ \frac{1}{2}||w||^2 + C\left(\sum_{i=1}^{n} \xi_i \right) \tag{2.60}$$

A constante C é um parâmetro de regularização que determina um peso à minimização dos erros nos dados de treinamento comparando-os com à minimização da complexidade do modelo. A existência do item $\sum_{i=1}^{n} \xi_i$ na questão colocada de otimização irá também ser analisada como um problema de minimização de erros marginais devido ao fato de que um valor de $\xi_i \in (0,1]$ aponta para um objeto entre as margens (FACELI; LORENA; CARVALHO, 2011).

De novo a questão de otimização criada é quadrática, com as restrições lineares descritas na Equação 2.59.A resolução da expressão de otimização inclui passos matemáticos análogos aos apresentados anteriormente, introduzindo-se uma função lagrangiana e fazendo suas derivadas parciais nulas. Assim, encontra-se a seguinte solução para o problema (FACELI; LORENA; CARVALHO, 2011):

$$\text{Maximizar}_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i . x_j) \tag{2.61}$$

$$\text{Com as restrições: } \begin{cases} 0 \leq \alpha_i \leq C, & \forall i = 1, ..., n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{2.62}$$

Observa-se que esta expressão é igual à proposta para SVMs de margens rígidas a menos da limitação de α_i, que agora será restringido pelo parâmetro C (FACELI; LORENA; CARVALHO, 2011).

Dado α^* a resposta do problema dual e a resposta do problema primal sendo w^*, b^* e ξ^*. O vetor w^* permanece sendo determinado pela expressão contida na Equação 2.53 (FACELI; LORENA; CARVALHO, 2011). As variáveis ξ^* serão calculadas pela equação 2.63 (CRISTIANINI; SHAWE-TAYLOR, 2000):

$$\xi = \max\left\{0, 1 - y_i \sum_{j=1}^{n} y_i \alpha_j^* x_j . x_i + b^*\right\} \tag{2.63}$$

A variável b^* origina-se da variável α^* e de condições de Kuhn-Tucker, que neste caso serão (FACELI; LORENA; CARVALHO, 2011):

$$\alpha_i^*(y_i(w^*.x_i + b^*) - 1 + \xi_i^*) = 0 \tag{2.64}$$

$$(C - \alpha_i^*)\xi_i^* = 0 \tag{2.65}$$

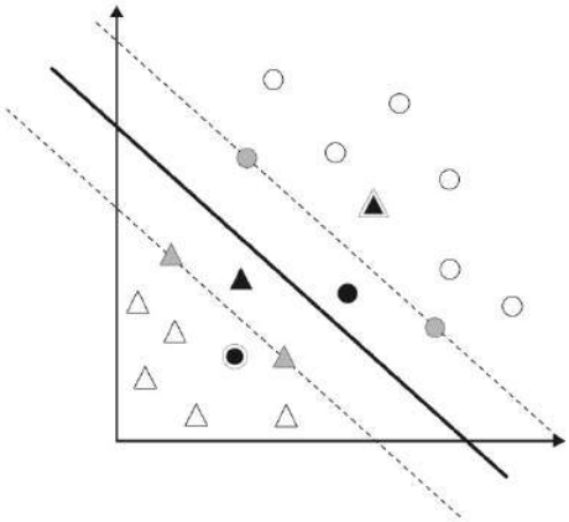
Como ocorrem nas SVMs de margem rígidas, os pontos x_i para os quais $\alpha_i^* > 0$ são chamados de vetores suporte (SVs) e são os objetos que compõem a formação do hiperplano separador. No entanto, para SVMs com margens suaves, existe a possibilidade de se classificar tipos diferentes de SVs. Se $\alpha_i^* < C$, dada a Equação 2.65, $\xi_i^* = 0$ e daí, avaliando a Equação 2.64 conclui-se que os vetores suporte estão sobre as margens e também são entitulados de livres. Os SVs com $\alpha_i^* = C$ irão se dividir em três casos distintos (FACELI; LORENA; CARVALHO, 2011):

1. Se $\xi_i^* > 1$: pontos classificados de maneira correta;
2. Se $0 < \xi_i^* \leq 1$: pontos classificados entre as margens;
3. Se $\xi_i^* = 0$: pontos classificados sobre as margens. Este caso é raro.

Os casos enumerados acima são nominados de SVs limitadas. A seguir são ilustradas as váriações de tipos de SVs na Figura 20. Objetos na cor cinza apresentam SVs livres; SVs limitados são colocados em preto. Objetos pretos com segunda borda representam SVs limitados que são erros de treinamento. Os demais objetos em branco estão corretamente classificados, estando então fora das margens e possuem $\xi_i^* = 0$ e $\alpha_i^* = 0$ (FACELI; LORENA; CARVALHO, 2011).

O calculo de b^* será realizado pela média da Equação 2.58 em todos os SVs x_j entre as margens, isto é, com $\alpha_i^* < C$). Obtém-se ao final a mesma função de classificação da

Figura 20 – Tipos SVs: limitados(preto) e livres(cinza).



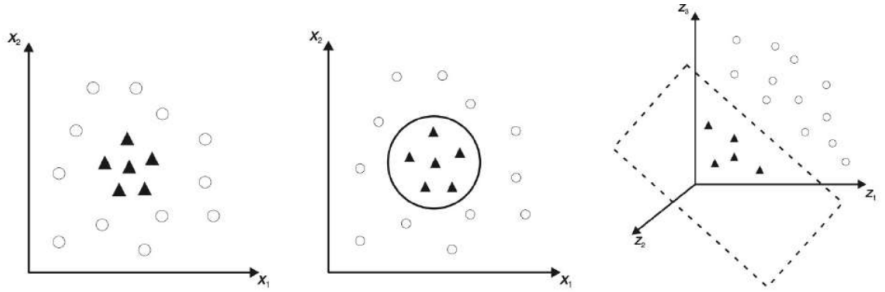
(FACELI; LORENA; CARVALHO, 2011)

Equação 2.56, mas neste caso as variáveis α_i^* são determinadas pela resposta da Equação 2.61.

2.2.3.2 SVMs Não Lineares

Nas duas seções anteriores foram apresentadas soluções para conjuntos e dados linearmente separáveis. No entanto, em muitos casos não é possível realizar a divisão dos dados por meio de um hiperplano. A Figura 21 representa o uso de uma fronteira não linear para a separação das classes (FACELI; LORENA; CARVALHO, 2011).

Figura 21 – Transformações de dados não linerares para o espaço de características.



(FACELI; LORENA; CARVALHO, 2011)

As SVMs resolvem as questões não lineares representando o conjunto de treinamento de seu espaço original - entradas - para um novo espaço de maior dimensão. Este novo espaço é entitulado *espaço de características* - *feature space* (Hearst et al, 1998). Dado um mapeamento $\phi : X \rightarrow \Im$, onde X é o espaço de entradas e $\Im$ denota espaço de

características. A seleção adequada de ϕ resulta em um conjunto de treinamento em $\Im$ que possa ser separado por uma SVM linear (FACELI; LORENA; CARVALHO, 2011).

O teorema de Cover, proposto por (HAYKIN, 1999), foi utilizado como motivação para este método. Fornecido um conjunto de dados não linear no espaço de entradas X, o teorema de Cover destaca que X pode ser transformado em um espaço de características $\Im$ onde existe alta probabilidade de os objetos serem linearmente separáveis. Para isto duas regras devem ser satisfeitas (FACELI; LORENA; CARVALHO, 2011):

1. A transformação deve ser não linear;
2. A dimensão do espaço de características seja suficientemente alta.

Considerando o conjunto de dados da Figura 21(a), ao transformar os objetos de $\Re^2$ para $\Re^3$ com o mapeamento apresentado na Equação 2.66, o conjunto de dados não linear em $\Re^2$ passa a ser linearmente separável em $\Re^3$ Figura 21(c). É factível então localizar um um hiperplano capaz de realizar a separação desses objetos, conforme a Equação 2.67. Observa-se que a função adotada é linear em $\Re^3$ e em $\Re^2$ equivale a uma função de fronteira não linear (FACELI; LORENA; CARVALHO, 2011).

$$\phi(x) = \phi(x_1, x_2) = \Big(x_1^2, \sqrt{2}x_1x_2, x_2^2\Big) \tag{2.66}$$

$$h(x) = w.\phi(x) + b = w_1x_1^2 + w_2\sqrt{2}x_1x_2 + w_3x_2^2 + b = 0 \tag{2.67}$$

Do exposto, pode-se concluir os seguinte: mapeiam-se primeiramente os objetos para um espaço de dimensão superior por meio de ϕ e utiliza-se a SVM linear sobre este espaço. A SVM irá encontrar um hiperplano com maior margem de separação, assegurando então uma boa generalização. Para o caso colocado, faz-se uso da SVM linear com margens suaves pois esta permite trabalhar com ruídos e outlier nos dados de treinamento. O mapeamento será realizado aplicando-se ϕ aos dados do problema de otimização descritos na expressão 2.61 de acordo com o demosntrado a seguir (FACELI; LORENA; CARVALHO, 2011):

$$\text{Maximizar}_\alpha \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j (\phi(x_i).\phi(x_j)) \tag{2.68}$$

Assim, o classificador extraído se torna:

$$g(x) = sgn(h(x) = sgn\Big(\sum_{x_i \in SV} \alpha_i^* y_i \phi(x_i).\phi(x) + b^* \Big) \tag{2.69}$$

Em que b^* é a adaptação da Equação 2.58 aplicando-se o mapeamento dos objetos:

$$b^* = \frac{1}{n_{SV_{:\alpha^*<C}}} \sum_{x_j \in SV_{:\alpha^*<C}} \left(\frac{1}{y_j} - \sum_{x_i \in SV} \alpha_i^* y_i \phi(x_i).\phi(x_j) \right) \tag{2.70}$$

Sabe-se que $\Im$ pode ter dimensão muito alta ou até mesmo infinita e isto irá tornar a computação de ϕ extremamente trabalhosa ou até impossível. No entanto, pelas Equações 2.68, 2.69 e 2.70, visualiza-se que somente uma informação é necessária em relação ao mapeamento que é de como aplicar o cálculo dos produtos escalares entre os objetos no espaço de características. Esse trabalhado é executado através de funções intituladas kernels (FACELI; LORENA; CARVALHO, 2011).

Um kernel K é uma função que admite como entrada dois pontos x_i e x_j e computa o produto escalar desses objetos no espaço de características (HERBRICH, 2001). Obtendo-se assim:

$$K(x_i, x_j) = \phi(x_i).\phi(x_j)) \tag{2.71}$$

Com exemplo, para o mapeamento expresso na Equação 2.66 e dois objetos x_i e x_j em $\Re^2$, o kernel é formulado como a seguir (FACELI; LORENA; CARVALHO, 2011):

$$K(x_i, x_j) = (x_i, x_j)^2 \tag{2.72}$$

É habitual aplicar a função kernel sem ter conhecimento do mapeamento ϕ, mapeamento este que que é gerado implicitamente. Assim, a aplicabilidade dos kernels estão na simplicidade de cálculo e em sua capacidade de representar espaços abstratos (FACELI; LORENA; CARVALHO, 2011).

Para garantir a convexidade do problema de otimização apresentado na Equação 2.68 e que o kernel forme mapeamentos onde seja possível a computação dos produtos escalares conforme a Equação 2.71, faz-se uso de funções kernel que obedeçam as restrições estabelecidas pelo teorema de Mercer (MERCER, 1909). Simplificadamente, as restrições de Mercer serão respeitadas quando um kernel é reconhecido por dar origem a matrizes positivas semidefinidas $\mathbf{K}$, em que cada componente K_{ij} é caracterizado por $K_{ij} = K(x_i, x_j)$, para todo $i, j = 1, ..., n$ (HERBRICH, 2001).

Os kernels mais utilizados no dia a dia são os polinomiais, os de função base radial (*radial basis function* - RBF) e os sigmoidais, conforme apresentados na Tabela 1. Nessa Tabela são apresentadas as expressões matemáticas de cada kernel e também são apresentados os parâmetros que devem ser prédeterminados pelos usuários.

A extração de um classificador fazendo uso de SVMs é constituído pelas fases de escolha de uma função kernel, escolha dos parâmetros dessa função e do valor da constante de regularização. A definição do kernel e dos parâmetros escolhidos afetam o desempenho

Tabela 1 – Funções kernel mais comuns.

Tipo de Kernel	Função $K(x_i, x_j)$	Parâmetros
Polinomial	$(\delta(x_i.x_j) + \kappa)^d$	δ, κ e d
RBF	$e^{(-\sigma\|\|x_i - x_j\|\|^2)}$	σ
Sigmoidal	$\tanh(\delta(x_i.x_j + \kappa)$	δ e κ

Fonte: (FACELI; LORENA; CARVALHO, 2011)

do classificador obtido, pois eles determinam a fronteira de decisão induzida (FACELI; LORENA; CARVALHO, 2011).

O Algoritmo apresentado a seguir descreve a formulação final seguida pelas SVMs na fase de treinamento:

Algoritmo de treinamento SVM

Entrada

– Um conjunto de n objetos de treinamento

Saida

– SVM treinada

- Seja $\alpha^* = (a_1^*, ..., a_n^*)$ a solução de:
- Maximizar $\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$
- Sob as restrições $\begin{cases} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, \qquad i = 1, ..., n \end{cases}$
- O classificador é dado por: $\mathrm{g(x)} = \mathrm{sgn(h(x))} = \mathrm{sgn}\left(\sum_{x_i \in SV} \alpha_i^* y_i K(x_i, x) + b^* \right)$
- Em que: $b^* = \frac{1}{n_{SV_{:\alpha^* < C}}} \sum_{x_j \in SV_{:\alpha^* < C}} \left(\frac{1}{y_j} - \sum_{x_i \in SV} \alpha_i^* y_i K(x_i, x_j) \right)$

.

3 MATERIAIS E MÉTODOS

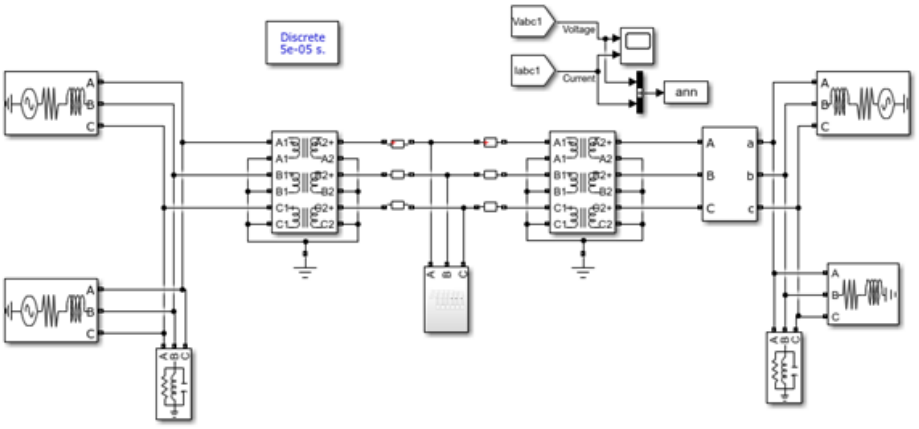
Neste capítulo serão apresentados os dados de treinamento/teste e as ferramentas adotadas para as soluções em RNA e SVM.

3.1 Conjunto de dados de treinamento e Tratamento dos Dados

Para o treinamento e teste da classificação de curto-circuito proposta foi adotado o conjunto de dados "Electric Faults - Detection and Classification"disponibilizado em (PRAKASH, 2021). São disponibilizados dois arquivos: $detect-dataset.csv$ e $classData.csv$. O primeiro traz um DataSet para treinar e classificar a presença ou não de alguma falta no sistema. O segundo arquivo disponibilizado apresenta um conjunto de dados para treinar e classificar as faltas em: falta trifásica isolada, falta trifásica aterrada, falta bifásica isolada, falta bifásica aterrada, falta monofásica pra terra e sistema sem falta. O arquivo de dados $classData.csv$ foi o adotado neste trabalho

Os Datasets fornecem tensões e correntes trifásicas de um sistema de potência teórico que trabalha em 11KV de tensão. O sistema teórico é composto por 3 geradores, dois transformadores de potência, um reator de barra e uma LT conectando-os. A Figura 22 ilustra a rede proposta:

Figura 22 – Sistema de Potência em Estudo.



(PRAKASH, 2021)

No arquivo classData.csv, o sistema foi simulado em condições normais de operação e também em várias condições de falta. Das condições propostas foram coletadas e salvas as medidas de tensão e corrente trifásicas de linha como saída do sistema de potência. Adquiriu-se cerca de 7861 pontos e então o dataset foi classificado (tipos de falta e sem falta). A distribuição das amostras são de 2365 entradas para o caso Sem Falta; 1134

entradas para o caso Falta entre as Fases A,B e Terra; 1133 entradas para o caso Falta entre as Fases A, B e C e a Terra; 1129 para o caso Falta entre a Fase A e a Terra; 1096 entradas para o caso Falta entre as Fases A, B e C e 1004 entradas para o caso Falta entre a Fase B e Fase C. A seguinte saida é fornecida para classificar os tipos de falta:

Inputs - [Ia,Ib,Ic,Va,Vb,Vc] ; Outputs - [G C B A]

[G C B A]

[0 0 0 0] - Sem Falta

[1 0 0 1] - LG Falta (Falta entre a Fase A e a Terra)

[0 1 1 0] - LL Falta (Falta entre a Fase B e Fase C)

[1 0 1 1] - LLG Falta (Falta entre as Fases A,B e Terra)

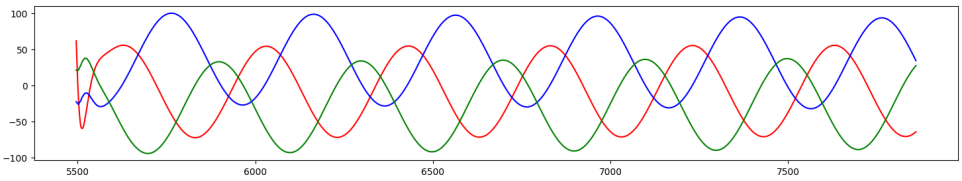
[0 1 1 1] - LLL Falta (Falta entre as Fases A, B e C)

[1 1 1 1] - LLLG Falta (Falta entre as Fases A, B e C e a Terra)

Para este trabalho, os dados contidos em classData.csv foram divididos em duas partes: 80% para treinamento e 20% para teste do modelo. O dataset de teste foi reservado para a validação da eficiência. Como 1ª etapa de pre-processamento foi verificada a existência ou não de valores nulos e os mesmos não estão presentes. Como 2ª etapa de pré-processamentos, uma padronização e uma normalização foram utilizadas para preparar o dataset completo em dois novos conjuntos de dados. No primeiro, foi realizada a padronização subtraindo de cada valor do dataset a média da sua coluna e dividindo o resultado pelo desvio padrão dessa coluna. Foi utilizada a ferramenta StdScaler no Python. Os dados estarão no intervalo entre 0 e 1. O segundo tratamento do dataset foi feito utilizando a ferramenta MaxAbsScaler. O MaxAbsScaler irá dividir cada coluna pelo módulo do seu maior valor, transformando os resultados dentro do intervalo de -1 a 1 (caso existam valores negativos). A ferramenta MaxAbsScaler do Python foi utilizada.

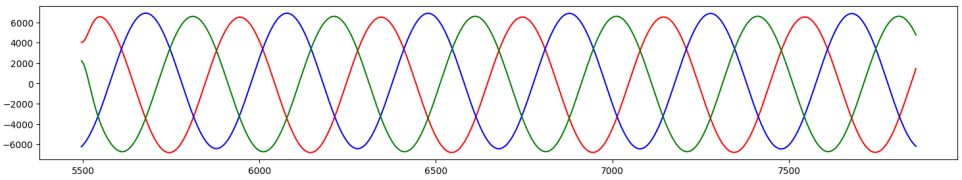
As Figuras 23 e 24 trazem as formas de onda do sinal de corrente e de tensão sem curto circuito do dataset.

Figura 23 – Corrente em Amperes - Sem Falta.



(PRAKASH, 2021)

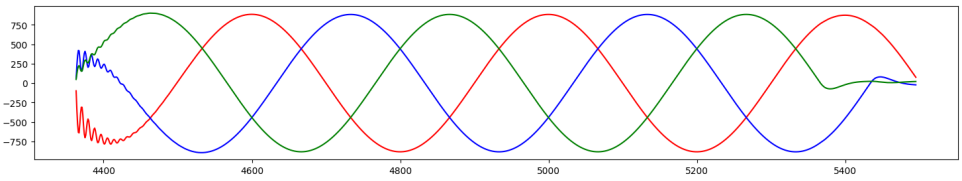
Figura 24 – Tensão em Volts - Sem Falta.



(PRAKASH, 2021)

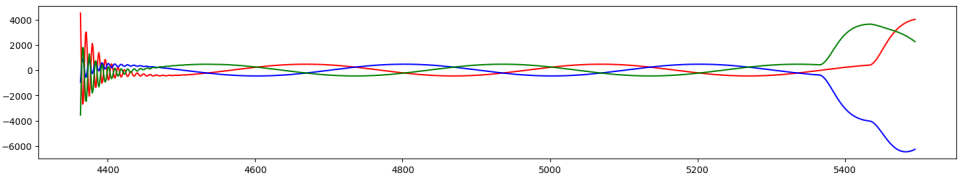
As Figuras 25 e 26 trazem as formas de onda do sinal de corrente e de tensão com um curto circuito trifásico aterrado do dataset.

Figura 25 – Corrente em Amperes - Falta 3F Aterrada.



(PRAKASH, 2021)

Figura 26 – Tensão em Volts - Falta 3F Aterrada.



(PRAKASH, 2021)

3.2 Definições e Métricas para Avaliação dos Modelos

Métricas: As métricas são utilizadas em AM para avaliar o desempenho. Existem métricas específicas para o problema multiclasse. Algumas delas estão descritas a seguir.

1. Acurácia (Accuracy): A acurácia é uma métrica comumente usada em problemas multiclasse. Ela mede a proporção de exemplos corretamente classificados (verdadeiros positivos e verdadeiros negativos) em relação ao número total de exemplos. A acurácia fornece uma visão geral do desempenho do modelo em todas as classes (CHEN *et al.*, 2023).

2. Precisão (Precision): A precisão é uma métrica que mede a proporção de verdadeiros positivos (exemplos corretamente classificados para uma classe específica) em relação à soma de verdadeiros positivos e falsos positivos (exemplos incorretamente classificados como pertencentes a essa classe). A precisão é útil quando o foco está na minimização de falsos positivos (CHEN *et al.*, 2023).

3. Revocação (Recall): A revocação é uma métrica que mede a proporção de verdadeiros positivos em relação à soma de verdadeiros positivos e falsos negativos (exemplos erroneamente classificados como não pertencentes a essa classe, mas que deveriam ser). A revocação é útil quando o objetivo é minimizar os falsos negativos (CHEN *et al.*, 2023).

4. AUC ROC: esta métrica apresenta o valor da área baixo da curva ROC. A curva ROC - Receiving Operating Characteristics - é um gráfico bi-dimensional com eixos X e Y que significam respectivamente a taxa de falsos positivos - TFP - e a taxa de verdadeiros positivos - TVP - (FACELI; LORENA; CARVALHO, 2011).

   - O valor AUC ficará entre 0 e 1, sendo que quanto mais próximo de 1 melhores são considerados os resultados.

3.2.1 Definições Complementares para Redes Neurais

Funções de Ativação: estas funções irão implementar não linearidade aos sinais ponderados na entrada do neurônio.

1. Função Sigmóide (Logística):

   A função sigmoide é uma função não linear que mapeia os valores de entrada em um intervalo entre 0 e 1. Esta função é amplamente usada em problemas de classificação binária, onde a saída desejada é uma probabilidade entre 0 e 1 (GOODFELLOW; BENGIO; COURVILLE, 2016). Sua definição é:

$$f(x) = \frac{1}{(1 + e^{-x})} \tag{3.1}$$

2. Função Tangente Hiperbólica (Tanh):

   A função tangente hiperbólica é outra função de ativação que mapeia os valores de entrada em um intervalo entre -1 e 1. Esta função é comumente usada em problemas de classificação binária ou multiclasse. De acordo com (WEBB; SAMMUT, 2017), sua definição é:

$$f(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \tag{3.2}$$

3. Função ReLU (Rectified Linear Unit):

   A função ReLU é uma função de ativação não linear que retorna zero para valores de entrada negativos e mantém os valores positivos inalterados (GOODFELLOW; BENGIO; COURVILLE, 2016). Sua definição é:

$$f(x) = max(0, x) \qquad f(x) = \begin{cases} 1 & se \quad x \geq 0 \\ 0 & se \quad caso \quad contrário \end{cases} \tag{3.3}$$

4. Função Leaky ReLU:

   A função Leaky ReLU é uma variação da função ReLU que permite um pequeno valor não nulo para entradas negativas, em vez de retornar zero. Esta função foi introduzida para resolver o problema de morte dos neurônios que ocorre quando a função ReLU transforma valores negativos em zero (MAAS; HANNUN; NG, 2013). Sua definição é:

$$f(x) = \begin{cases} \alpha * x & se \quad x < 0 \\ x & se \quad x \geq 0 \end{cases} \tag{3.4}$$

5. Função SoftMax:

   A função de ativação SoftMax é uma função não linear amplamente utilizada em redes neurais, especialmente em problemas de classificação multiclasse. Ela é projetada para mapear as saídas ponderadas dos neurônios em uma distribuição de probabilidade (GOODFELLOW; BENGIO; COURVILLE, 2016).

   A fórmula da função SoftMax para um vetor de entrada x de dimensão K é dada por:

$$softMax(x_i) = \frac{e^{x_i}}{\sum_1^k e^{x_k}}, \quad para \quad i = 1, 2, ..., K \tag{3.5}$$

   Onde $e^{(x_i)}$ representa a função exponencial de x_i e $\sum_1^k e^{x_k}$ é a soma exponencial de todos os elementos do vetor x.

Ao aplicar a função SoftMax, o valor resultante para cada elemento do vetor de entrada representa a probabilidade de que o elemento pertença a uma classe específica (GOODFELLOW; BENGIO; COURVILLE, 2016).

Função de Perdas(loss): Ao lidar com problemas de classificação multiclasse em redes neurais, é necessário escolher uma função de perda apropriada que meça a diferença entre as saídas da rede e as classes reais dos dados. A seguir são descritas algumas das funções de perda comumente usadas para problemas de classificação multiclasse em redes neurais com a biblioteca Keras. A escolha da função de perda depende do tipo de problema, da natureza dos dados e dos requisitos específicos do projeto.

1. Entropia Cruzada Categórica (Categorical Cross-Entropy): A entropia cruzada categórica é uma função de perda comumente usada para problemas de classificação multiclasse. Ela é adequada quando as classes são mutuamente exclusivas, ou seja, um exemplo de entrada pertence a apenas uma classe. A entropia cruzada categórica mede a divergência entre a distribuição de probabilidade prevista pela rede neural e a distribuição verdadeira das classes (CHOLLET, 2023a).

2. Entropia Cruzada Esparsa (Sparse Categorical Cross-Entropy): A entropia cruzada esparsa é uma variação da entropia cruzada categórica. É usada quando as classes são mutuamente exclusivas e representadas como rótulos inteiros (por exemplo, 0, 1, 2, etc.) em vez de vetores de codificação one-hot. A entropia cruzada esparsa também mede a divergência entre a distribuição prevista e a distribuição verdadeira das classes (CHOLLET, 2023a).

3. Erro Categórico (Categorical Hinge Loss): O erro categórico é uma função de perda alternativa para problemas de classificação multiclasse. É baseado na margem de classificação entre as classes e é adequado para problemas onde o objetivo é maximizar a margem entre as classes (CHOLLET, 2023b).

Otimizador(optimizer): o otimizador é usado para ajustar os pesos da rede com o objetivo de minimizar a função de perda e melhorar o desempenho do modelo.

1. Gradiente Descendente Estocástico (Stochastic Gradient Descent - SGD): O Gradiente Descendente Estocástico é um dos otimizadores mais básicos e amplamente utilizados em redes neurais. Ele atualiza os pesos da rede de acordo com o gradiente negativo da função de perda, multiplicado por uma taxa de aprendizado (CHOLLET, 2021).

2. RMSprop: O RMSprop é um otimizador que adapta a taxa de aprendizado com base nas médias móveis das magnitudes recentes dos gradientes. Este otimizador usa uma

média de decaimento exponencial para descartar os dados muito antigos. A partir deste descarte, o algoritmo poderá convergir rapidamente antes de encontrar um vale de mínimo local (CHOLLET, 2021).

3. Adaptive Moment Estimation (Adam): O Adam é um otimizador que combina os conceitos do RMSprop e do Moment Estimation. Ele mantém uma média móvel do gradiente e do momento do gradiente. O Adam adapta a taxa de aprendizado para cada parâmetro individualmente, com base nas estimativas de momento de primeira e segunda ordem (CHOLLET, 2021).

4. Adagrad: O Adagrad é um otimizador que adapta a taxa de aprendizado de forma individual para cada parâmetro, ajustando-o com base na soma dos gradientes passados. Ele dá maior importância aos parâmetros com gradientes menores (CHOLLET, 2023).

3.2.2 Definições Complementares para Máquina de Vetores de Suporte

Classificação de Margem Suave

A Classificação de Margem Suave tem como objetivo encontrar o melhor equilíbrio entre manter via o mais larga possível e reduzir as violações de margem(instâncias que acabam no meio da via ou classificadas erradamente). Este ajuste é feito variando-se o parâmetro C (GERON, 2019).

1. Kernel Polinomial

   A utilização de características polinomiais funciona bem em vários tipos de algoritmos de AM, nao somente com SVM. O modelo polinomial possui 3 hiperparâmetros a serem ajustados: o parâmetro C, o grau do polinômio e coef0.

   a) C: este hyperparâmetro já foi comentado anteriormente.

   b) Grau: Neste hyperparâmetro o que se observa é que um baixo grau do polinômio não lidará bem com conjuntos complexos. Em contrapartida, graus de polinômios muito altos criam muitas características, o que tornam o modelo muito lento (GERON, 2019).

   c) Coef: Este hyperparâmetro é o responsável por uma técnica matemática conhecida como truque de Kernel. Este truque permite se obter resultados como se fossem incluídas várias características polinomiais - mesmo com polinomios de alto grau - sem realmente incluí-las e assim evita-se a explosão combinatória resultante do aumento do grau do polinômio. O coef0 controla o quanto o modelo é influenciado por polinômios de alto grau versus polinômios de baixo grau (GERON, 2019). Este truque é implementado pela classe SVC que será utilizada nos testes.

2. Kernel RBF Gaussiano

   O kernel RBF Gaussiano utiliza o método de características de similaridade e pode ser utilizado em qualquer algoritmo de AM. Cabe ressaltar que o RBF Gaussiano pode gerar um grande custo computacional especialmente em grandes datasets. Assim como no Kernel Polinomial, pode-se utilizar o truque de kernel no RBF Gaussiano. Aqui o hiperparâmetro de trabalho será o **gamma** (Γ) e o **C**. Aumentar o **gamma** estreitará a curva de sino, reduzindo o raio de atuação da instância. Se reduzir o **gamma** as instâncias terão um maior raio de atuação, e o limite de decisão fica mais suave (GERON, 2019). Neste trabalho, o parâmetro **gamma** será utilizado no modo *"auto"*.

3.3 Aplicação de Redes Neurais Artificiais

Para detectar e classificar curtos-circuitos em Sistemas Elétricos de Potência, foi proposto a utilização de Redes Neurais Artificias. Para a implementação destas RNAs será utilizada a biblioteca Keras, que é um pacote Tensorflow Python, na modelagem da rede. Na conexão da rede neural utilizando a biblioteca Keras do TensorFlow as seguintes etapas são cumpridas:

a) **Importar as bibliotecas necessárias**: Comece importando as bibliotecas do TensorFlow e do Keras.

b) **Definir a arquitetura da rede neural**: Usar a classe Sequential do Keras para criar a sequência de camadas da rede neural. Adicionar as camadas desejadas, especificando o número de perceptrons/neurônios em cada camada e a função de ativação.

c) **Adicionar as camadas ocultas da rede neural**: a arquitetura da rede neural utilizando a ferramenta keras é conectada da seguinte forma:

   $model = keras.Sequential()$
   $model.add(keras.layers.Dense(units = 64, activation = ''))$
   $model.add(keras.layers.Dense(units = 32, activation = ''))$
   $model.add(keras.layers.Dense(units = 1, activation = ''))$

   Na estrutura acima, é definida uma rede neural com duas camadas densas (totalmente conectadas) e uma camada de saída com ativação sigmoidal.

d) **Definir a Função de Ativação**: As funções de ativação serão alternadas com as funções propostas anteriormente.

e) **Compilar o modelo**: Após definir a arquitetura da rede é necessário compilar o modelo. Nesta etapa devem ser definidas a função de perda, o otimizador e as métricas de avaliação. A seguir é apresentado um exemplo de código para compilação:

$model.compile(loss = '', optimizer = '', metrics = [''])$

f) **Treinar o modelo**: Nesta etapa o modelo é treinado com os dados de treinamento especificando o número de épocas e o tamanho do lote(batch size).

$model.fit(Xtrain, ytrain, epochs = 10, batchsize = 32)$

g) **Avaliar o modelo**: Após treinar o conjunto de dados, o modelo terá seu desempenho avaliado com os dados separados para teste/validação de acordo com a métrica selecionada.

$loss, accuracy = model.evaluate(Xtest, ytest)$

3.4 Aplicação de Máquina de Vetores de Suporte

Para detectar e classificar curtos-circuitos em Sistemas Elétricos de Potência também foi proposto a utilização de SVMs. Para a aplicação das SVMs foi utilizado o pacote Scikit-Learn em Python sendo utilizados dois kernels para Classificação Não Linear: Polinomial e RBF Gaussiano. Os hiperparâmetros configurados estão descritos na Seção 3.2. A seguir segue o modelo de como será implementado o kernel RBF Gaussiano:

$$krl = Pipeline([("SVM", SVC(kernel = "rbf", gamma = auto, C = 1))]) \quad (3.6)$$

$$kernel.fit(X, y) \quad (3.7)$$

4 APRESENTAÇÃO DE RESULTADOS

Foram realizados dois blocos de treinamentos para as RNAs e para as SVMs. Em um bloco foi implementada a padronização dos dados com *StandartScaler* e no outro bloco foi implementada a normalização dos dados com *MaxAbsScaler*.

As métricas testadas foram a Acurácia, Precisão, Revocação e valor AUC. Para avaliações em cima do SEP a acertividade da métrica a ser avaliada deve ser o objetivo devido ao elevado custo dos erros/falhas no sistema. Assim, o foco das análises esteve principalmente em cima da acurácia pois esta mede a proporção dos resultados corretamente classificados com o número total de exemplos.

4.1 RNA

As avaliações das RNAs foram divididas em duas etapas: Na primeira etapa, todos os modelos de RNA foram testados com as configurações propostas abaixo. Para a segunda etapa de avaliações, somente os modelos testados na primeira etapa com os valores de acurácia, precisão, revocação e auc acima de 86%, 86%, 70%, 70%, respectivamente, foram mais profundamente examinados com a utilização da ferramenta EarlyStopping.

As redes neurais treinadas e testadas seguiram as configurações descritas a seguir:

1. **Função de Ativação**: foram testadas as funções de ativação a seguir: softmax, tanh, ReLU e LeakyReLU.
2. **Otimizador**: foram testados os otimizadores rmsprop, adam, sgd e adagrad.
3. **Modelo da RNA**: a RNA testada possui a camada de entrada, uma camada oculta e a camada de saída. A camada de entrada possui 6 inputs e o número de neurônios é variado durante o treino [128, 256, 512]. O número de neurônios da camada oculta varia com as simulações [64, 128, 256, 512]. A camada de saída possui seis outputs e foi fixada a função de ativação sigmoid para a mesma.
4. **Função de Perdas**: foram testadas as funções de perdas categorical_crossentropy e categorical_hinge.
5. **EarlyStopping** - foi utilizada a ferramenta EarlyStopping para otimizar o número de épocas para cada Modelo de RNA evitando assim overtraining. O EarlyngStopping foi utilizado na segunda etapa das simulações e aplicados em modelos com acurácia superior a 84%.
6. **Camada de Saída**: foi utilizada a função SOFTMAX em todas as camadas de saída das redes.

4.1.1 RNA Standart Scaler

A seguir são apresentadas a Tabela 2 e a Figura 27. A Tabela 2 apresenta os 5 melhores resultados e o detalhamento do modelo testado.

Tabela 2 – 5 melhores resultados.

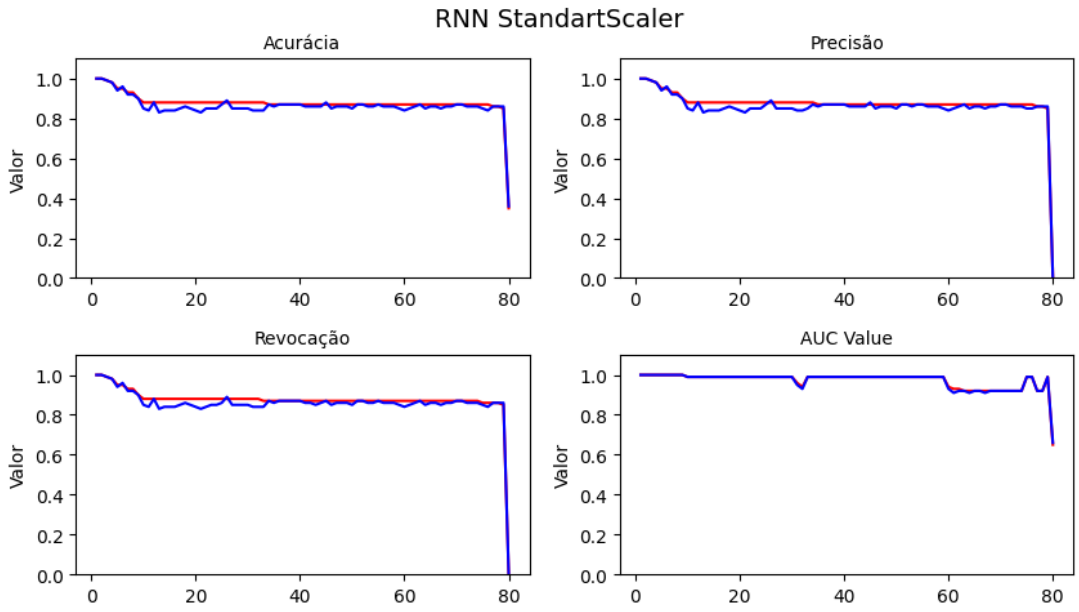
Epocas	FA	Otm	Loss	Cmd1	Cmd2	Loss_trn	Acc_trn	Revoc_trn	AUC_trn	Prec_trn	Loss_tst	Acc_tst	Revoc_tst	AUC_tst	Prec_tst
2836	LeakyReLU	adam	cc	512	128	0.01	1.0	1.0	1.0	1.0	0.02	1.0	1.0	1.0	1.0
3805	LeakyReLU	adam	cc	256	512	0.01	1.0	1.0	1.0	1.0	0.02	1.0	1.0	1.0	1.0
6192	softmax	adam	cc	256	256	0.03	0.99	0.99	1.0	0.99	0.05	0.99	0.99	1.0	0.99
8276	softmax	adam	cc	256	512	0.04	0.98	0.98	1.0	0.98	0.06	0.98	0.98	1.0	0.98
1701	LeakyReLU	adam	cc	512	512	0.12	0.95	0.95	1.0	0.95	0.13	0.94	0.94	1.0	0.94

Na tabela acima, as abreviações colocadas como títulos das colunas significam: FA - função de ativação; Otm - Otimizador; Loss - função de perdas; Cmd1 e 2 - camadas 1 e 2; _trn - indicação de grupo de treinamento; _tst - indicação de grupo de teste; acc - acurácia; Revoc - revocação; AUC - valor da área abaixo da curva ROC; Prec - precisão. As funções de perdas (Loss) na tabela estão também abreviadas como cc e ch e significam: categorical_crossentropy e categorical_hinge.

Os resultados da Tabela 2 para o dataset de testes trazem perdas bem pequenas para as melhores classificações e isto significa poucas diferenças entre as classes reais dos dados e as saídas da rede. Focando na acurácia dos resultados do topo da tabela, tem-se que a quantidade de classificações correntamente feitas é máxima, não havendo verdadeiros negativos nos resultados. As métricas precisão, revocação e valor AUC avaliam a proporção de falsos positivos, falsos negativos e o valor da área sob a curva ROC respectivamente e pelos valores exibidos conclui-se que os verdadeiros positivos prevalecem pois os valores das medidas são ou se aproximam do valor máximo 1. Os ótimos resultados nos testes validam que o treinamento da rede apresentou excelente generalização.

A Figura 27 apresenta 4 gráficos com o desempenho dos modelos testados para a acurácia, a precisão, a revocação e a auc. O eixo das absissas traz a quantidade de modelos testados na segunda etapa. No caso apresentado são aproximandamente 80 modelos testados e suas métricas.

Figura 27 – Métricas Standart Scaler.



Linha Vermelha: Treino e Linha Azul: Teste.

4.1.2 RNA MaxAbs Scaler

A seguir são apresentadas a Tabela 3 e a Figura 28. A Tabela 3 apresenta os 5 melhores resultados e o detalhamento do modelo testado.

Tabela 3 – 5 melhores resultados.

Epocas	FA	Otm	Loss	Cmd1	Cmd2	Loss_trn	Acc_trn	Revoc_trn	AUC_trn	Prec_trn	Loss_tst	Acc_tst	Revoc_tst	AUC_tst	Prec_tst
3460	LeakyReLU	adam	cc	512	512	0.02	1.0	1.0	1.0	1.0	0.03	1.0	1.0	1.0	1.0
3436	LeakyReLU	adam	cc	128	128	0.01	1.0	1.0	1.0	1.0	0.02	0.99	0.99	1.0	0.99
1969	tanh	adam	cc	256	128	0.03	1.0	1.0	1.0	1.0	0.05	0.99	0.99	1.0	0.99
3636	LeakyReLU	adam	cc	256	512	0.04	0.99	0.99	1.0	0.99	0.04	0.99	0.99	1.0	0.99
5624	softmax	adam	cc	512	256	0.1	0.97	0.97	1.0	0.97	0.11	0.97	0.97	1.0	0.97

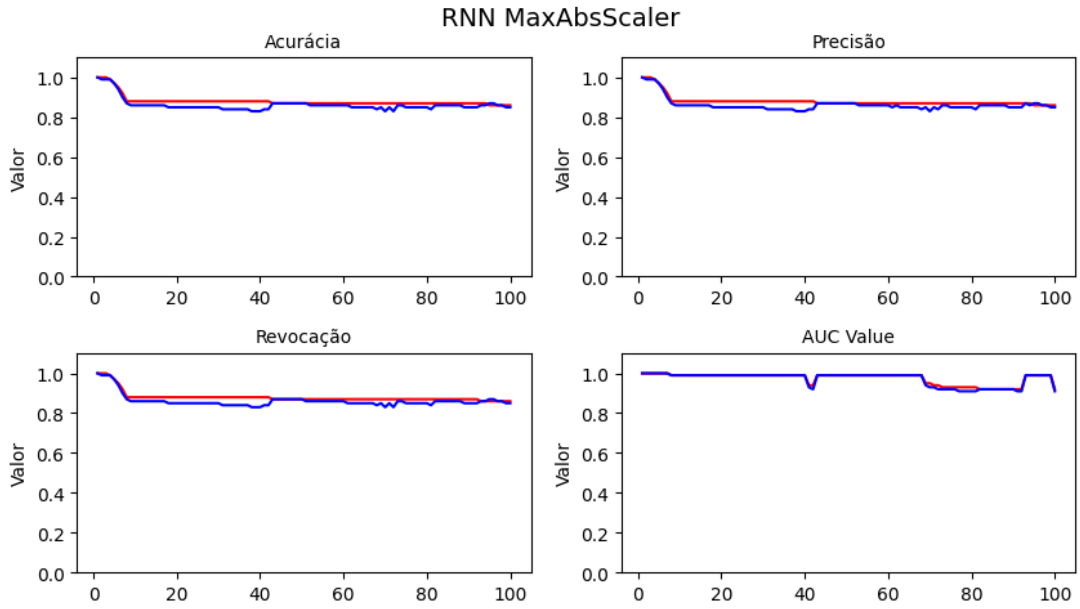
Na tabela acima, as abreviações colocadas como títulos das colunas significam: FA - função de ativação; Otm - Otimizador; Loss - função de perdas; Cmd1 e 2 - camadas 1 e 2; _trn - indicação de grupo de treinamento; _tst - indicação de grupo de teste; acc - acurácia; Revoc - revocação; AUC - valor da área abaixo da curva ROC; Prec - precisão. As funções de perdas (Loss) na tabela estão também abreviadas como cc e ch e significam: categorical_crossentropy e categorical_hinge.

Os resultados da Tabela 3 para o dataset de testes trazem perdas bem pequenas para as melhores classificações e isto significa poucas diferenças entre as classes reais dos dados e as saídas da rede. Focando na acurácia dos resultados do topo da tabela, tem-se que a quantidade de classificações correntamente feitas é máxima, não havendo verdadeiros negativos nos resultados. As métricas precisão, revocação e valor AUC mostram que os verdadeiros positivos prevalecem pois os valores das medidas são ou se aproximam do valor

máximo 1. Para este caso também são observados ótimos resultados nos testes e validam que o treinamento da rede apresentou excelente generalização.

A Figura 28 apresenta 4 gráficos com o desempenho dos modelos testados para a acurácia, a precisão, a revocação e a auc. O eixo das absissas traz a quantidade de modelos testados na segunda etapa. No caso apresentado são aproximandamente 100 modelos testados e suas métricas.

Figura 28 – Métricas com MaxAbs Scaler.



Linha Vermelha: Treino e Linha Azul: Teste.

4.1.3 Avaliação dos Resultados

Conforme proposto, o objetivo desta avaliação é focar nas configurações que retornaram as maiores proporções de classificações corretas. Esta linha de observação é destacada devido a sensibilidade financeira do tema abordado.

Avaliando-se as Tabelas 2 e 3, observa-se que os três primeiros modelos para ambos os casos obtiveram resultados acima de 99% de acurácia, sendo considerados resultados muito bons. Pela análise dos gráficos apresentados nas Figuras 27 e 28, percebe-se que os modelos testados apresentaram boa generalização do conjunto de treino para o conjunto de testes. Tal afirmação é feita uma vez que seus valores se mantiveram próximos.

4.2 SVM

Os modelos de SVMs além de serem testadas com Standart e MaxAbs Scaler, foram divididos em dois grupos: SVMs com kernel RBF(e gama auto) e SVMs polinomiais.

1. **C**: o parâmetro C foi variado nos testes de ambos os kernels e será apresentado junto com os resultados.
2. **Coef**: o parâmetro coef somente é utilizado para o kernel polinomial. Esse foi variado e será apresentado junto com os resultados.
3. **Grau**: foram testados polinômios com grau 2, 3, 4, 5 e 6.

As métricas avaliadas foram a Acurácia, Precisão, Revocação, valor AUC.

4.2.1 SVM - Kernel RBF com Standart Scaler

A seguir são apresentadas a Tabela 4 e a Figura 29. A Tabela 4 apresenta os 5 melhores resultados e o detalhamento do modelo testado.

Tabela 4 – 5 melhores resultados.

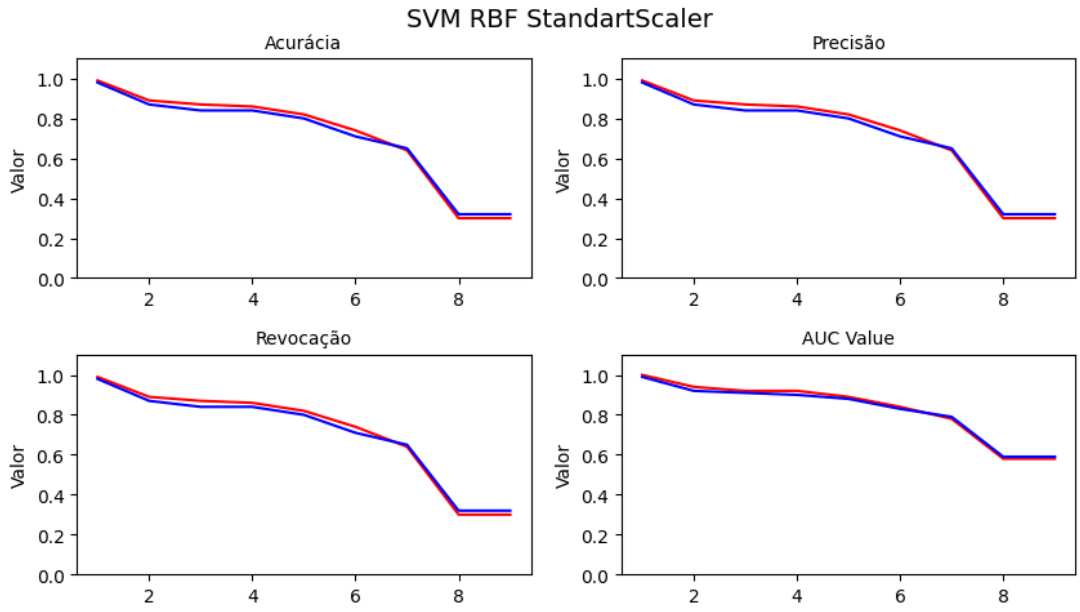
Nome	C	Acc_trn	Revoc_trn	AUC_trn	Prec_trn	Acc_tst	Revoc_tst	AUC_tst	Prec_tst	Scaler
Kernel rbf - gamma auto	10000.0	0.99	0.99	1.0	0.99	0.98	0.98	0.99	0.98	Std
Kernel rbf - gamma auto	1000.0	0.89	0.89	0.94	0.89	0.87	0.87	0.92	0.87	Std
Kernel rbf - gamma auto	100.0	0.87	0.87	0.92	0.87	0.84	0.84	0.91	0.84	Std
Kernel rbf - gamma auto	10.0	0.86	0.86	0.92	0.86	0.84	0.84	0.9	0.84	Std
Kernel rbf - gamma auto	1.0	0.82	0.82	0.89	0.82	0.8	0.8	0.88	0.8	Std

Na tabela acima, as abreviações colocadas como títulos das colunas significam: Coef - truque de Kernel; _trn - indicação de grupo de treinamento; _tst - indicação de grupo de teste; acc - acurácia; Revoc - revocação; AUC - valor da área abaixo da curva ROC; Prec - precisão.

Os resultados da Tabela 4 trazem os valores de acurácia encontrados para o dataset de treino e no topo da tabela observa-se que a quantidade de classificações correntamente feitas é próxima da máxima, retornando poucos verdadeiros negativos nos resultados. As métricas precisão, revocação e valor AUC mostram que os verdadeiros positivos prevalecem pois os valores das medidas se aproximam do valor máximo 1. Para este caso são observados ótimos resultados nos testes e validam que o treinamento da rede apresentou excelente generalização.

A Figura 29 apresenta 4 gráficos com o desempenho dos modelos testados para a acurácia, a precisão, a revocação e a auc. O eixo das absissas traz a quantidade de modelos testados na segunda etapa. No caso apresentado são aproximandamente 9 modelos testados e suas métricas. Foram testados mais modelos, mais os mesmos não conseguiram alcançar respostas e travaram as execuções.

Figura 29 – Métricas com Standart Scaler.



Linha Vermelha: Treino e Linha Azul: Teste.

4.2.2 SVM - Kernel RBF com MaxAbs Scaler

A seguir são apresentadas a Tabela 5 e a Figura 30. A Tabela 5 apresenta os 5 melhores resultados e o detalhamento do modelo testado.

Tabela 5 – 5 melhores resultados.

Nome	C	Acc_trn	Revoc_trn	AUC_trn	Prec_trn	Acc_tst	Revoc_tst	AUC_tst	Prec_tst	Scaler
Kernel rbf - gamma auto	10000.0	0.95	0.95	0.97	0.95	0.93	0.93	0.96	0.93	MAbs
Kernel rbf - gamma auto	1000.0	0.87	0.87	0.92	0.87	0.85	0.85	0.91	0.85	MAbs
Kernel rbf - gamma auto	100.0	0.86	0.86	0.92	0.86	0.84	0.84	0.9	0.84	MAbs
Kernel rbf - gamma auto	10.0	0.84	0.84	0.9	0.84	0.82	0.82	0.89	0.82	MAbs
Kernel rbf - gamma auto	1.0	0.74	0.74	0.84	0.74	0.71	0.71	0.83	0.71	MAbs

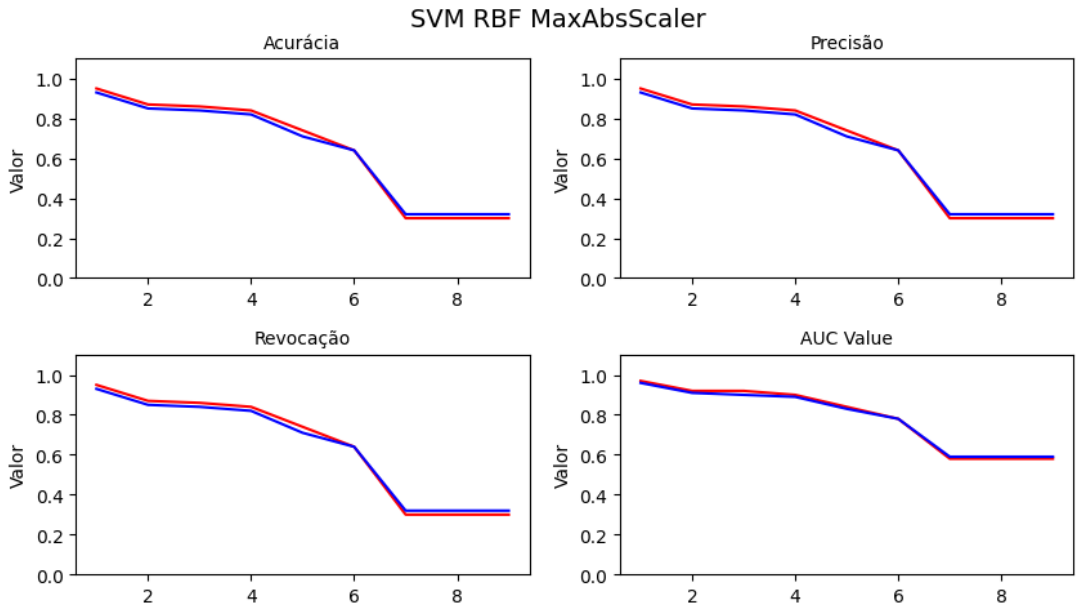
Na tabela acima, as abreviações colocadas como títulos das colunas significam: Coef - truque de Kernel; _trn - indicação de grupo de treinamento; _tst - indicação de grupo de teste; acc - acurácia; Revoc - revocação; AUC - valor da área abaixo da curva ROC; Prec - precisão.

Os resultados da Tabela 5 trazem os valores de acurácia encontrados para o dataset de treino e no topo da tabela observa-se que a quantidade de classificações correntamente feitas é próxima da máxima, retornando poucos verdadeiros negativos nos resultados. As métricas precisão, revocação e valor AUC mostram que os verdadeiros positivos prevalecem pois os valores das medidas se aproximam do valor máximo 1. Para este caso observa-se um ótimo resultado nos testes e este resultado valida que o treinamento da rede apresentou excelente generalização.

A Figura 30 apresenta 4 gráficos com o desempenho dos modelos testados para a

acurácia, a precisão, a revocação e a auc. O eixo das absissas traz a quantidade de modelos testados na segunda etapa. No caso apresentado são aproximandamente 9 modelos testados e suas métricas. Foram testados mais modelos, mais os mesmos não conseguiram alcançar respostas e travaram as execuções.

Figura 30 – Métricas com MaxAbs Scaler.



Linha Vermelha: Treino e Linha Azul: Teste.

4.2.3 SVM - Kernel Polinomial com Standart Scaler

A seguir são apresentadas a Tabela 6 e a Figura 31. A Tabela 6 apresenta os 5 melhores resultados e o detalhamento do modelo testado.

Tabela 6 – 5 melhores resultados.

Nome	Grau	Coef	C	Acc_trn	Revoc_trn	AUC_trn	Prec_trn	Acc_tst	Revoc_tst	AUC_tst	Prec_tst	Scaler
Kernel polinomial	4	4	10000	1.0	1.0	1.0	1.0	0.99	0.99	1.0	0.99	Std
Kernel polinomial	4	4	100	1.0	1.0	1.0	1.0	0.99	0.99	0.99	0.99	Std
Kernel polinomial	3	4	1000	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	Std
Kernel polinomial	6	1	100	0.99	0.99	0.99	0.99	0.98	0.98	0.99	0.98	Std
Kernel polinomial	3	3	1000	0.98	0.98	0.99	0.98	0.98	0.98	0.99	0.98	Std

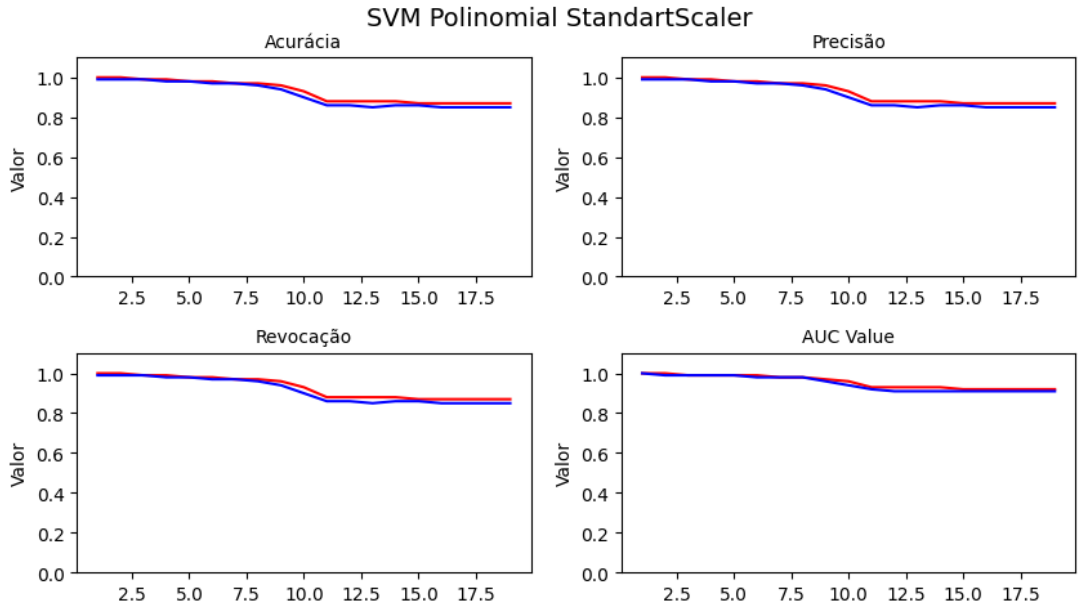
Na tabela acima, as abreviações colocadas como títulos das colunas significam: Coef - truque de Kernel; _trn - indicação de grupo de treinamento; _tst - indicação de grupo de teste; acc - acurácia; Revoc - revocação; AUC - valor da área abaixo da curva ROC; Prec - precisão.

Os resultados da Tabela 6 trazem os valores de acurácia encontrados para o dataset de treino e no topo da tabela observa-se que a quantidade de classificações correntamente feitas é próxima da máxima, retornando poucos verdadeiros negativos nos resultados. As métricas precisão, revocação e valor AUC mostram que os verdadeiros positivos prevalecem

pois os valores das medidas se aproximam ou possuem valor máximo 1. Para este caso observam-se ótimos resultados nos testes e estes resultados validam que o treinamento da rede apresentou excelente generalização.

A Figura 31 apresenta 4 gráficos com o desempenho dos modelos testados para a acurácia, a precisão, a revocação e a auc. O eixo das absissas traz a quantidade de modelos testados na segunda etapa. No caso apresentado são aproximandamente 18 modelos testados e suas métricas. Foram testados mais modelos, mais os mesmos não conseguiram alcançar respostas e travaram as execuções.

Figura 31 – Métricas com Standart Scaler.



Linha Vermelha: Treino e Linha Azul: Teste.

4.2.4 SVM - Kernel Polinomial com MaxAbs Scaler

A seguir são apresentadas a Tabela 7 e a Figura 32. A Tabela 7 apresenta os 5 melhores resultados e o detalhamento do modelo testado.

Tabela 7 – 5 melhores resultados.

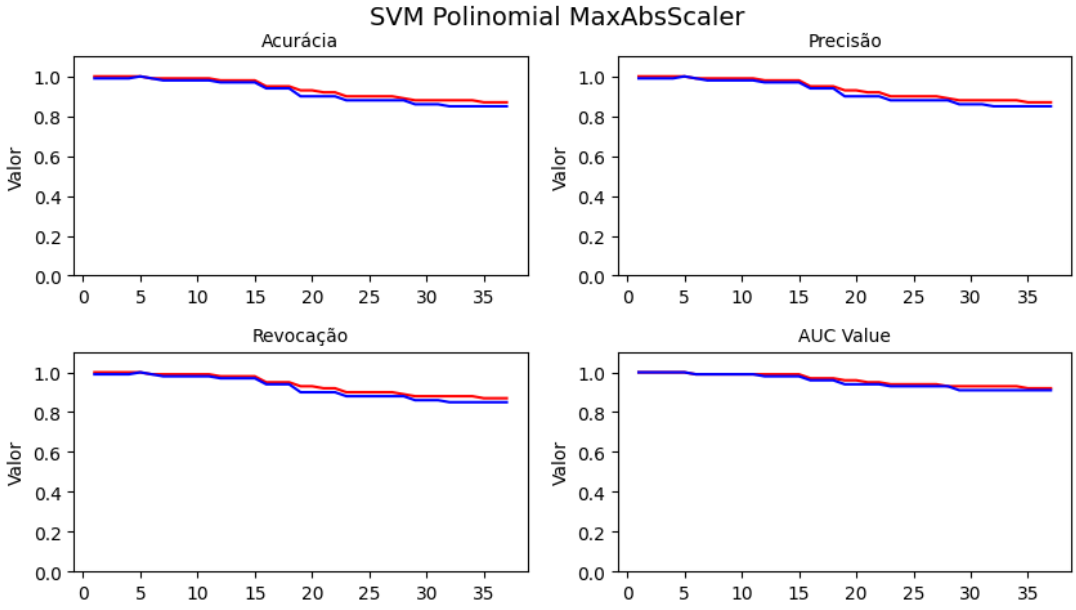
	Nome	Grau	Coef	C	Acc_trn	Revoc_trn	AUC_trn	Prec_trn	Acc_tst	Revoc_tst	AUC_tst	Prec_tst	Scaler
22	Kernel polinomial	4	4	1000	1.0	1.0	1.0	1.0	0.99	0.99	1.0	0.99	MAbs
32	Kernel polinomial	5	1	10000	1.0	1.0	1.0	1.0	0.99	0.99	1.0	0.99	MAbs
25	Kernel polinomial	4	3	1000	1.0	1.0	1.0	1.0	0.99	0.99	1.0	0.99	MAbs
28	Kernel polinomial	4	2	1000	1.0	1.0	1.0	1.0	0.99	0.99	1.0	0.99	MAbs
35	Kernel polinomial	6	1	1000	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	MAbs

Na tabela acima, as abreviações colocadas como títulos das colunas significam: Coef - truque de Kernel;_trn - indicação de grupo de treinamento; _tst - indicação de grupo de teste; acc - acurácia; Revoc - revocação; AUC - valor da área abaixo da curva ROC; Prec - precisão.

Os resultados da Tabela 7 trazem os valores de acurácia encontrados para o dataset de treino e no topo da tabela observa-se que a quantidade de classificações correntamente feitas é próxima da máxima, retornando poucos verdadeiros negativos nos resultados. As métricas precisão, revocação e valor AUC mostram que os verdadeiros positivos prevalecem pois os valores das medidas se aproximam ou possuem valor máximo 1. Para este caso observam-se ótimos resultados nos testes e estes resultados validam que o treinamento da rede apresentou excelente generalização.

A Figura 32 apresenta 4 gráficos com o desempenho dos modelos testados para a acurácia, a precisão, a revocação e a auc. O eixo das absissas traz a quantidade de modelos testados na segunda etapa. No caso apresentado são aproximandamente 35 modelos testados e suas métricas. Foram testados mais modelos, mais os mesmos não conseguiram alcançar respostas e travaram as execuções.

Figura 32 – Métricas com MaxAbs Scaler.



Linha Vermelha: Treino e Linha Azul: Teste.

4.2.5 Avaliação dos Resultados

Conforme proposto, o objetivo desta avaliação é focar nas configurações que retornaram as maiores proporções de classificações corretas.

Avaliando-se os resultados apresentados na Tabela 4 para o Kernel RBF, observa-se que a SVM com Standart Scaler não obteve um resultado satisfatório pois acurácia dos modelos não alcançaram valores de no mínimo 99%. Ainda para o Kernel RBF, da análise da Tabela 5 observa-se que o modelo da SVM com MaxAbs Scaler não obteve um resultado satisfatório pois nenhuma acurácia apresentou valores de no mínimo 99% .

Avaliando-se a Tabela 6 para o Kernel Polinomial, observa-se que a SVM com Standart Scaler obteve resultados iguais ou superiores a 99% para a acurácia. Ainda para o Kernel Polinomial, da análise da Tabela 7 observa-se que o modelo da SVM com MaxAbs Scaler também obteve resultados com valores iguais ou superiores a 99% para a acurácia.

Pela análise dos gráficos apresentados nas Figuras 29, 30, 31 e 32, percebe-se que os modelos testados apresentaram boa generalização do conjunto de treino para o conjunto de testes. Tal afirmação é feita uma vez que seus valores se mantiveram próximos.

5 CONCLUSÃO

As formas de onda apresentadas nas Figuras 23 e 24 apresentam graficamente o dataset para o sistema em operação normal e as Figuras 25 e 26 apresentam um dos cinco casos estudados de falha no sistema - o curto circuito trifásico. Essas representações gráficas evidenciam como o comportamento das tensões e correntes são diferentes em regime normal e em contigência. A partir dessa diferença, o trabalho se propôs na questão problema a avaliar a identificação e classificação de curtos circuitos por meio de Redes Neurais Artificiais e Máquina de Vetores de Suporte.

A detecção e classificação de curtos-circuitos com a utilização de RNA foi marcada por uma grande quantidade de simulações que vão de uma primeira etapa de testes para a seleção dos modelos com resultados de acurácia, precisão, revocação e valor auc acima de 86%, 86%, 70% e 70% respectivamante e uma segunda etapa onde se realizou os testes dos modelos com a utilização da ferramenta EarlyStopping com o objetivo de gerenciar o número de épocas e evitar overfitting. Todo este processo foi realizado duas vezes: na primeira vez o dataset foi tratado com a padronização Standart Scaler e na segunda vez o dataset foi tratado com a normalização MaxAbs Scaler.

Foram organizados no Capítulo 4 os resultados das simulações com as várias configurações propostas e a Seção 4.1 apresenta os resultados obtidos para as Redes Neurais Artificiais. As RNAs testadas com o dataset tratado conforme proposto no parágrafo anterior retornaram resultados excepcionais de acurácia para ambos os casos, sendo que os melhores apresentaram-se entre 99% e 100% no dataset de teste. As métricas de precisão, revocação e valor auc avaliadas também apresentaram resultados excepcionais entre 99% e 100% no dataset de teste.

Já a detecção e classificação de curtos-circuitos obtida com as SVMs foram marcadas pela grande quantidade de modelos que não obtinham respostas. Essa não obtenção de resposta ocasionou redução do número de modelos a serem apresentados. Aqui também os dados foram tratados com a padronização Standart Scaler e com a normalização MaxAbs Scaler, produzindo assim dois conjuntos de testes separados. Foram escolhidos os kernel Polinomial e RGB para a realização das simulações.

Na Seção 4.2 foram organizados e apresentadas as respostas encontradas para acurácia, precisão, revocação e valor auc para a Máquina de Vetores de Suporte. Observa-se que o kernel RBF com Standart Scaler apresentou um resultado excepcional de 98% para a acurácia no dataset de testes. Valores excepcionais para precisão, revocação e valor auc acima de 98% foram também observados para o dataset de testes. O kernel RBF com MaxAbs Scaler apresentou resultados bons no dataset de testes, mas abaixo do esperado

para uma aplicação sensível como a classificação de curto-circuito em Sistemas Elétricos de Potência.

Para o kernel Polinomial com Standart Scaler o modelo retornou alguns resultados excepcionais de acurácia acima de 99% de acertos nas classificações realizadas no dataset de testes. Os valores de precisão, revocação e valor auc também se mostraram excepcionias e acima de 99%. Para o kernel Polinomial com MaxAbs Scaler os resultados obtidos para acurácia foram também excepcionias com acurácia acima de 99% de acertos nas classificações realizadas no dataset de testes. Os valores de precisão, revocação e valor auc também se mostraram excepcionias e acima de 99%.

Por meio desses resultados, é possível concluir que os testes realizados com Redes Neurais Artificiais e Máquina de Vetores de Suporte apresentam excelentes resultados de acurácia, precisão, revocação e valor auc na tarefa de detecção e classificação de curto-circuito.

Os modelos simulados e com os melhores valores de acurácia podem ser utilizados no processo de Operação em Tempo Real de Sistemas Elétricos de Potência auxiliando os Operadores de Sistema em situações de contingência que haja necessidade de identificar que tipo de curto-circuito ocorreu. Esse tipo de informação traz segurança ao operador no momento de retornar o equipamento a rede e também no momento em que se realiza comunicação com o Operador Nacional do Sistema - ONS. As análises com as Redes Neurais Artificiais e Máquina de Vetores de Suporte estudadas podem também compor itens específicos dos Relatórios de Ocorrência relacionados as contingências no SEP. Esses relatórios, que devem ser apresentados ao ONS, constantemente exigem grande quantidade de homem-hora especilizada para a sua produção.

5.1 Trabalhos Futuros

A partir da pequena experiência obtida com o desenvolvimento deste trabalho, várias propostas são visualidades como possíveis trabalhos futuros.

1. Identificação de problemas em Transformadores de Potência a partir da avaliação das características das Tensões e Correntes nos equipamento.

2. Identificação de problemas em Reatores de Potência a partir da avaliação o das características das Tensões e Correntes nos equipamentos.

3. Identificação de problemas em Reatores a Núcleo de Ar em Sistema HVDC a partir da avaliação das características das Tensões e Correntes nos equipamentos.

4. Identificação de problemas em Capacitores Shunt e Compensações Série a partir da avaliação das características das Tensões e Correntes nos equipamentos.

5. Identificação X/R para Regiões do Sistema de Potência.

6. Identificação de possíveis problemas com Reguladores de Tensão em Geradores de Energia Elétrica.

REFERÊNCIAS

ALPAYDIN, E. **Introduction to Machine Learning.** 4. ed. [*S.l.: s.n.*]: The MIT Press, 2020.

ANGLUIN, D.; SMITH, C. H. Inductive inference: Theory and methods. **Comput. Surveys**, v. 15, n. 03, 1983.

BORGES, C. L. T. **Análise de Sistemas de Potência**. [*S.l.: s.n.*]: UFRJ, 2005.

BOSER, B.; I.L., G.; VAPNIK, V. **A training algorithm for optimal margin classifiers.** [*S.l.: s.n.*]: Proceedings of the 5th Annual Woekshop on Computational Learning Theory., 1992.

CHEN, D. *et al.* **Métricas de Avaliação em Machine Learning; Classificação**. 2023. Disponível em: https://medium.com/kunumi/metricas-de-avaliacao-em-machine-learning-classificacao-49340dcdb198. Acesso em: 17 jun. 2023.

CHOLLET, F. **Deep Learning with Python**. 2. ed. [*S.l.: s.n.*]: Manning, 2021. ISBN 9781617296864.

CHOLLET, F. **ADAGRAD Documentation**. 2023. Disponível em: https://keras.io/api/optimizers/adagrad/. Acesso em: 17 jun. 2023.

CHOLLET, F. **Probabilistic losses**. 2023. Disponível em: https://keras.io/api/losses/probabilistic_losses/. Acesso em: 17 jun. 2023.

CHOLLET, F. **Probabilistic losses**. 2023. Disponível em: 'https://keras.io/api/losses/hinge_losses/'. Acesso em: 17 jun. 2023.

CORTES, C.; VAPNIK, V. **Support Vector Machine.** [*S.l.: s.n.*]: Machine Learning: 273-296, 1995.

CRISTIANINI, N.; SHAWE-TAYLOR, J. **An introduction to Support Vector Machines and Other-Kernel-based Learning Methods**. [*S.l.: s.n.*]: Cambridge University Press, 2000.

D'AJUZ, A. *et al.* **Equipamentos Elétricos; especificação e aplicação em subestações de alta tensão.** [*S.l.: s.n.*]: Universidade Federal Fluminense and FURNAS SA, 1985.

DAS, J. **Transients in Electrical Systems - Analysis, Recognition, And Mitigation**. [*S.l.: s.n.*]: McGRaw Hill, 2010.

FACELI, K.; LORENA, A.; CARVALHO, A. **Inteligência Artificial – Uma abordagem de Aprendizado de Máquina.** [*S.l.: s.n.*]: LTC, 2011.

FACURE, M. **Funções de Ativação - Entendendo a importância da ativação correta nas redes neurais.** 2012. Disponível em: https://matheusfacure.github.io/2017/07/12/activ-func/. Acesso em: 17 jun. 2023.

FRANCO, C. **Inteligência Artificial.** [*S.l.: s.n.*]: Editora e Distribuidora Educacional S.A., 2014.

GERON, A. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn TensorFlow: Conceitos, Ferramentas e Técnicas para a Construção de Sistemas Inteligentes.** [*S.l.: s.n.*]: Alta Books, 2019. ISBN 978-85-508-0902-1.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [*S.l.: s.n.*]: MIT Press, 2016. Http://www.deeplearningbook.org. Acesso em: 03 set. 2023.

HASSAN, J. U.; NIZAMI, I. F. Machine learning algorithm analysis for detecting and classification faults in power transmission system. *In*: **2022 2nd International Conference on Digital Futures and Transformative Technologies (ICoDT2)**. [*S.l.: s.n.*]: IEEE, 2022. p. 1–5.

HAYKIN, S. **Neural Networks: A Comprehensive Foundation**. [*S.l.: s.n.*]: Prentice-Hall, 1999.

HERBRICH, R. **Learning Kernels Classifiers: Theory and Algorithms**. [*S.l.: s.n.*]: MIT Press, 2001.

IRWIN, J. D.; NELMS, R. M. **Basic engineering circuit analysis**. [*S.l.: s.n.*]: John Willey and Sons, Inc, 2007.

KINDERMANN, G. **Curto Circuito.** [*S.l.: s.n.*]: SAGRA-LUZZATTO, 1997.

MAAS, A. L.; HANNUN, A. Y.; NG, A. Y. Rectifier nonlinearities improve neural network acoustic models. **Workshop on Deep Learning for Audio, Speech, and Language Processing**, 2013. In ICML Workshop on Deep Learning for Audio, Speech, and Language Processing.

MERCER, J. **Functions of positive and negative type and their connection with the theory of integral equations.** [*S.l.: s.n.*]: Philosophical Transactions of the Royal Society, 1909.

MUNDRA, M.; SHAH, S.; CHATTERJEE, S. **Detection of Symmetrical/Asymmetrical Faults with DULR and XGBoost.** [*S.l.: s.n.*]: IEEE, 2022.

ONS. **Mapa do Sistema de Transmissao - Horizonte 2027**. 2023. Disponível em: https://www.ons.org.br/paginas/sobre-o-sin/mapas. Acesso em: 10 set. 2023.

ONS. **Matriz de Energia Elétrica**. 2023. Disponível em: https://www.ons.org.br/paginas/sobre-o-sin/o-sistema-em-numeros. Acesso em: 10 set. 2023.

PRAKASH, E. S. **Electrical Fault detection and classification - A collection of line currents and voltages for different fault conditions**. 2021. Disponível em: https://www.kaggle.com/datasets/esathyaprakash/electrical-fault-detection-and-classification. Acesso em: 10 jun. 2023.

REIS, L. B. d. **Geração de Energia Elétrica: tecnologia, inserção ambiental, planejamento, operação e análise de viabilidade.** [*S.l.: s.n.*]: Manole, 2003. ISBN 85-204-1536-9.

RUMELHART, D.; MCCLELLAND, J. **Parallel Distributed Processing.** [*S.l.: s.n.*]: The MIT Press, 1986.

SMOLA, A.; SCHOLKOPF, B. **Learning with Kernels**. [*S.l.: s.n.*]: The MIT Press, 2002.

STEVENSON, W. D. **Elementos da Análise de Sistemas de Potência.** [*S.l.: s.n.*]: McGraw-Hill, 1986.

UDDIN, S. *et al.* **An Intelligent Short-Circuit Fault Classification Scheme for Power Transmission Line.** [*S.l.: s.n.*]: IEEE, 2021.

VAPNIK, V. **The Nature of Statistical Learning Theory**. [*S.l.: s.n.*]: Springer-Verlag, New York, 1995.

VAPNIK, V.; CHERVONENKIS, A. **On the uniform convergence of relative frequencies of events to their probabilities.** [*S.l.: s.n.*]: Theory of Probability and its Applications, 1971.

WEBB, G. I.; SAMMUT, C. **Encyclopedia of Machine Learning and Data Mining**. [*S.l.: s.n.*], 2017. Disponível em: http://hdl.handle.net/2078/ebook:90011. Acesso em: 26 set. 2023.