UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

LAURA FERNANDA DE BARROS NOGUEIRA

AVALIAÇÃO DE FERRAMENTAS DE *MACHINE LEARNING* PARA CLASSIFICAÇÃO DE FALTAS EM LINHAS DE TRANSMISSÃO

Recife
2024

LAURA FERNANDA DE BARROS NOGUEIRA

AVALIAÇÃO DE FERRAMENTAS DE *MACHINE LEARNING* PARA CLASSIFICAÇÃO DE FALTAS EM LINHAS DE TRANSMISSÃO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Orientador(a): Prof. Dr. Douglas Contente Pimentel Barbosa

Recife
2024

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Nogueira, Laura Fernanda De Barros .
Avaliação de Ferramentas de Machine Learning para Classificação de Faltas em Linhas de Transmissão / Laura Fernanda De Barros Nogueira. - Recife, 2024.
77 p. : il., tab.

Orientador(a): Douglas Contente Pimentel Barbosa
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia de Controle e Automação - Bacharelado, 2024.
Inclui referências, apêndices.

1. Machine Learning. 2. Classificação de Faltas. 3. Linhas de Transmissão. 4. Sistemas Elétricos. 5. Pós-Falta. I. Barbosa, Douglas Contente Pimentel. (Orientação). II. Título.

620 CDD (22.ed.)

LAURA FERNANDA DE BARROS NOGUEIRA

AVALIAÇÃO DE FERRAMENTAS DE *MACHINE LEARNING* PARA CLASSIFICAÇÃO DE FALTAS EM LINHAS DE TRANSMISSÃO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Aprovado em: 06/08/2024.

BANCA EXAMINADORA

_____

Prof. Dr. Douglas Contente Pimentel Barbosa (Orientador)
Universidade Federal de Pernambuco

_____

Prof. M.Sc. Alex Ferreira Falcão Moreira (Examinador Interno)
Universidade Federal de Pernambuco

_____

Eng. M.Sc. Pablo Luiz Tabosa da Silva (Examinador Externo)
Universidade Federal de Pernambuco

AGRADECIMENTOS

Primeiramente, expresso minha imensa gratidão a Deus, por me conceder saúde, foco e resiliência ao longo da minha jornada acadêmica. Sou profundamente grata aos meus pais, Maria de Fátima e Adalberto, cujo apoio emocional, orientação e amor incondicional foram fundamentais para as minhas conquistas. Estendo meus agradecimentos à Interest Engenharia, especialmente ao time de SPCS, cujo apoio foi essencial para o desenvolvimento do meu projeto. Aos meus amigos, obrigada por todas as palavras de incentivo e pelos momentos relaxantes que ajudaram a equilibrar a intensidade dos estudos. Um agradecimento especial ao meu orientador, Prof. Dr. Douglas Contente, cuja sabedoria, paciência e orientação foram decisivas para a conclusão e sucesso deste trabalho. A todos vocês, meu respeito e gratidão por cada momento compartilhado e cada lição aprendida nesta jornada.

“Não devemos ter medo das novas ideias! Elas podem significar a diferença entre o triunfo e o fracasso.”
Napoleon Hill

RESUMO

Este estudo propõe a aplicação de técnicas de *Machine Learning* para a classificação de faltas em linhas de transmissão no cenário pós-falta. O objetivo principal é desenvolver um sistema eficaz capaz de classificar diferentes tipos de faltas na rede elétrica de transmissão após a ocorrência de uma falha. Utilizando dados simulados, são comparados diversos modelos de aprendizado de máquina, como Regressão Logística, kNN, *Random Forest*, SVM e Rede Neural Artificial, para determinar qual apresenta maior precisão na classificação dos tipos de faltas. Este trabalho visa contribuir para a eficiência operacional e a confiabilidade do sistema elétrico, inserindo-se no âmbito da automação e monitoramento inteligente do setor elétrico brasileiro.

Palavras-chave: *Machine Learning*, Classificação de Faltas, Linhas de Transmissão, Sistemas Elétricos, Pós-Falta.

ABSTRACT

This study proposes the application of Machine Learning techniques for fault classification in transmission lines in the post-fault scenario. The main objective is to develop an effective system capable of classifying different types of faults in the electrical transmission network after a fault has occurred. Using simulated data, various machine learning models, such as Logistic Regression, kNN, Random Forest, SVM, and Artificial Neural Networks, are compared to determine which model achieves the highest accuracy in classifying fault types. This work aims to contribute to the operational efficiency and reliability of the electrical system, aligning with the automation and intelligent monitoring of the Brazilian electrical sector.

Keywords: Machine Learning, Fault Classification, Transmission Lines, Electrical Systems, Post-Fault.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ATPDraw - Alternative Transients Program Draw

CSV - Comma-Separated Values

GRU - Gated Recurrent Unit

IEEE - Institute of Electrical and Electronics Engineers

kNN - K-Nearest Neighbor

LSTM - Long Short-Term Memory

RF – Random Forest

RNA - Rede Neural Artificial

SCADA - Supervisory Control and Data Acquisition

SMOTE - Synthetic Minority Over-sampling Technique

SVM - Support Vector Machine

THD - Total Harmonic Distortion

SUMÁRIO

1 INTRODUÇÃO

Este estudo propõe a aplicação de técnicas de *Machine Learning* (Aprendizado de Máquina) para a classificação de faltas em linhas de transmissão. Dada a complexidade desses eventos, é fundamental adotar abordagens inovadoras. Os algoritmos de aprendizado de máquina se destacam por sua capacidade de aprender padrões complexos, com base em análises por repetição. Uma base de dados simulados no software ATPDraw, que possui diversas faltas em linhas de transmissão, será utilizada para treinar e validar o melhor modelo. Serão comparados cinco modelos de aprendizado de máquina, a fim de encontrar qual tem o maior percentual de acerto na tarefa de classificar os tipos de faltas, especificamente as de curto-circuito, em linhas de transmissão. O intuito é contribuir para a eficiência operacional e a confiabilidade do sistema elétrico, inserindo-se no âmbito da automação e monitoramento inteligente do setor elétrico brasileiro.

1.1 Objetivos

O objetivo deste trabalho é desenvolver um sistema de classificação de faltas em linhas de transmissão usando técnicas de *Machine Learning*.

1.1.1 Geral

O objetivo geral deste trabalho é comparar modelos de *Machine Learning* para desenvolver um sistema de classificação de faltas em linhas de transmissão, visando determinar o mais adequado para a realização de análises de faltas no sistema de transmissão de energia elétrica.

1.1.2 Organização do Trabalho

A estrutura deste trabalho seguirá uma abordagem que começa por uma revisão bibliográfica que contextualiza os fundamentos teóricos e pesquisas relacionadas à

classificação de faltas, com ênfase no papel do aprendizado de máquina aplicado ao problema. Em seguida, no capítulo 3, serão apresentados a metodologia empregada para realização do estudo, abordando a escolha da base de dados e como será feita a análise utilizando ferramentas de *Machine Learning*. O desenvolvimento e otimização do algoritmo empregado nesse estudo serão apresentados no capítulo 4, mostrando a implementação prática do modelo por meio de linguagem de programação de alto nível. As análises e resultados obtidos serão discutidos no capítulo 5, quanto ao desempenho do sistema. No capítulo 6, por fim, serão apresentadas as conclusões do estudo, destacando contribuições, limitações e sugestões para futuras pesquisas na área de monitoramento de faltas em linhas de transmissão.

2 FUNDAMENTAÇÃO TEÓRICA

A utilização de algoritmos de *Machine Learning* para classificar faltas no sistema elétrico de transmissão tem como intuito aprimorar a confiabilidade e a eficiência dos sistemas de alta potência, a fim de garantir uma resposta mais rápida e eficaz a eventuais problemas em linhas de transmissão. Por meio da análise de padrões complexos de dados simulados de falhas em linhas de transmissão, algoritmos de aprendizado de máquina podem identificar diferentes tipos de faltas reais, possibilitando intervenções que minimizem o tempo de inatividade e melhorem a segurança operacional. Além disso, a automação do processo de classificação proporcionada pelo processo utilizando aprendizado de máquina reduz a necessidade de intervenção humana, tornando a operação do sistema mais eficiente e econômica. Para tal análise faz-se necessário o entendimento do funcionamento de algoritmos de aprendizado de máquinas e alguns de seus modelos de classificação, entendendo como pode ser aplicado para o caso de faltas em linhas de transmissão.

2.1 Machine Learning

O aprendizado de máquina é um ramo da inteligência artificial que se baseia na ideia de fornecer um conjunto de dados para uma máquina e permitir que ela aprenda de forma autônoma a executar uma determinada tarefa. À medida que a máquina ganha mais experiência, ela se torna cada vez mais habilidosa. Formalmente, um programa de computador pode ser considerado como aprendendo a partir de uma experiência (E) em relação a uma classe de tarefas (T) e uma medida de desempenho (P), se seu desempenho medido por (P) em tarefas (T) melhora com a experiência (E) [1]. A Figura 1 oferece uma visão geral das categorias de aprendizado de máquina, que se dividem em três tipos principais: supervisionado, não supervisionado e por reforço. Cada um desses conceitos será detalhado nos tópicos seguintes.

Figura 1 – Tipos de Aprendizado de Máquina

Fonte: (A autora, 2024)

2.1.1 Aprendizado Não Supervisionado

O aprendizado não supervisionado é um método de Machine Learning. Ele opera com dados que não têm etiquetas ou classificações anotadas. Nesse método os modelos são utilizados para identificar padrões em dados que não tem supervisão explicita, os algoritmos são treinados em conjuntos de dados não rotulados, ou seja, onde os dados não têm saídas desejadas fornecidas. O objetivo do aprendizado não supervisionado é encontrar estrutura nos dados, como agrupamentos naturais ou padrões subjacentes, e assim, encontrar e classificar grupos a partir da análise feita, sem supervisão ou designação de classes. Alguns dos métodos mais comuns incluem clustering (agrupamento), sumarização e associação. [2]

2.1.2 Aprendizado Supervisionado

O aprendizado supervisionado é a abordagem em *Machine Learning* mais utilizada, pois ela envolve algoritmos feitos para treinar conjuntos de dados tendo tanto a saída quanto a entrada sendo conhecidas. A partir dos dados de entrada deseja-se que o modelo aprenda a mapear as saídas de uma forma que seja generalizada para todos os novos dados. Para o aprendizado supervisionado uma entrada sempre tem uma saída correspondente, ou seja, um rótulo. Nesse método de aprendizado são

utilizados os pares entrada-saída como uma forma de mapear os dados durante o algoritmo. [2]

Tanto para o aprendizado supervisionado quando para o aprendizado não supervisionado é utilizado um conjunto para treino e um para teste. O conjunto de treino geralmente é maior e é usado para que o algoritmo seja capaz de conseguir uma maior generalização nas aplicações, já para no conjunto de teste ele checa a porcentagem de acerto que o algoritmo conseguiu generalizar, baseado agora em um novo conjunto de dados diferente dos usados para treino. [3]

Esse método é comumente utilizado em diversas aplicações para classificação e regressão. Na tarefa de classificação ela tem como objetivo prever uma categoria ou classe para uma entrada, já na tarefa de regressão, ela pretende prever um valor contínuo. Esse tipo de aprendizado também enfrenta alguns desafios, sendo os principais dentre eles *o overfitting* e o *underfitting*.

2.1.2.1 Overfitting

Overfitting é um fenômeno comum em modelos de aprendizado de máquina, ele ocorre quando o modelo se ajusta muito bem aos dados de treinamento, mas não consegue generalizar adequadamente para novos dados. Isso acontece quando o modelo se torna excessivamente complexo, capturando não apenas os padrões relevantes nos dados, mas também o ruído ou detalhes irrelevantes dos dados de treinamento. No entanto, o modelo pode apresentar um desempenho muito satisfatório nos dados de treinamento, mas um desempenho inferior nos dados de teste ou em situações reais. Modelos super ajustados geralmente tem alta variância, ou seja, adaptam-se demais aos dados de treinamento e são sensíveis a pequenas variações nos dados. Isso pode levar a previsões imprecisas ou inúteis em novos exemplos. [4]

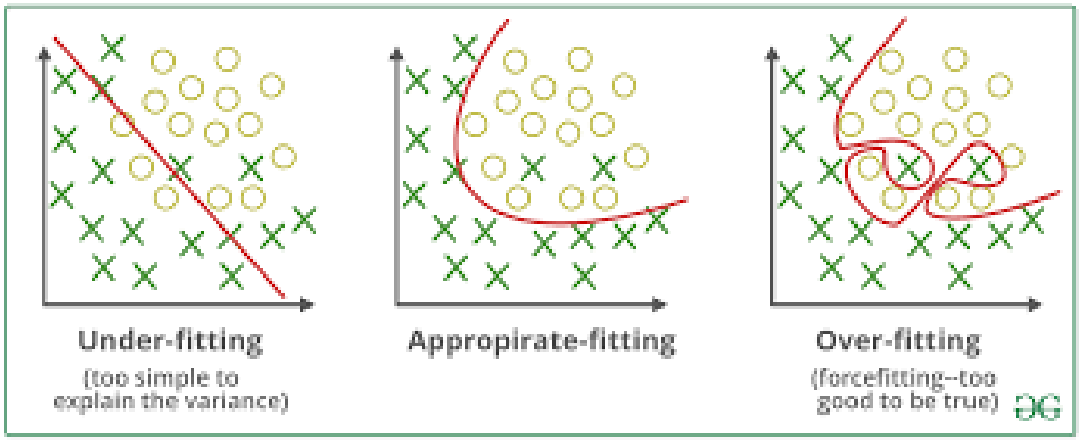
2.1.2.2 Underfitting

Underfitting acontece quando o modelo de aprendizado de máquina é muito simples para entender os dados que está tentando aprender. Isso resulta em um

modelo que não consegue capturar os padrões importantes e como resultados, tem um desempenho ruim tanto nos dados de treinamento quanto nos dados de teste. O *underfitting* ocorre quando um modelo não é capaz de fazer previsões precisas com base nos dados de treinamento e, portanto, não tem a capacidade de generalizar bem em novos dados, tornando-se difícil aprender tendências. Modelos de aprendizado de máquina com *underfitting* tendem a ter desempenho ruim tanto nos conjuntos de treinamento quanto nos conjuntos de teste [4].

A Figura 2 ilustra três cenários de ajuste de modelos de *Machine Learning*: *underfitting*, ajuste apropriado e *overfitting*. No *underfitting* (gráfico à esquerda), o modelo é muito simples para capturar a variabilidade dos dados, resultando em previsões imprecisas. No ajuste apropriado (gráfico central), o modelo consegue equilibrar a complexidade e a capacidade de generalização, capturando corretamente os padrões subjacentes nos dados e oferecendo boas previsões tanto para dados de treino quanto para dados de teste. No *overfitting* (gráfico à direita), o modelo é excessivamente complexo, ajustando-se aos detalhes e ruídos específicos dos dados de treino, o que leva a previsões imprecisas para novos dados. Esses cenários destacam a importância de escolher um modelo com a complexidade adequada para maximizar a performance preditiva.

Figura 2 - *Overfitting e Underfitting*

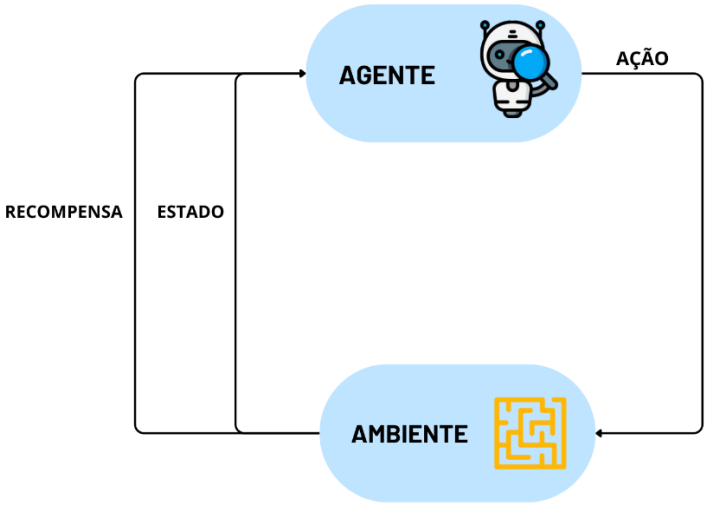


Fonte: (Autor Desconhecido)

2.1.3 Aprendizado por Reforço

A abordagem de aprendizado por reforço consiste em otimizar uma medida de recompensa ao longo do tempo. Nesse método um agente toma ações num ambiente que tem um mecanismo de recompensas ao longo do tempo. Assim, o ambiente retorna às ações com a alteração do estado de aprendizado no dado momento e uma recompensa que é proporcional a ação que foi tomada para que se alcance o objetivo desejado no modelo [5]. A figura 3 mostra o ciclo de aprendizado por reforço.

Figura 3- Aprendizado por Reforço



Fonte: (A autora, 2024)

Um dos maiores desafios do aprendizado por reforço atualmente, tem sido encontrar o equilíbrio entre as estratégias de explorar o ambiente e novos meios de aumentar os métodos de otimização de recompensas [5].

2.1.4 Modelos de Classificação *Machine Learning*

Modelos de classificação em aprendizado de máquina são algoritmos projetados para identificar e categorizar objetos com base em características específicas. Essas técnicas são amplamente usadas em atividades de todos os níveis de complexidade, desde detecções binárias à classificações multiclasses. O método de classificação

envolve o treinamento dos modelos com uma base de dados pré-processada, tendo pelo menos um exemplo de cada classificação já conhecida.

O objetivo é que, após aprender com os dados de exemplo, o modelo seja capaz de identificar corretamente a classe de novos dados não vistos antes, a partir de padrões e diferenças que distingam as classes desejadas na análise [2]. A comparação de modelos de classificação em *Machine Learning* é crucial para identificar o mais adequado para tarefas específicas. Esta comparação é essencial para garantir a segurança, operacionalidade e disponibilidade do sistema envolvido.

Como exemplo, no artigo [6] ilustra como diferentes algoritmos de classificação de *Machine Learning* foram avaliados para identificar o mais eficaz em detectar falhas estruturais em torres de transmissão. Os modelos de *Machine Learning*, incluindo Regressão Logística, *K-Nearest Neighbors* (kNN), *Random Forest*, *Support Vector Machine* (SVM) e Rede Neural Artificial (RNA), foram comparados. Cada modelo foi testado em sua capacidade de processar parâmetros de dispersão para diferenciar entre astes normais e defeituosas. Esse tipo de abordagem permite a adoção de técnicas não invasivas e não destrutivas, substituindo processos mais caros e invasivos, e destacando a importância de uma análise comparativa rigorosa para implementar a solução mais eficaz e confiável.

O presente estudo tem como objetivo comparar os modelos de classificação aplicados ao problema de faltas em linhas de transmissão. Serão comparados os seguintes modelos de classificação: Regressão Logística, *k-Nearest Neighbors* (kNN), *Random Forest, Support Vector Machine* (SVM) e Rede Neurais Artificiais (RNA).

2.1.4.1 Regressão Logística

A regressão logística é um modelo estatístico usado para descrever a probabilidade de uma variável dependente binária, ou seja, uma variável com duas categorias possíveis. Ela é frequentemente aplicada em problemas de classificação, onde o objetivo é determinar a qual das duas categorias uma observação pertence com base em uma ou mais variáveis independentes.

De acordo com o artigo [7], a regressão logística é classificada como um método paramétrico, que significa que ela modela a probabilidade de uma variável

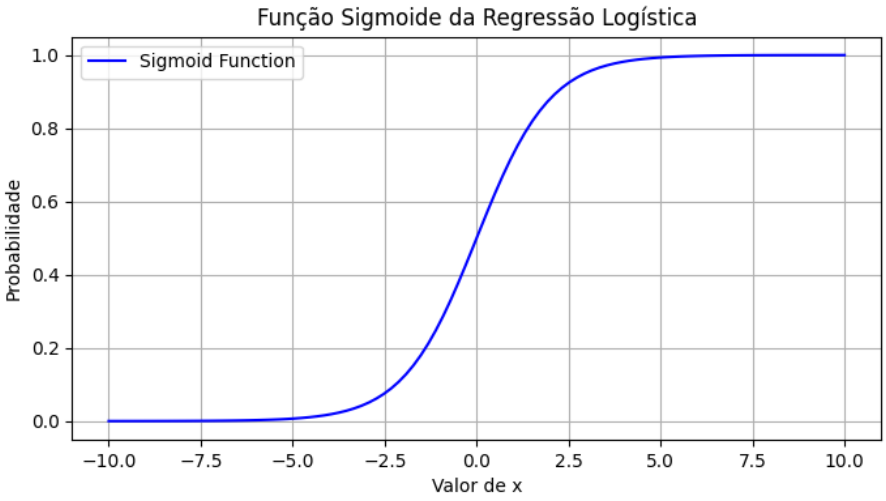
dependente, como uma função logística das variáveis independentes. Esse modelo é expresso matematicamente pela equação:

$$P(y|x) = \frac{1}{1 + e^{-(\beta 0 + \beta 1 x 1 + \ldots + \beta n x n)}} \quad (2.1.4)$$

Onde, $P(y|x)$ representa a probabilidade de um evento ocorrer dado as variáveis independentes x, $\beta 0$, $\beta 1$, ... , βn são os parâmetros do modelo que são estimados a partir dos dados. O modelo de regressão logística utiliza o método de máxima verossimilhança para estimar esses parâmetros, que maximizam a probabilidade dos dados observados sob o modelo especificado.

No contexto de classificação, a regressão logística é especialmente útil pois não apenas classifica um dado em uma das duas categorias, mas também fornece a probabilidade dessa classificação, o que pode ser uma informação valiosa na tomada de decisões. Além disso, [7] destaca que a regressão logística é frequentemente comparada com redes neurais artificiais em tarefas de classificação médica, demonstrando sua relevância e aplicabilidade contínua em contextos em que interpretações claras são necessárias, como em diagnósticos médicos e outras decisões clínicas.

Figura 4 - Modelo de Regressão Logística



Fonte: (A autora, 2024)

2.1.4.2 K-Nearest Neighbors (kNN)

O kNN é um algoritmo de aprendizado supervisionado utilizado frequentemente para classificação e regressão. No kNN, a classificação de uma nova amostra é realizada através da identificação dos 'k' vizinhos mais próximos no conjunto de treinamento, e a nova amostra é geralmente atribuída à classe que é mais frequente entre seus vizinhos mais próximos [8].

Na dissertação [9], o algoritmo kNN foi utilizado no contexto de classificação de faltas do tipo curto-circuito em linhas de transmissão. A escolha desse algoritmo se deve à sua simplicidade e eficácia em problemas de classificação, sendo especialmente adequado para lidar com dados multivariados, como formas de onda de tensão e corrente coletadas em sistemas elétricos de potência. O kNN classifica uma nova amostra com base na proximidade de k amostras conhecidas, utilizando uma medida de similaridade, geralmente a distância euclidiana, o que o torna intuitivo e fácil de implementar. A dissertação focou na classificação de faltas em um cenário pós-falta, sendo realizada após a detecção da falta, utilizando uma série temporal completa das formas de onda de tensão e corrente. Em resumo, o uso do kNN foi justificado pela sua simplicidade, eficácia e capacidade de lidar com dados multivariados e sequências temporais, tornando-se uma abordagem robusta e prática para a classificação de faltas em linhas de transmissão.

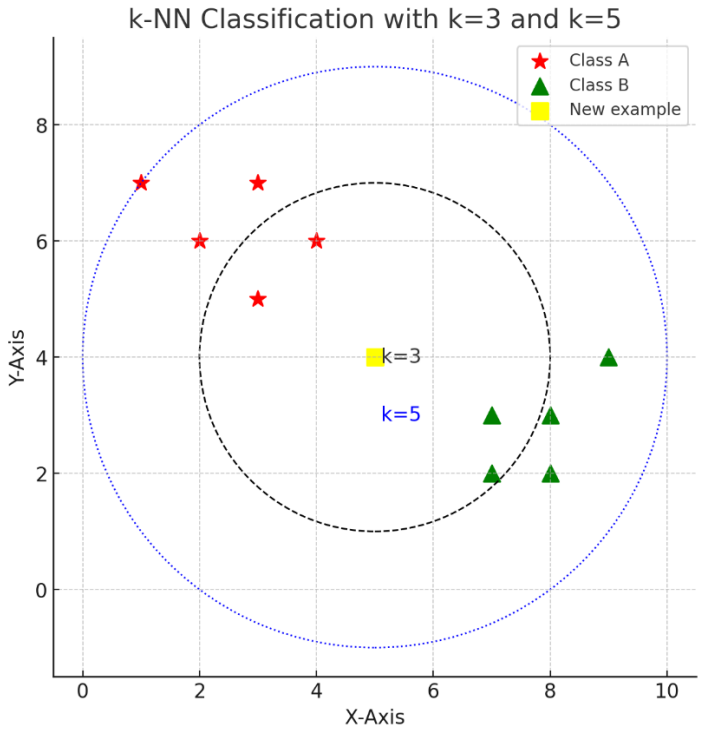
A Figura 5 ilustra o funcionamento do algoritmo k-Nearest Neighbors (kNN) para a classificação de um ponto desconhecido, destacando a importância da escolha do valor de 'k'. O ponto desconhecido, representado por um quadrado amarelo, precisa ser classificado como pertencente à Classe A (estrelas vermelhas) ou à Classe B (triângulos verdes). O valor de 'k' determina quantos vizinhos próximos serão considerados pelo algoritmo ao tomar essa decisão.

Quando k=3, o algoritmo examina os três vizinhos mais próximos do ponto desconhecido. Neste cenário, dois desses três vizinhos pertencem à Classe B, enquanto um pertence à Classe A. Consequentemente, a maioria dos vizinhos (dois de três) pertence à Classe B, e o ponto desconhecido é classificado como Classe B. Este exemplo demonstra como um valor menor de 'k' pode fazer com que o algoritmo seja sensível ao conjunto de dados local, o que pode incluir ruído.

Por outro lado, quando k=5, o algoritmo considera os cinco vizinhos mais próximos. Nesta situação, três dos cinco vizinhos pertencem à Classe A e dois pertencem à Classe B. Com a maioria (três de cinco) dos vizinhos pertencendo à Classe A, o ponto desconhecido é classificado como Classe A. Este exemplo mostra que um valor maior de 'k' pode proporcionar uma classificação mais robusta, suavizando a influência de qualquer dado atípico e levando a uma decisão que representa melhor a distribuição geral dos dados.

Esses exemplos evidenciam que a escolha do valor de 'k' no algoritmo kNN é crucial, pois diferentes valores podem levar a diferentes classificações para o mesmo ponto. Um valor de 'k' muito pequeno pode tornar o modelo suscetível ao ruído, enquanto um valor muito grande pode tornar o modelo excessivamente generalista, não capturando detalhes importantes. Portanto, a seleção apropriada de 'k' é essencial para equilibrar a precisão e a robustez do modelo.

Figura 5 - Classificação pelo kNN



Fonte: (A autora, 2024)

2.1.4.3 Random Forest

O modelo *Random Forest* (RF) é um algoritmo de *Machine Learning* utilizado para a classificação e detecção de falhas nos mais variados casos. *Random Forest* é uma extensão do método *Decision Tree*. A principal vantagem do RF sobre métodos tradicionais, como redes neurais e máquinas de vetores de suporte, reside na sua capacidade de manejar um grande volume de dados e sua habilidade em evitar o sobre ajuste, mesmo quando o número de variáveis é significativo [10].

A Figura 6 ilustra o funcionamento do algoritmo *Random Forest*, no centro do processo estão várias árvores de decisão, cada uma fazendo previsões independentes sobre a classe de uma instância de dados. Na figura, essas árvores de decisão são representadas em três cores diferentes: vermelho, azul e verde, destacando suas previsões individuais e independentes. Cada árvore de decisão analisa o mesmo conjunto de dados e gera uma previsão baseada em suas regras internas, construídas durante o treinamento. As árvores de decisão na RF são variadas, ou seja, cada árvore é ligeiramente diferente das outras porque o algoritmo utiliza diferentes amostras de dados e subconjuntos de características para treinar cada árvore. Este processo de variação ajuda a evitar o sobreajuste que pode ocorrer quando se utiliza uma única árvore de decisão.

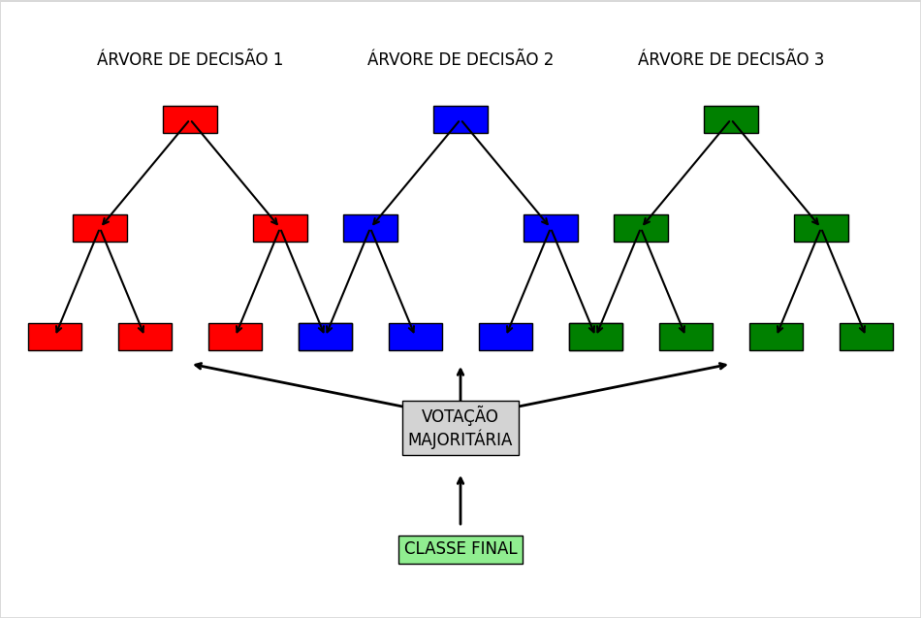
As previsões individuais de cada árvore são então combinadas através de um processo chamado votação majoritária. Na figura, isso é representado por setas que convergem de cada árvore de decisão para um retângulo cinza, que simboliza a votação majoritária. Nesse ponto, a classe que recebe a maioria dos votos entre todas as árvores de decisão é selecionada como a previsão final do modelo. Finalmente, o resultado da votação majoritária é indicado por uma seta que aponta para o retângulo verde na parte inferior da figura, representando a classe final atribuída à instância de dados. Este método de combinar múltiplas árvores de decisão ajuda a melhorar a precisão do modelo e a sua capacidade de generalização, tornando o Random Forest uma técnica poderosa e robusta em aprendizado de máquina.

O Random Forest foi aplicado para diagnosticar falhas em rolamentos de motores de indução utilizando sinais de vibração [11], que são capturados por acelerômetros. O RF opera construindo múltiplas árvores de decisão durante o processo de treinamento. Cada árvore é criada a partir de uma amostra dos dados de

treinamento, selecionada aleatoriamente com reposição — um método conhecido como *bagging* ou *bootstrap aggregation*. Cada árvore dá um voto na classificação final, e a classificação que recebe a maioria dos votos torna-se a previsão do modelo. O algoritmo também avalia a importância de cada característica (feature), que ajuda a compreender quais variáveis são mais influentes na previsão da falha do motor. Isso é crucial para a otimização de recursos e para focar em medidas mais impactantes em cenários de manutenção preditiva.

Nesse artigo, a eficácia do RF foi comparada com outros algoritmos de inteligência artificial, demonstrando melhor desempenho e maior precisão na classificação de falhas, o que reforça sua utilidade em aplicações de monitoramento da condição de equipamentos.

Figura 6 - Classificação por Random Forest



Fonte: (A autora, 2024)

2.1.4.4 Support Vector Machine (SVM)

Support Vector Machine (SVM) é um algoritmo de aprendizado de máquina utilizado para classificação binária. Este algoritmo, ao receber um conjunto de dados

contendo amostras de duas classes distintas (positivas e negativas), constrói um hiperplano que maximiza a margem de separação entre essas classes, criando assim uma superfície de decisão ótima [12].

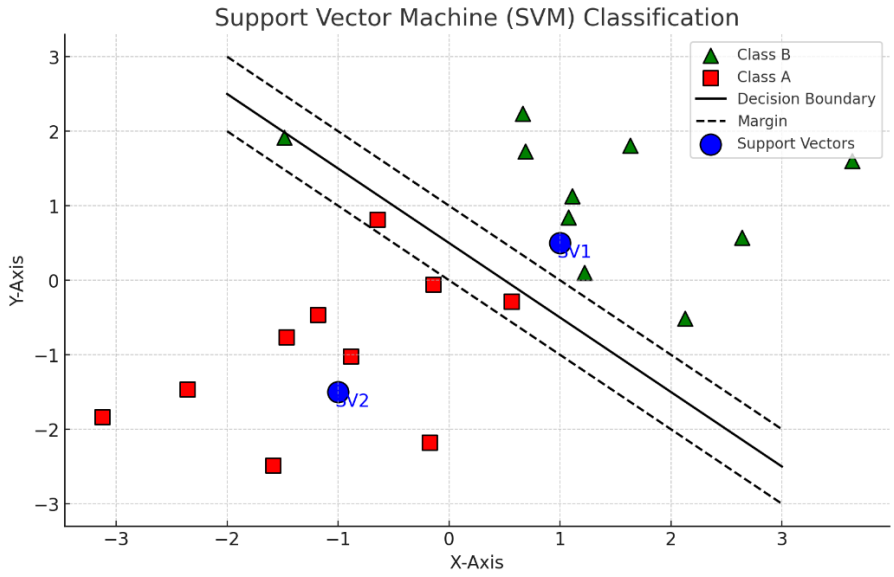
A Figura 7 ilustra a aplicação de uma Máquina de Vetores de Suporte (SVM) para a classificação de duas classes de dados, representadas por quadrados vermelhos (Classe A) e triângulos verdes (Classe B). A linha preta sólida no centro é o hiperplano de decisão que separa as duas classes, enquanto as linhas pretas tracejadas paralelas representam a margem, que é a distância máxima entre a linha de decisão e os pontos mais próximos de cada classe. Esses pontos mais próximos, destacados em azul e chamados de vetores de suporte, são cruciais para definir a posição do hiperplano. A maximização da margem é essencial para garantir uma melhor generalização do modelo para novos dados.

O artigo [13], apresenta uma abordagem que combina SVM com a Transformada de *Wavelet* (WT) para classificar tipos de falhas e prever a localização de falhas em linhas de transmissão de alta tensão. A metodologia emprega sinais de voltagem e corrente medidos em um terminal, utilizando transformada de Wavelet para reduzir o tamanho do vetor de características antes das etapas de classificação e predição.

No contexto do artigo, o SVM é aplicado de maneira a maximizar a capacidade de generalização do classificador. Isso é feito mapeando o espaço de entrada original para um espaço de produto ponto de alta dimensão, onde um hiperplano ótimo é determinado. Este hiperplano é encontrado usando a teoria de otimização e a Teoria Estatística da Aprendizagem. O SVM mostrou ter melhores propriedades de generalização em comparação com classificadores baseados em redes neurais artificiais (ANN), principalmente porque sua eficiência não depende do número de características, tornando-o particularmente útil em diagnósticos de falhas onde o número de características é grande.

Os resultados experimentais descritos no artigo indicam que o algoritmo proposto alcançou uma precisão de localização de falhas superior a 99%, com erros de classificação abaixo de 1% para todas as condições de falha testadas. Este alto nível de precisão demonstra a viabilidade do uso de SVM para aplicações de proteção de linhas de transmissão de energia.

Figura 7 - Classificação por Support Vector Machine



Fonte: (A autora, 2024)

2.1.4.5 Redes Neurais Artificiais (RNA)

Redes Neurais Artificiais (RNAs) são modelos computacionais de aprendizagem de máquina que imitam o cérebro humano, especialmente em aplicações de diagnóstico e prognóstico em sistemas complexos, como linhas de transmissão de energia elétrica [14].

A figura 8 apresenta a estrutura típica de uma rede neural artificial (RNA), mostrando suas três partes principais: a camada de entrada, a camada oculta, e a camada de saída. Na camada de entrada, os dados são recebidos e encaminhados para a camada oculta, onde neurônios intermediários processam as informações utilizando pesos ajustáveis. Esses pesos são otimizados durante o treinamento para minimizar o erro de previsão da rede. A camada de saída, composta por neurônios de saída, fornece o resultado da rede, que pode ser uma classificação ou um valor contínuo, dependendo da aplicação. As conexões entre os neurônios em diferentes camadas são ponderadas e ajustadas para capturar as relações complexas nos dados, permitindo que a rede aprenda e faça previsões ou classificações eficazes [15].

No artigo, [16] descreve RNAs como estruturas compostas por camadas de neurônios interconectados que processam informações através dessas conexões, onde cada neurônio simula uma operação de processamento de sinal neural. Essas

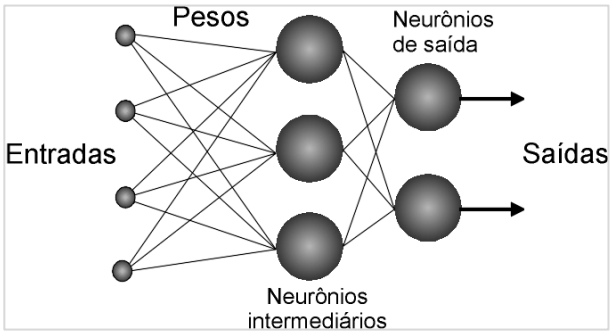
redes são capazes de aprender a partir de exemplos de dados, ajustando os pesos das conexões para minimizar erros na previsão ou classificação de dados novos, através de métodos como *backpropagation*. Esse procedimento consiste em duas fases principais: a propagação direta, que processa os dados de entrada através das camadas da rede para produzir uma saída, e a propagação reversa, na qual o erro entre a saída gerada e a desejada é determinado e enviado de volta através da rede. Este erro é utilizado para atualizar os pesos utilizando o algoritmo de gradiente descendente, aprimorando a capacidade da rede de executar tarefas como classificação e regressão com mais eficácia.

As RNAs são destacadas por suas capacidades de processamento paralelo e mapeamento não-linear, sendo ideais para o manejo de dados complexos e variados, como os encontrados em sistemas de transmissão de energia. Elas permitem uma adaptação contínua a novas condições sem a necessidade de reprogramação completa, o que é benéfico para a proteção e operação eficiente de sistemas de energia, oferecendo melhorias significativas na detecção e classificação de falhas, localização de falhas e discriminação de direção de falhas em relação aos métodos tradicionais.

De acordo com o artigo, as redes neurais artificiais (RNAs) têm uma aplicação significativa na classificação de falhas em linhas de transmissão de energia. Esse processo envolve a detecção e classificação de tipos de falhas, localização de falhas e discriminação da direção das falhas, essencial para a manutenção da estabilidade e segurança dos sistemas de energia.

A aplicação das RNAs nesse contexto se dá pela sua habilidade em aprender com dados históricos de falhas, permitindo que elas identifiquem padrões complexos nos dados de entrada, que são geralmente sinais de voltagem e corrente coletados de sensores ao longo da linha de transmissão. Após o treinamento, as RNAs podem eficientemente classificar o tipo de falha, determinar sua localização exata e até discernir a direção da falha, o que é crucial para ações corretivas rápidas e eficazes. Esta capacidade de análise e resposta rápida é fundamental para minimizar o tempo de inatividade do sistema e prevenir danos maiores, tornando as RNAs uma ferramenta valiosa para operações de linhas de transmissão.

Figura 8 - Redes Neurais Artificiais



Fonte: (Malcon Anderson, 1998)

2.2 Faltas na Rede Elétrica de Transmissão

No contexto da engenharia elétrica, o termo "falta" é frequentemente utilizado para descrever qualquer condição que interrompa o funcionamento normal de um sistema elétrico, geralmente resultando em uma condução anormal de corrente [17]. As faltas podem variar de interrupções simples e breves até complicações graves que podem causar danos significativos aos sistemas elétricos.

Estudar as falhas em linhas de transmissão é crucial porque são o elemento mais vulnerável do sistema elétrico. As linhas de transmissão cobrem grandes extensões e estão expostas a diversos riscos ambientais e humanos, como intempéries, descargas atmosféricas, poluição industrial, interferência de animais, vandalismo, entre outros. Cada um desses fatores pode causar faltas, resultando em interrupções no fornecimento de energia [17].

Devido à sua importância na entrega de energia gerada nas usinas até os pontos de consumo, qualquer falha nas linhas de transmissão pode ter impactos significativos na confiabilidade do sistema elétrico. Além disso, as características das linhas de transmissão, como a alta impedância, limitam a corrente de curto-circuito, o que torna a resposta a falhas mais complexa e exige um planejamento detalhado para garantir a proteção adequada.

Compreender as falhas nas linhas de transmissão permite o desenvolvimento e a implementação de medidas de proteção eficientes. Além disso, o estudo dessas falhas ajuda a identificar pontos fracos no sistema e a implementar estratégias de mitigação, como a poda de árvores próximas às linhas ou a instalação de para-raios. Essas

medidas não só aumentam a confiabilidade e a segurança do fornecimento de energia, mas também ajudam a reduzir custos associados a reparos e manutenção emergenciais. Portanto, estudar as falhas em linhas de transmissão é essencial para manter a integridade e a eficiência do sistema elétrico, garantindo um fornecimento contínuo e seguro de energia para consumidores e indústrias [18].

Os sinais de falta no sistema de transmissão são obtidos principalmente através de dispositivos de proteção instalados em vários pontos do sistema elétrico. Estes dispositivos são projetados para detectar anormalidades nas condições elétricas e, assim que uma falta é detectada, eles emitem sinais de alerta ou tomam medidas para isolar a parte afetada do sistema. Alguns dos dispositivos e métodos comuns para obter sinais de falta são:

- Relés de Proteção;
- Dispositivos de Medição;
- Sistemas SCADA (*Supervisory Control and Data Acquisition*);
- Sensores de Corrente e Tensão;
- Inspeções Visuais e Inspeções Aéreas.

Esses métodos e dispositivos trabalham em conjunto para fornecer uma vigilância contínua sobre o sistema de transmissão, permitindo uma resposta rápida e eficaz às faltas elétricas, minimizando os danos e as interrupções no fornecimento de energia [19].

2.2.1 Tipos de Faltas

No sistema elétrico, as faltas são classificadas em uma hierarquia com base na gravidade e na localização da falha, comumente as faltas elétricas, são organizadas das mais graves para as menos graves:

- Falta Trifásica;
- Falta Fase-Fase ou Fase-Fase-Terra;
- Falta Fase-Terra.

A identificação precisa e a classificação correta dessas falhas são fundamentais para a rápida restauração do serviço e para minimizar os impactos negativos.

2.2.1.1 Falta Trifásica

As faltas trifásicas representam um dos tipos mais severos de interrupções que podem ocorrer em sistemas elétricos de transmissão e distribuição. Esse tipo de falha acontece quando as três fases do circuito entram em curto simultaneamente, afetando todos os três canais de condução de energia ao mesmo tempo. Essa condição é especialmente crítica pois pode resultar em consequências substanciais para o funcionamento e a segurança do sistema elétrico. Devido à natureza abrangente da interrupção, essas faltas demandam uma atenção particular dos gestores de redes elétricas [20].

Uma característica marcante das faltas trifásicas é o impacto acentuado na tensão do sistema. A ocorrência desse tipo de falha pode provocar uma queda significativa na tensão em toda a rede, desestabilizando operações e podendo causar danos a dispositivos conectados. Além disso, a interrupção completa do fornecimento de energia é outra consequência direta, pois o curto-circuito nas três fases resulta em uma cessação total da entrega de eletricidade na região impactada.

As origens das faltas trifásicas são variadas, incluindo condições meteorológicas extremas, como tempestades e relâmpagos, defeitos ou falhas nos componentes do sistema elétrico, manipulações inadequadas ou erros operacionais durante manutenções, e interferências acidentais nas linhas, como contatos não intencionais durante atividades construtivas. Devido a essas múltiplas causas potenciais, o monitoramento e a manutenção preventiva são essenciais para minimizar a ocorrência desses eventos disruptivos.

No que se refere à identificação e resposta a essas falhas, relés de proteção desempenham um papel crucial, eles são empregados para detectar rapidamente as faltas trifásicas, monitorando as condições de corrente e tensão em tempo real [21]. Quando uma falha é identificada, sistemas automatizados de desligamento são

ativados para interromper o circuito afetado, limitando assim os danos ao sistema e facilitando a recuperação operacional.

Os impactos de uma falta trifásica são profundos, afetando desde consumidores residenciais até processos industriais. Podem ocorrer danos estruturais a equipamentos de alta tensão, como transformadores e linhas de transmissão, e há um elevado risco de incêndios ou explosões devido ao excesso de calor gerado pelas correntes de curto-circuito. Os procedimentos de recuperação após uma falha incluem exames detalhados e testes para certificar que os componentes e sistemas estão seguros e operacionais [17].

A figura 9 contém um diagrama que ilustram aspectos de um curto-circuito trifásico em um sistema elétrico. Esses diagramas são fundamentais para o estudo e a compreensão de faltas em sistemas de energia elétrica e são utilizados para simular e analisar as consequências dessas falhas.

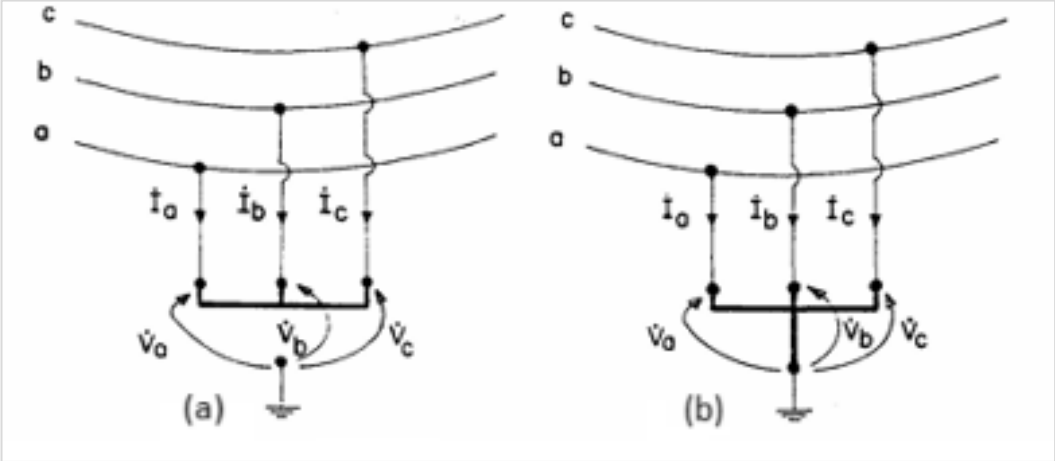
A figura 9(a) é representado onde todas as três fases (a, b, c) estão em curto-circuito diretamente entre si no mesmo ponto, sem a presença de uma impedância externa. Isso é indicado pelas tensões em cada fase (Va, Vb, Vc) serem iguais a zero no local do curto. Este tipo de curto-circuito é conhecido como curto-circuito trifásico puro ou sólido.

Ele é considerado o mais severo tipo de curto-circuito em um sistema trifásico, pois todas as fases estão conectadas diretamente uma à outra, permitindo que correntes muito altas fluam através do sistema, podendo causar danos significativos aos equipamentos e à infraestrutura elétrica. [20]

A figura 9(b) apresenta um diagrama de um sistema trifásico, destacando as linhas de transmissão das fases a, b e c, com suas respectivas correntes Ia, Ib e Ic fluindo para terra. As tensões Va, Vb e Vc são medidas em relação ao ponto de aterramento, indicado na parte inferior do diagrama.

Entender e gerenciar faltas trifásicas é crucial para manter a integridade e eficiência dos sistemas de distribuição de energia, assegurando que medidas adequadas de proteção e resposta estejam sempre em prática.

Figura 9- (a) Curto-Circuito Trifásico (b) Curto-Circuito Trifásico para Terra



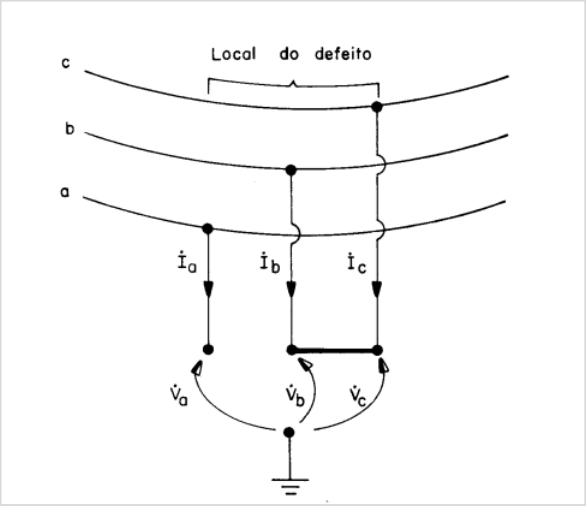
Fonte: (Geraldo Kinderemann, 1997)

2.2.1.2 Falta Fase-Fase

Uma falta fase-fase em um sistema elétrico trifásico ocorre quando duas fases diferentes entram em contato direto entre si, criando um caminho de baixa impedância que permite que correntes elevadas fluam entre essas fases. A Figura 10 mostra um curto-circuito entre as fases B e C no ponto F, enquanto a fase A permanece isolada para manter a simetria do sistema. As faltas fase-fase em um sistema elétrico trifásico é caracterizada pelo contato direto entre duas fases, resultando em correntes elevadas entre essas fases.

A imagem ilustra um diagrama de um sistema elétrico trifásico com um curto-circuito bifásico, onde as fases b e c entram em contato direto entre si, sem envolvimento da terra. Neste cenário, as correntes de falha (Ib) e (Ic) fluem entre as duas fases afetadas, resultando em uma queda significativa nas tensões das fases b (Vb) e c (Vc). A tensão da fase a (Va) pode permanecer mais estável, mas ainda pode ser indiretamente afetada. Este diagrama é fundamental para compreender o comportamento das correntes e tensões durante um curto-circuito bifásico, auxiliando no planejamento e implementação de estratégias de proteção eficazes [17].

Figura 10 - Curto-Circuito Fase-Fase.



Fonte: (Geraldo Kinderemann, 1997)

2.2.1.3 Falta Fase-Fase-Terra

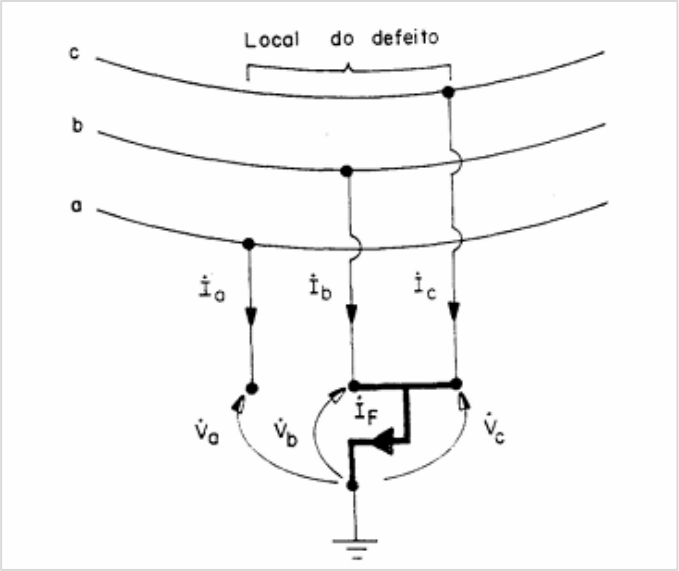
Uma falta fase-fase-terra em uma linha de transmissão de energia elétrica ocorre quando duas fases entram em curto-circuito e, ao mesmo tempo, entram em contato com a terra. No diagrama apresentado, vemos as fases a, b e c representadas pelas linhas horizontais. A área designada como "local do defeito" indica o ponto exato onde ocorre a falta, envolvendo duas fases e a terra. Esse tipo de falha pode ser causado por diversos fatores, como a queda de galhos de árvores sobre as linhas, isoladores danificados ou outras interferências físicas que resultam no contato entre as fases a e b com a terra [17].

Quando a falta fase-fase-terra ocorre, correntes de alta magnitude, designadas por Ia, Ib e Ic, fluem pelas fases a, b e c, respectivamente. A corrente de falta If é a corrente que resulta do curto-circuito e flui diretamente para a terra. Essa corrente de falta é significativamente maior do que a corrente normal de operação e pode causar danos severos aos equipamentos elétricos se não for rapidamente interrompida. As tensões nas fases afetadas, representadas por Va e Vb, tendem a cair drasticamente devido ao curto-circuito, enquanto a tensão na fase não afetada, Vc, pode sofrer distorções dependendo da gravidade da falta. A presença de uma falta fase-fase-terra

impõe a necessidade de um sistema de proteção robusto para detectar e isolar rapidamente a falha. Após a detecção e isolamento da falta, é fundamental que equipes de manutenção localizem o defeito físico na linha de transmissão e realizem os reparos necessários para restaurar a operação normal do sistema.

Em resumo, a falta fase-fase-terra é uma condição crítica que exige uma resposta rápida e eficiente dos sistemas de proteção para minimizar os danos e assegurar a continuidade da operação do sistema elétrico. A compreensão detalhada do comportamento das correntes e tensões durante esse tipo de falta é vital para o design, implementação e operação eficaz dos sistemas de proteção e manutenção de linhas de transmissão.

Figura 11 - Curto-Circuito Fase-Fase-Terra



Fonte: (Geraldo Kinderemann, 1997)

2.2.1.4 Falta Fase-Terra

Uma falta monofásica para terra em linhas de transmissão é uma condição em que uma única fase de um sistema trifásico entra em contato direto com a terra, resultando em um curto-circuito [17]. Esse tipo de falha é uma das mais comuns em sistemas de transmissão de energia elétrica e pode ser causada por diversos fatores,

como falhas de isolamento, contato com objetos estranhos (por exemplo, galhos de árvores) ou até mesmo por condições climáticas adversas.

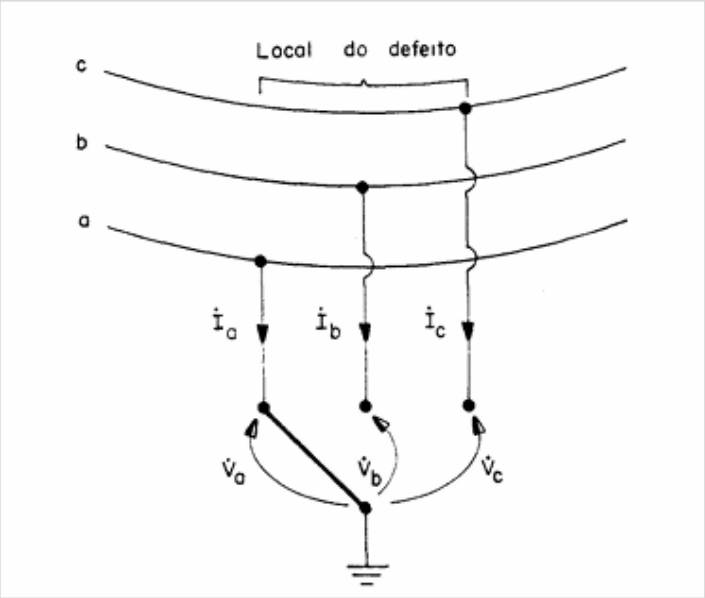
A figura 12 ilustra uma falta fase-terra em uma linha de transmissão de energia elétrica, destacando os principais elementos e fenômenos envolvidos nesse tipo de falha. As três linhas horizontais, identificadas como a, b e c, representam as três fases de um sistema trifásico de transmissão. O ponto marcado como "local do defeito" indica onde ocorre a falha, que neste caso envolve uma fase entrando em contato com a terra.

Quando isso ocorre, uma corrente de alta magnitude flui diretamente da fase afetada para a terra. Na figura 12 essa corrente é indicada como Ia na fase a. Essa corrente é muito maior do que a corrente de operação normal do sistema e pode causar danos significativos se não for rapidamente interrompida. A tensão na fase afetada (Va) cai drasticamente devido ao curto-circuito com a terra, enquanto as tensões nas outras fases (Vb e Vc) podem se alterar devido ao desequilíbrio causado pela falta.

Os sistemas de proteção são cruciais para detectar e isolar rapidamente a falta fase-terra. Após a interrupção da corrente de falta, equipes de manutenção devem localizar o defeito físico na linha de transmissão, identificar a causa (como um isolador quebrado ou contato com objetos externos) e realizar os reparos necessários para restabelecer a operação normal do sistema.

Em resumo, a falta fase-terra é uma das falhas mais comuns e perigosas em sistemas de transmissão de energia elétrica. Ela exige uma resposta rápida dos sistemas de proteção para minimizar os danos aos equipamentos e garantir a continuidade do fornecimento de energia. A compreensão detalhada do comportamento das correntes e tensões durante uma falta fase-terra é essencial para a configuração eficaz dos sistemas de proteção e para a realização de manutenções preventivas e corretivas [17].

Figura 12 - Curto-Circuito Fase-Terra



Fonte: (Geraldo Kinderemann, 1997)

3 METODOLOGIA

Neste capítulo, será explicada a metodologia envolvida no desenvolvimento das técnicas de *Machine Learning* utilizada para a classificação de tipos de faltas em linhas de transmissão. O processo abrange a escolha da base de dados, pré-processamento, detecção e tratamento de anomalias, balanceamento de classes, e a comparação de diversos algoritmos de *Machine Learning* para determinar o mais eficaz.

3.1 Definição do Problema

O problema de classificação de faltas em linhas de transmissão de energia elétrica é uma questão crítica que envolve a identificação e categorização de diferentes tipos de falhas que ocorrem em um sistema de transmissão. Essas falhas podem ser causadas por vários fatores, incluindo condições climáticas adversas, contato com objetos externos, defeitos nos isoladores, ou outros problemas técnicos. A classificação dessas falhas é essencial para uma resposta rápida e eficaz, garantindo a estabilidade e segurança do fornecimento de energia elétrica.

O objetivo principal do presente estudo é desenvolver um modelo de *Machine Learning* capaz de classificar com precisão os tipos de faltas em linhas de transmissão. As classes de falhas típicas a serem identificadas incluem faltas entre duas fases (AB, BC, AC), faltas entre três fases (ABC), faltas fase-terra (AG, BG, CG), faltas fase-fase-terra (ABG, BCG, ACG). A entrada para o modelo consiste em dados de corrente e tensão coletados em pontos específicos ao longo da linha de transmissão durante a ocorrência de uma falta, registrados em arquivos no formato .CSV contendo leituras em intervalos de tempo curtos (0,0012s cada amostra). A saída desejada é a classificação precisa do tipo de falta ocorrida.

A resolução do problema envolve vários passos importantes. Primeiramente, é necessária a simulação de dados, obtendo registros de corrente e tensão durante diferentes tipos de faltas e condições normais de operação. Em seguida, é crucial realizar o pré-processamento dos dados, o que inclui a limpeza dos dados para remover ruídos e inconsistências, normalização para padronizar as leituras, e a de

extração características relevantes, como desequilíbrios de fase e sinais de alta frequência (transitórios). Após o pré-processamento, os dados são divididos em conjuntos de treinamento e teste, geralmente na proporção de 80% para treinamento e 20% para teste.

A seleção e treinamento de modelos de *Machine Learning* são etapas cruciais. Diversos modelos, como Regressão Logística, RNA*, Random Forest*, kNN e SVM, serão testados e treinados com o conjunto de dados de treinamento. A avaliação dos modelos será feita usando métricas como acurácia, precisão, recall, F1-Score e matriz de confusão, permitindo a seleção do modelo com melhor desempenho.

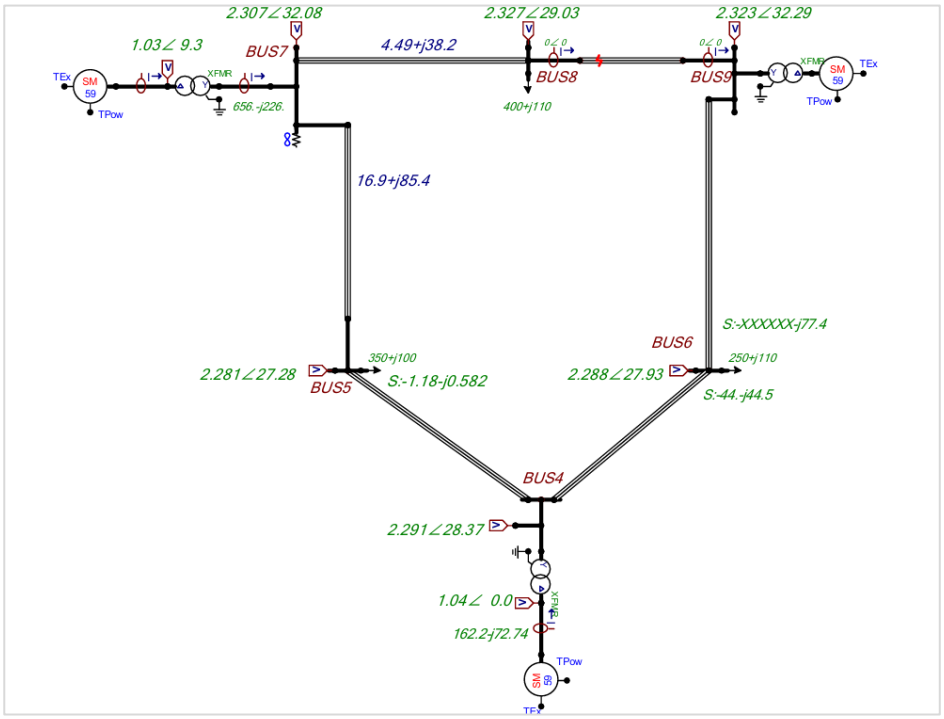
Lidar com a variabilidade e complexidade dos dados de corrente e tensão é um desafio, assim como garantir a precisão e robustez do modelo em condições variadas e inesperadas. Uma classificação precisa e rápida das faltas minimiza os danos e o tempo de inatividade, garantindo a continuidade do fornecimento de energia. Isso resulta em maior confiabilidade do sistema de transmissão e reduz os impactos negativos causados por interrupções no fornecimento de energia elétrica. Em última análise, a implementação eficaz deste processo contribui significativamente para a integridade do sistema de transmissão de energia elétrica e a segurança de seu fornecimento.

3.2 Base de Dados

A escolha da base de dados é crucial no papel de análises em aprendizado de máquina. Para o estudo atual foi utilizada uma base de dados que foi simulada no ATPDraw, onde o artigo "Localização de Faltas em Linhas de Transmissão de Energia Elétrica Utilizando as Redes Neurais Recorrentes LSTM e GRU" [22] aborda o uso de redes neurais recorrentes para identificar a localização de faltas em linhas de transmissão. A base de dados utilizada neste estudo é a Fault Analysis Database (FADb) [1], que é composta por simulações de faltas em uma rede de alta tensão baseada em um sistema elétrico de nove barramentos, apresentado na Figura 13. Esta base é pública e foi criada por Leandro A. Ensina em 2022.

[1] Base de Dados: https://onedrive.live.com/?authkey=%21ANXGIPHdvgIjeIs&id=CD803178CC7804B3%211377&cid=CD803178CC7804B3

Figura 13 - Simulação Linha de Transmissão 500 kV



Fonte: (Ensina, 2022)

A base de dados contém 168.000 simulações de faltas em uma linha de transmissão de 414 km de comprimento, cada simulação com uma janela de 0,012 s, com tensão de 500 kV e frequência de 60 Hz. Os dados foram coletados a uma taxa de amostragem de 10 kHz, armazenando informações de corrente e tensão para cada uma das três fases em ambos os terminais da linha. As simulações incluem diferentes tipos de falta, como AG, BG, CG, AB, AC, BC, ABG, ACG, BCG e ABC, com variação de localização entre 1% e 100% da extensão da linha, resistências de falta entre 0,01 e 200 Ω, e tempos de início da falta variando entre 0,091 s e 0,105 s.

O processo detalhado e estruturado que visa criar um conjunto abrangente de dados representativos de condições de falhas em sistemas de transmissão de energia elétrica. Esse processo envolve a criação de arquivos .csv que contêm informações cruciais sobre as medições de corrente e tensão em diferentes fases e pontos do sistema de transmissão, especialmente nos barramentos 8 e 9. Os nomes dos arquivos fornecem detalhes específicos sobre as condições de falha, como o momento de início da falha, o tipo de falha, a resistência da falha e a localização da falha ao longo da linha de transmissão.

A estrutura dos arquivos .CSV é cuidadosamente organizada para incluir colunas que representam medições detalhadas das tensões e correntes em várias fases. Por exemplo, colunas como v:X0014A, v:X0014B e v:X0014C registram as tensões nas fases A, B e C no barramento 9, respectivamente, enquanto colunas como c:X0013A:BUS8A, c:X0013B:BUS8B e c:X0013C:BUS8C registram as correntes nas fases A, B e C no barramento 8. Essas medições são essenciais para capturar a dinâmica do sistema durante as condições de falha e são utilizadas para treinar modelos de *Machine Learning* que podem identificar e classificar essas falhas com precisão.

O processo de simulação começa com a configuração do modelo no ATPDraw, utilizando o modelo IEEE de 9 barramentos como referência, conforme a figura 13. Diversos tipos de falhas são então definidos e simulados, incluindo faltas fase-fase, fase-terra e faltas envolvendo as três fases. Cada simulação é executada variando as condições, como a resistência da falta e a localização ao longo da linha de transmissão, para garantir que a base de dados seja abrangente e capture uma ampla gama de possíveis eventos de falta.

Durante cada simulação, os valores de tensão e corrente são registrados e salvos nos arquivos .csv correspondentes. A escolha de uma base de dados robusta a partir de simulações detalhadas é fundamental para o desenvolvimento de modelos de *Machine Learning* eficazes. Esses modelos são capazes de classificar diferentes tipos de falhas em linhas de transmissão de energia, o que é vital para garantir a estabilidade e a segurança do sistema elétrico. Além disso, a análise desses dados ajuda a identificar padrões de falhas, permitindo a implementação de medidas de manutenção preventiva e melhorando a confiabilidade geral da rede elétrica.

3.3 Preparação e Pré-processamento de Dados

A preparação e pré-processamento dos dados são etapas fundamentais no desenvolvimento de modelos de *Machine Learning* para a classificação de faltas em linhas de transmissão. Realizar os passos de pré-processamento são essenciais para assegurar que os dados sejam representativos e equilibrados, facilitando o

treinamento eficaz dos modelos de *Machine Learning* e melhorando sua performance na classificação de diferentes tipos de faltas.

3.3.1 Detecção e Tratamento de Anomalias

No processo de detecção e tratamento de anomalias, é utilizada a técnica do z-score para identificar valores atípicos nos dados. O *z-score* calcula a distância de um valor da média em termos de desvios padrão. Um valor de *z-score* alto ou baixo indica que o dado está longe da média, possivelmente sendo um outlier. Nos algoritmos, foi considerado valores com um z-score absoluto maior que 3 como anômalos e os removemos. Esta técnica é baseada no princípio estatístico conhecido como Regra 68-95-99.7, que indica que cerca de 99,7% dos dados de uma distribuição normal estão dentro de três desvios padrão da média. Ao remover esses outliers, asseguramos que os dados sejam mais representativos do comportamento típico do sistema, melhorando a precisão e a robustez dos modelos de *Machine Learning* subsequentes. Essa abordagem é amplamente aceita na análise estatística por sua eficácia na detecção de outliers [23].

3.3.2 Transformações dos Dados

A transformação dos dados, especificamente a normalização, é um passo crítico no pré-processamento de dados para *Machine Learning*. Nos códigos, utilizou-se a biblioteca *Dask* para aplicar a normalização de forma eficiente e distribuída, usando o *StandardScaler*. A normalização ajusta os dados para que tenham uma média de 0 e um desvio padrão de 1. Este passo é crucial porque muitos algoritmos de *Machine Learning*, como redes neurais e *k-Nearest Neighbors* (kNN), assumem que os dados estão normalizados. Sem normalização, variáveis com magnitudes diferentes podem influenciar de maneira desproporcional o treinamento do modelo. A normalização assegura que todas as características contribuam igualmente, prevenindo que variáveis com maior escala dominem o processo de aprendizado. Por fim, a normalização melhora a velocidade e a eficiência da convergência dos algoritmos de otimização, resultando em um treinamento mais rápido e eficiente. [24].

3.3.3 Balanceamento de Classes

O balanceamento de classes é um passo essencial quando se trabalha com conjuntos de dados desbalanceados, onde algumas classes estão sub-representadas. No código, utilizamos a técnica *SMOTE* (Synthetic Minority *Over-sampling Technique*) para equilibrar as classes. O *SMOTE* gera novos exemplos sintéticos para as classes minoritárias, criando dados através da interpolação entre exemplos existentes da mesma classe. Esta técnica aumenta a quantidade de dados das classes sub-representadas, resultando em um conjunto de dados mais equilibrado. O balanceamento de classes é vital para evitar que modelos de *Machine Learning* se tornem tendenciosos em relação às classes majoritárias, garantindo que todas as classes sejam igualmente representadas durante o treinamento. Isso é crucial para melhorar a capacidade do modelo de generalizar e fazer previsões precisas em situações reais, onde a distribuição das classes pode ser altamente desbalanceada [25].

3.4 Análise Exploratória e Visualização de Dados

A análise exploratória e a visualização de dados são cruciais para entender padrões, detectar anomalias e identificar relações nos dados, fornecendo percepções essenciais para a construção de modelos preditivos eficazes.

3.4.1 Análise Multivariada

A análise multivariada é um processo de examinar e interpretar múltiplas variáveis simultaneamente para entender as relações complexas e padrões nos dados. No algoritmo desenvolvido, essa análise começa com a carga de dados de múltiplos arquivos .CSV que contêm medições de variáveis, como tensões e correntes em várias fases e locais. A função *load_data* é usada para ler esses arquivos de maneira eficiente usando *Dask*, o que permite manipular grandes volumes de dados distribuídos em várias partições.

Após a carga, o pré-processamento dos dados envolve a remoção de duplicatas e preenchimento de valores ausentes com a média, assegurando que os dados estejam limpos e consistentes. A normalização dos dados através do *StandardScaler* é um passo crucial, pois ajusta os valores das variáveis para uma escala comum, com média zero e desvio padrão um. Isso é essencial em análise multivariada, permitindo que todas as variáveis contribuam de maneira equilibrada para o modelo.

A detecção e tratamento de anomalias utilizando o *z-score* garantem que valores extremos, que podem distorcer a análise, sejam removidos. Esta etapa é vital para a integridade dos dados, especialmente quando se lida com grandes volumes de dados multivariados, onde anomalias podem surgir devido a erros de medição ou eventos raros. A técnica *SMOTE* é então aplicada para balancear as classes no conjunto de dados, criando exemplos sintéticos das classes minoritárias. O balanceamento de classes é crucial em análise multivariada, pois assegura que todas as classes sejam representadas de maneira equitativa, permitindo que os modelos de *Machine Learning* detectem padrões complexos e façam previsões precisas.

3.5 Validação Cruzada

A validação cruzada com 5 folds foi utilizada para avaliar a performance dos cinco modelos de classificação (Regressão Logística, Rede Neural Artificial, *Random Forest*, kNN e SVM). Neste método, os dados são divididos em cinco partes (folds) de tamanho aproximadamente igual. Em cada iteração, quatro dessas partes são usadas para treinar o modelo, enquanto a quinta parte é utilizada para testar o modelo. Esse processo é repetido cinco vezes, cada vez utilizando um fold diferente para teste e os restantes para treino.

Esse procedimento garante que todos os dados sejam utilizados tanto para treino quanto para teste, permitindo uma avaliação mais robusta e menos suscetível a variações específicas do conjunto de dados. Os resultados das cinco iterações são então combinados para fornecer uma estimativa mais precisa das métricas de performance do modelo, como acurácia, precisão, recall, F1-score e matriz de confusão. A validação cruzada com 5 folds ajuda a assegurar que o modelo generalize

bem para dados não vistos, evitando o *overfitting* e proporcionando uma avaliação confiável da eficácia de cada modelo de classificação.

3.6 Visualizações Avançadas

As visualizações avançadas são ferramentas poderosas para explorar e entender a complexidade dos dados multivariados. No algoritmo, a função *plot_data* cria várias visualizações para analisar os dados de forma abrangente e detalhada. Os gráficos de linhas das tensões (v:X0014A, v:X0014B, v:X0014C) e correntes (c:X0013A:BUS8A, c:X0013B:BUS8B, c:X0013C:BUS8C) das fases permitem observar as variações e tendências ao longo do tempo. Esses gráficos são essenciais para identificar comportamentos anômalos, como picos ou quedas súbitas, que podem indicar falhas no sistema de transmissão.

Os histogramas das tensões e correntes fornecem uma visão clara das distribuições de frequência das variáveis, mostrando como os dados estão distribuídos e revelando a presença de outliers. Analisar a distribuição das variáveis ajuda a validar a eficácia das etapas de pré-processamento, como a normalização e a remoção de anomalias. Uma distribuição normal esperada após a normalização indicaria que a técnica foi aplicada corretamente.

A análise visual pode destacar padrões e relações entre variáveis que podem ser explorados mais profundamente durante a modelagem. Além disso, visualizações claras e informativas são essenciais para comunicar descobertas e justificar decisões baseadas nos dados a partes interessadas que podem não ter uma formação técnica.

4 DESENVOLVIMENTO DO TRABALHO

Para a análise, foi utilizada uma amostra representativa de cerca de 2% da base de dados completa, que se encontra no anexo 1. A base de dados completa é composta por um volume muito grande de dados, cerca de 168.000 arquivos de faltas cada um com aproximadamente 5.000 linhas e 12 colunas, o que torna o processamento e análise de todos os dados uma tarefa intensiva em termos de recursos computacionais, por não se possuir o processamento e os recursos necessários, para se processar a base por completo. Para realizar uma análise eficiente, foi selecionada um *sample* de cerca de 2% da base de dados total, que ainda é suficientemente representativa para obter resultados significativos sobre o desempenho dos modelos de *Machine Learning* para classificação.

4.1 Estrutura dos Arquivos .csv

Cada um dos 2.000 arquivos .csv contém cerca de 5.000 linhas e 12 colunas, resultando em 10 milhões de linhas no total da amostra. As colunas representam medições de corrente e tensão em diferentes pontos e fases do sistema, bem como os rótulos que indicam o tipo de falta correspondente. Essa estrutura permite que cada linha do arquivo represente um conjunto de medições em um determinado instante de tempo.

4.2 Pré-Processamento dos Dados

Inicialmente, o algoritmo desenvolvido em linguagem de programação de alto nível, Python, acessa arquivos armazenados em um ambiente de nuvem, pois foi utilizada a plataforma do Google Colab. Nesta etapa existe a integração dos dados armazenados remotamente, permitindo o acesso e manipulação direta dos arquivos CSV sem a necessidade de transferências manuais.

Em seguida, a função *load_data* foi projetada para ler arquivos CSV de forma eficiente, utilizando o Dask para processamento paralelo. Esta função permite a seleção apenas

das colunas necessárias, otimizando o uso da memória e acelerando o processo de carregamento dos dados, ainda sendo um sample da base de dados completa, ainda se trata de um volume muito grande de dados. O Dask é ideal para lidar com esse tipo de situação, distribuindo as operações em vários núcleos e realizando cálculos de forma lazy (apenas quando necessário).

A função *preprocess_data* serve para etapas de limpeza, como remoção de duplicatas ou preenchimento de valores ausentes. A função em seguida, *detect_and_treat_anomalies* segue a etapa inicial, aplicando um filtro baseado no z-score para eliminar anomalias. Este tratamento é essencial para assegurar que valores extremos não influenciem análises ou o treinamento de modelos de classificação em *Machine Learning*.

A normalização dos dados é realizada pela função *scale_data*, que utiliza o *StandardScaler* do *Dask* para ajustar as características numéricas de modo que tenham média zero e desvio padrão unitário. Esta normalização é crucial para algoritmos de *Machine Learning* que presumem que todas as características numéricas estejam na mesma escala.

Cada amostra de dados recebe um rótulo correspondente ao tipo de falha, com base no diretório de origem dos dados. Isso é realizado pela função *add_labels*. Após a rotulação, a função *balance_classes* implementa a técnica SMOTE, que por sua vez, equilibra as classes no conjunto de dados. O SMOTE é uma abordagem que sintetiza novas instâncias para as classes minoritárias, ajudando a mitigar o desbalanceamento de classes que pode levar a modelos enviesados.

As funções *process_files_in_folder_dask* e *process_all_folders_dask* são responsáveis por executar todas as etapas mencionadas acima para cada pasta de dados. Este DataFrame tratado é usado como entrada nos 5 algoritmos de classificação que foram utilizados no presente estudo.

4.3 Exploração dos Dados

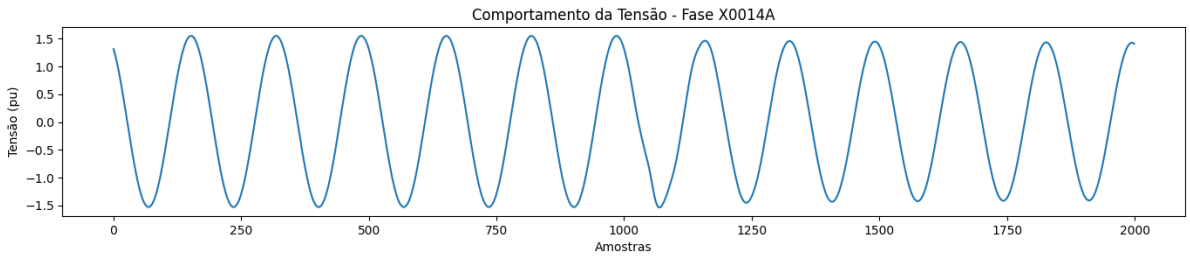
A exploração e visualização de dados são etapas fundamentais na análise de dados e *Machine Learning*, pois permitem uma compreensão profunda da estrutura, distribuição e características dos dados. Esse processo, conhecido como Análise

Exploratória de Dados, ajuda a identificar padrões, tendências e anomalias, além de revelar possíveis problemas como valores ausentes ou duplicados. Através de gráficos e visualizações, é possível observar relações entre variáveis e tomar decisões informadas sobre as técnicas de pré-processamento necessárias, como normalização e preenchimento de valores ausentes.

No contexto da análise de faltas em linhas de transmissão, essa etapa é crucial para identificar comportamentos anômalos nos dados de tensão e corrente. Gráficos de linha e histogramas, por exemplo, podem mostrar variações abruptas e desbalanceamentos de fase, indicando possíveis faltas ou anomalias no sistema elétrico. Optou-se por usar apenas uma das fases para estas visualizações simplificando a análise dos formatos de onda e permitir uma compreensão mais clara do comportamento da tensão ao longo do tempo. Focar em uma única fase ajuda a destacar o padrão senoidal típico e facilita a detecção de qualquer irregularidade que possa ocorrer durante a amostragem.

A Figura 14 apresenta o gráfico de linha que mostra o comportamento da tensão na fase A ao longo de 2000 amostras. No eixo horizontal (x), representa as amostras, no tempo em unidades discretas, enquanto no eixo vertical (y), temos a tensão medida em unidades por unidade (pu). Observa-se que a tensão exibe um padrão senoidal regular, característico de um sistema elétrico operando em condições normais. Este comportamento esperado confirma que, durante o intervalo de amostragem mostrado, não há perturbações ou faltas significativas na fase A em termos de tensão.

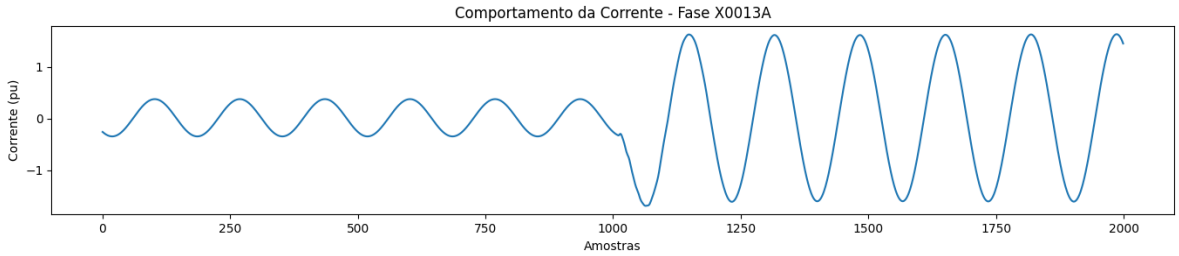
Figura 14 - Comportamento das Tensão na Fase A durante amostragem



Fonte: (A autora, 2024)

A Figura 15 apresenta o gráfico de linha que mostra o comportamento da corrente na fase A ao longo de 2000 amostras. Este gráfico é utilizado para visualizar o formato de onda da corrente durante um período de amostragem, permitindo a identificação de padrões e anomalias no sistema elétrico.

Figura 15 - Comportamento da Corrente na Fase A durante amostragem



Fonte: (A autora, 2024)

No eixo horizontal (x), temos as amostras, que representam o tempo em unidades discretas, enquanto no eixo vertical (y), temos a corrente medida em unidades por unidade (pu). Observa-se que a corrente exibe um padrão senoidal regular até a amostra 1000. Após esse ponto, ocorre uma perturbação significativa, onde a corrente sofre uma queda abrupta antes de se recuperar e continuar com um padrão senoidal. Esta variação abrupta é indicativa de uma possível falta ou anomalia na linha de transmissão. A corrente inicialmente apresenta uma oscilação regular, mas a queda brusca sugere a presença de um evento transitório, curto-circuito, que impacta momentaneamente o comportamento da corrente.

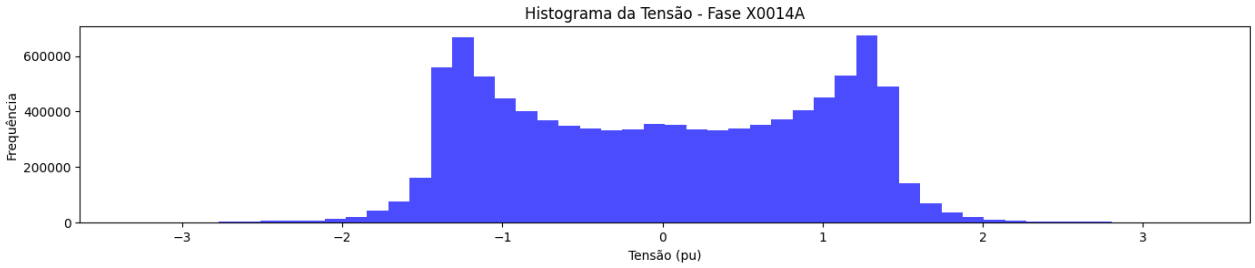
A diferença entre os gráficos de tensão e corrente durante uma falta é atribuída aos mecanismos de regulação e proteção presentes no sistema elétrico. A tensão é mantida estável por reguladores e transformadores que reagem rapidamente para isolar a falha e assegurar que a tensão permaneça dentro dos limites operacionais, mesmo durante distúrbios. Esse controle explica por que a tensão permanece constante e exibe um padrão senoidal regular, como mostrado na Figura 14.

Por outro lado, a corrente é diretamente influenciada por qualquer anomalia ou falta no sistema, como um curto-circuito, que reduz a impedância e provoca um aumento súbito na corrente. Durante uma falta, a corrente apresenta variações

abruptas, visíveis na Figura 15, devido à resposta imediata dos sistemas de proteção que atuam para isolar a falha. Assim, enquanto a tensão permanece controlada, a corrente registra as alterações rápidas causadas pela falta, refletindo o impacto direto das anomalias no sistema de transmissão.

A Figura 16 apresenta um histograma da tensão na fase X0014A, mostrando a distribuição dos valores de tensão ao longo das amostras. No eixo horizontal (x), temos a tensão medida em unidades por unidade (pu), enquanto o eixo vertical (y) representa a frequência de ocorrência desses valores. Esses picos indicam que a maioria das medições de tensão se concentra nesses valores, sugerindo que a tensão oscila predominantemente dentro desses intervalos durante o período de amostragem.

Figura 16 - Histograma da Tensão na Fase A

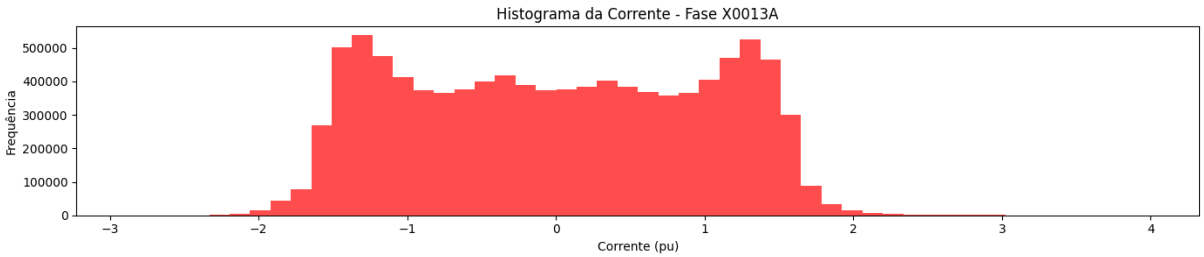


Fonte: (A autora, 2024)

A forma simétrica do histograma ao redor do zero também revela que a tensão na fase X0014A tem um comportamento equilibrado, com valores positivos e negativos distribuídos de maneira semelhante. Essa visualização é útil para entender a variação e a distribuição da tensão, fornecendo informações sobre o desempenho e a estabilidade da linha de transmissão ao longo do tempo.

A Figura 17 apresenta um histograma da corrente na fase X0013A, ilustrando a distribuição dos valores de corrente ao longo das amostras. No eixo horizontal (x), temos a corrente medida em unidades por unidade (pu), enquanto o eixo vertical (y) representa a frequência de ocorrência desses valores. Observa-se que a distribuição é mais dispersa comparada à da tensão. Esses picos indicam que a maioria das medições de corrente se concentra nesses valores, sugerindo que a corrente oscila predominantemente dentro desses intervalos durante o período de amostragem.

Figura 17 - Histograma da Corrente na Fase A



Fonte: (A autora, 2024)

A forma geral do histograma revela uma assimetria, com um maior número de valores negativos em comparação aos positivos, e uma distribuição mais larga em torno de zero. Isso pode ser indicativo de variações de eventos transitórios que afetam a corrente de maneira mais significativa. A presença de valores extremos, tanto negativos quanto positivos, pode sugerir a ocorrência de faltas ou anomalias no sistema, que resultam em variações abruptas da corrente.

5 RESULTADOS E ANÁLISES DOS MODELOS

Para o projeto de classificação de faltas em linhas de transmissão, foram desenvolvidos cinco algoritmos distintos, cada um representando um modelo de classificação: *Random Forest*, Regressão Logística, SVM (*Support Vector Machine*), Rede Neural Artificial (RNA) e kNN (*k-Nearest Neighbors*). Esses algoritmos foram implementados e testados com o objetivo de comparar sua eficácia na detecção e classificação das faltas.

No Anexo 1 da dissertação, há uma página dedicada onde todos os notebooks dos cinco modelos estão disponíveis. Cada notebook inclui a implementação detalhada do algoritmo correspondente, cobrindo desde a configuração inicial até o treinamento e a avaliação dos modelos. Além disso, os notebooks contêm as visualizações das métricas de desempenho, como matrizes de confusão, acurácia, precisão, recall e F1-Score, bem como gráficos que mostram a distribuição dos acertos de cada modelo.

5.1 Desempenho Regressão Logística

Após os dados serem carregados e pré-processados, eles são divididos em características (X) e rótulos (y). As características (X) consistem nas medições de tensão e corrente, enquanto os rótulos (y) indicam o tipo de falta em cada registro. Esse processo assegura que os dados estejam prontos para serem utilizados no treinamento do modelo de *Machine Learning*.

A Regressão Logística, escolhida como modelo de classificação, é então integrada a esse pipeline para treinar e prever os tipos de faltas. Para avaliar o desempenho do modelo, é configurada a validação cruzada K-Fold, com 5 *folders* (K=5). Neste método, os dados são divididos em 5 partes iguais. Em cada iteração, o modelo é treinado em 4 dessas partes e testado na parte restante. Esse processo é repetido 5 vezes, alternando a parte utilizada para teste em cada iteração. Isso permite uma avaliação robusta do modelo, pois garante que cada amostra dos dados seja utilizada tanto para treinamento quanto para teste.

Durante cada iteração da validação cruzada, os índices de treinamento e teste são gerados. O conjunto de dados de treinamento é balanceado e treinado usando o pipeline, e o modelo treinado é então utilizado para fazer previsões no conjunto de teste. As previsões são comparadas com os rótulos reais para calcular a acurácia. Além disso, são gerados gráficos que comparam a distribuição das previsões com a dos valores reais. Esses gráficos de barras sobrepostas (histogramas) fornecem uma visualização clara de como o modelo está performando em termos de previsões corretas e incorretas para cada classe de falta. Os resultados das iterações da validação cruzada mostram as métricas de desempenho do modelo, incluindo tempo de treinamento, acurácia, precisão, recall e F1-Score para cada iteração. Essas métricas são cruciais para entender como o modelo está performando e onde pode haver melhorias. As médias dessas métricas após todas as iterações fornecem uma estimativa geral do desempenho do modelo. As médias indicam que o modelo tem uma acurácia de aproximadamente 18,57%, uma precisão de 17,63%, um recall de 18,57% e um F1-Score de 16,88%, conforme indicado na Tabela 1.

Tabela 1 - Métricas de Desempenho Regressão Logística

Métrica	**Valor**
Acurácia	0.1857
Precisão	0.1763
Recall	0.1857
F1-Score	0.1688

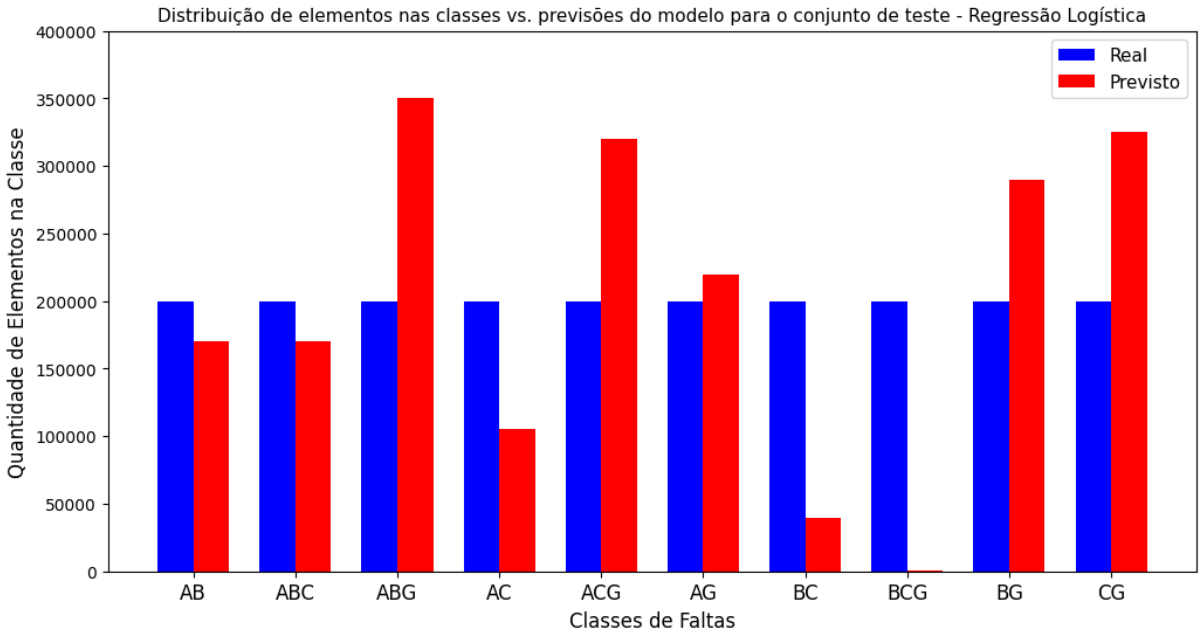
Fonte: (A autora, 2024)

O tempo de treinamento do modelo para cada iteração variou significativamente, indo de aproximadamente 69,37 segundos a 360,50 segundos, com um tempo de treinamento médio total de 83,49 segundos. Esse tempo de treinamento é facilitado pela utilização de uma instância do Google Compute Engine com suporte a TPU (Unidade de Processamento Tensor). As especificações da máquina incluem 334.6 GB de RAM e 225.3 GB de espaço em disco, tornando-a adequada para o processamento de grandes volumes de dados e para o treinamento de modelos de *Machine Learning* que exigem recursos intensivos. A presença da TPU

sugere que essa configuração é otimizada para acelerar o treinamento de algoritmos de *Machine Learning*, tornando-a ideal para tarefas complexas e de grande escala.

A Figura 18 mostra a distribuição das previsões do modelo de Regressão Logística comparada aos valores verdadeiros para cada classe de falhas nas linhas de transmissão. As barras azuis representam a densidade dos valores verdadeiros de cada classe, enquanto as barras vermelhas representam a densidade das previsões feitas pelo modelo.

Figura 18 - Desempenho do modelo Regressão Logística no conjunto teste



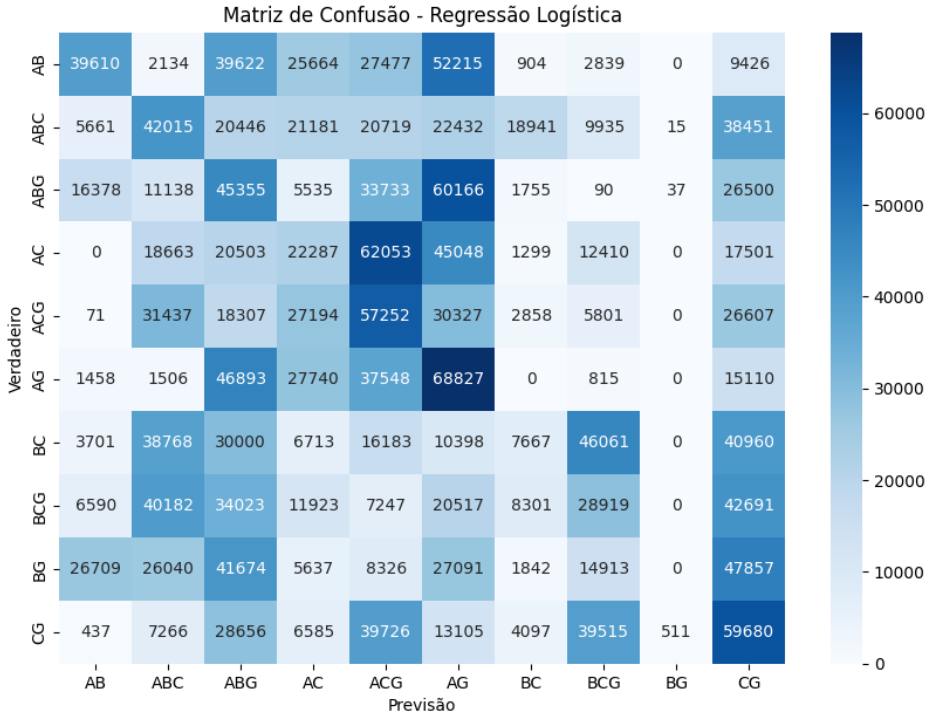
Fonte: (A autora, 2024)

Observa-se que, para algumas classes, como ABG, ACG, BG e CG, há uma discrepância significativa entre as previsões e os valores reais. Por exemplo, a classe ABG é repetidamente escolhida pelo modelo, com uma densidade de previsões muito maior do que a densidade dos valores reais. Isso indica que o modelo tende a prever excessivamente essa classe em comparação com outras classes. Em contraste, a classe BCG é rara de ser identificada pelo modelo, com a densidade de previsões sendo significativamente menor do que a densidade dos valores reais, o que mostra que o modelo raramente prevê essa classe corretamente. Essa distribuição desigual

sugere que o modelo tem dificuldades em distinguir corretamente entre certas classes de falhas.

A Figura 19 apresenta a matriz de confusão do modelo de Regressão Logística, cada linha representa a classe verdadeira (real), enquanto cada coluna representa a classe prevista pelo modelo. A diagonal principal, onde as classes previstas coincidem com as classes reais, mostra as previsões corretas feitas pelo modelo. Os valores mais altos nesta diagonal indicam um melhor desempenho do modelo para essas classes específicas. Por exemplo, o modelo apresenta um desempenho relativamente bom na previsão da classe AG, como evidenciado pelos valores elevados na diagonal correspondente a essas classes.

Figura 19 - Matriz de Confusão – Regressão Logística



Fonte: (A autora, 2024)

No entanto, a matriz de confusão também revela que há confusões significativas entre diferentes classes. Observa-se que a classe ABG é frequentemente prevista erroneamente como outras classes, como AB e BCG. Este padrão de erro sugere que o modelo tem dificuldades em distinguir corretamente entre essas classes, devido à similaridade nas características dos dados.

5.2 Desempenho Random Forest

Depois da fase de pré-processamento dos dados, o código implementado realiza uma série de etapas para balanceamento de classes, treinamento e avaliação do modelo de *Random Forest*. Depois de dividir os dados em conjuntos de treinamento e teste, é aplicado o SMOTE, para garantir que todas as classes sejam representadas de forma justa durante o treinamento.

O modelo *Random Forest* é então configurado com hiperparâmetros específicos para otimizar o uso de memória e a eficiência do treinamento. A poda das árvores de decisão é realizada para controlar o crescimento excessivo das árvores e melhorar a generalização do modelo. Sem poda, as árvores de decisão podem crescer até capturar perfeitamente os dados de treinamento, criando uma árvore extremamente complexa que memoriza os dados de entrada. Essa ferramenta é utilizada para reduzir o *overfitting*, onde o modelo se ajusta muito bem aos dados de treinamento, mas perde a capacidade de generalizar para novos dados. Isso resulta em um modelo que tem excelente desempenho nos dados de treinamento, mas desempenho ruim nos dados de teste.

Ao podar as árvores, restringimos a profundidade máxima de cada árvore e o número mínimo de amostras necessárias para dividir um nó ou para formar uma folha. No código, a profundidade máxima *(max_depth*) foi definida como 10, o número mínimo de amostras para dividir um nó (*min_samples_split*) foi definido como 10 e o número mínimo de amostras em cada folha (*min_samples_leaf*) foi definido como 5. Esses parâmetros garantem que as árvores não cresçam excessivamente complexas, o que melhora a capacidade do modelo de generalizar para dados não vistos.

Modelos de Random Forest com árvores muito profundas e complexas requerem mais tempo e recursos computacionais para treinamento e inferência. Ao limitar a profundidade das árvores e o tamanho dos nós, reduz-se a quantidade de tempo e memória necessários para construir e utilizar o modelo.

Para avaliar o desempenho do modelo, é utilizada a validação cruzada K-Fold com 5 *folders*, neste método, o conjunto de dados de treinamento balanceado é dividido em 5 partes iguais. O modelo é treinado em 4 dessas partes e testado na parte restante, repetindo esse processo 5 vezes, alternando a parte utilizada para teste em cada iteração. Isso permite uma avaliação robusta do modelo, garantindo

que cada amostra dos dados seja utilizada tanto para treinamento quanto para teste. Durante cada iteração, o tempo de treinamento é medido e registrado, fornecendo informações sobre a eficiência do modelo.

O treinamento do modelo foi realizado em uma instância do Google Compute Engine com suporte a TPU (Unidade de Processamento Tensor). As especificações da máquina incluem 334.6 GB de RAM e 225.3 GB de espaço em disco.

Após a validação cruzada, as previsões são comparadas com os rótulos reais para calcular métricas de desempenho, como acurácia, precisão, recall e F1-Score. O tempo total de treinamento foi de aproximadamente 2128,76 segundos, e as métricas de desempenho indicam uma acurácia de 92,67%, precisão de 92,89%, recall de 92,67% e F1-Score de 92,63%, conforme mostrado na Tabela 2.

Tabela 2 - Métricas de Desempenho Random Forest.

Métrica	**Valor**
Acurácia	0.9267
Precisão	0.9289
Recall	0.9267
F1-Score	0.9263

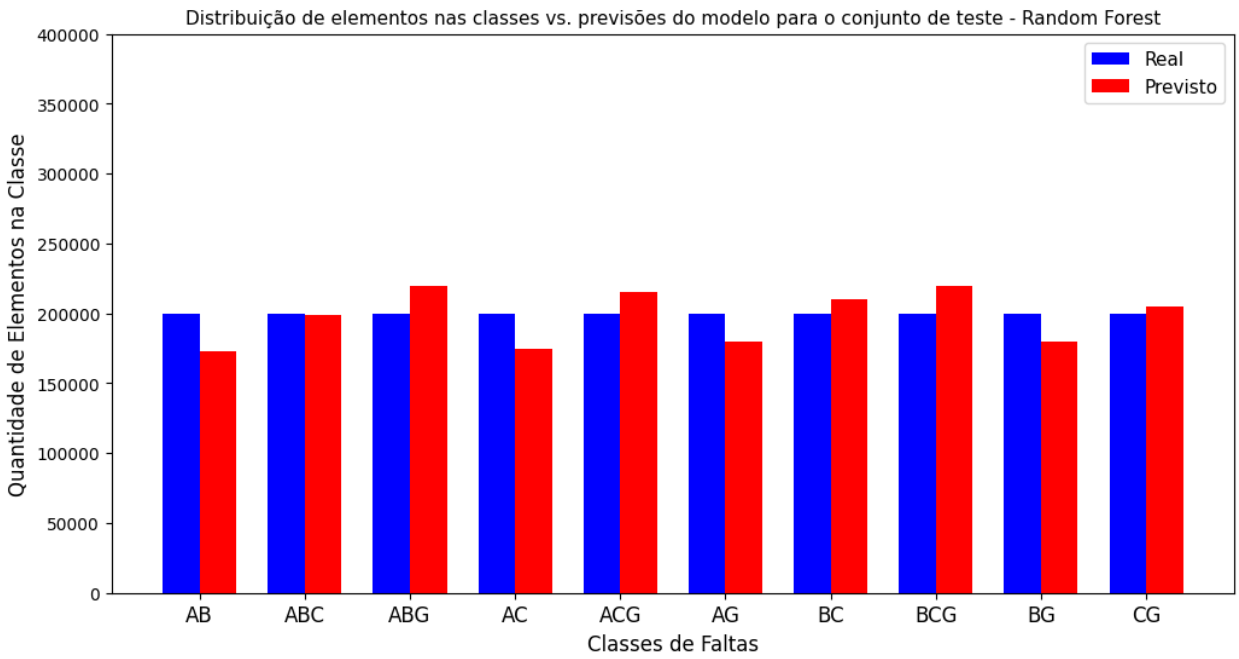
Fonte: (A autora, 2024)

Para visualizar os resultados, são geradas uma matriz de confusão e um gráfico de distribuição de previsões versus valores reais. A matriz de confusão mostra o número de previsões corretas e incorretas para cada classe, ajudando a identificar onde o modelo está se saindo bem e onde ele está falhando. O gráfico de distribuição compara visualmente a densidade das previsões do modelo com os valores reais, fornecendo uma visão clara de como o modelo está performando em termos de previsões corretas e incorretas para cada classe de falha. Essas visualizações são ferramentas importantes para analisar o desempenho do modelo e planejar melhorias futuras.

A Figura 20 mostra a distribuição das previsões do modelo de *Random Forest* comparada aos valores reais para cada classe de falhas nas linhas de transmissão. As barras azuis representam a densidade dos valores verdadeiros de cada classe de falha, enquanto as barras vermelhas representam a densidade das previsões feitas

pelo modelo. Observa-se que, para a maioria das classes, como ABC, BCG e CG, as previsões (barras vermelhas) estão relativamente alinhadas com os valores reais (barras azuis). Por exemplo, para a classe ABG, as previsões superam os valores reais, indicando que o modelo tende a superestimar essa classe. Essas discrepâncias sugerem que o modelo de *Random Forest* enfrenta dificuldades em distinguir corretamente entre certas classes de falhas.

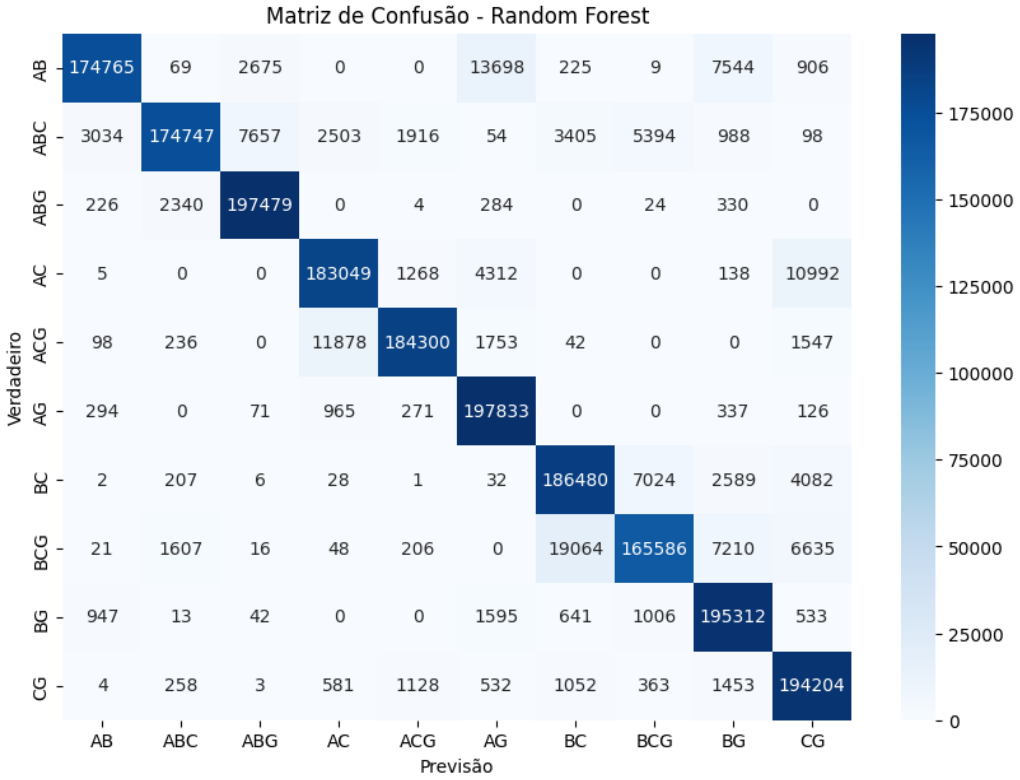
Figura 20 - Desempenho do modelo Random Forest no conjunto teste



Fonte: (A autora, 2024)

A matriz de confusão apresentada na Figura 21 ilustra o desempenho do modelo Random Forest na previsão dos tipos de faltas nas linhas de transmissão. Nesta matriz, cada linha representa a classe verdadeira (real), enquanto cada coluna representa a classe prevista pelo modelo. A diagonal principal mostra as previsões corretas, onde a classe prevista coincide com a classe real.

Figura 21 - Matriz de Confusão – Random Forest



Fonte: (A autora, 2024)

Os valores na diagonal principal indicam que o modelo possui um desempenho robusto na maioria das classes. Por exemplo, para a classe AB, o modelo fez 174.765 previsões corretas, e para a classe CG, fez 194.204 previsões corretas. Esses números refletem a capacidade do modelo em identificar corretamente a maioria das ocorrências para essas classes.

No entanto, a matriz também revela algumas confusões entre classes, indicando dificuldades do modelo em distinguir entre certos tipos de faltas. Essas confusões podem ser atribuídas a similaridades nas características dos dados entre essas classes.

Esses resultados sugerem que o modelo Random Forest, com a configuração de poda das árvores, oferece um desempenho bom na classificação de faltas em linhas de transmissão. O ajuste dos hiperparâmetros para limitar o crescimento das árvores ajudou a evitar o *overfitting* e a melhorar a precisão das previsões. A análise visual das previsões versus os valores reais e a matriz de confusão confirma a eficácia do modelo, indicando que ele é uma escolha robusta para esta aplicação.

5.3 Desempenho kNN

Com os dados de treino balanceados, o próximo passo é treinar o modelo de k-Nearest Neighbors (kNN). A validação cruzada é realizada usando a técnica de K-Fold, que divide o conjunto de dados em K partes (folds). O modelo é treinado K vezes, cada vez utilizando um fold diferente como conjunto de teste e os K-1 folds restantes como conjunto de treino. Isso ajuda a garantir que o modelo seja validado em diferentes subconjuntos dos dados, fornecendo uma avaliação mais robusta de sua performance. Durante cada iteração, são calculadas métricas de desempenho como acurácia, precisão, recall e F1-Score, e uma matriz de confusão é gerada para cada fold.

Após a validação cruzada, são calculadas as métricas médias de desempenho do modelo, incluindo a acurácia, precisão, recall e F1-Score. Além disso, uma matriz de confusão média é gerada para visualizar o desempenho do modelo em termos de classificações corretas e incorretas para cada classe. Essas métricas fornecem uma visão abrangente da performance do modelo durante o processo de validação, permitindo ajustes e melhorias, se necessário, antes da avaliação final.

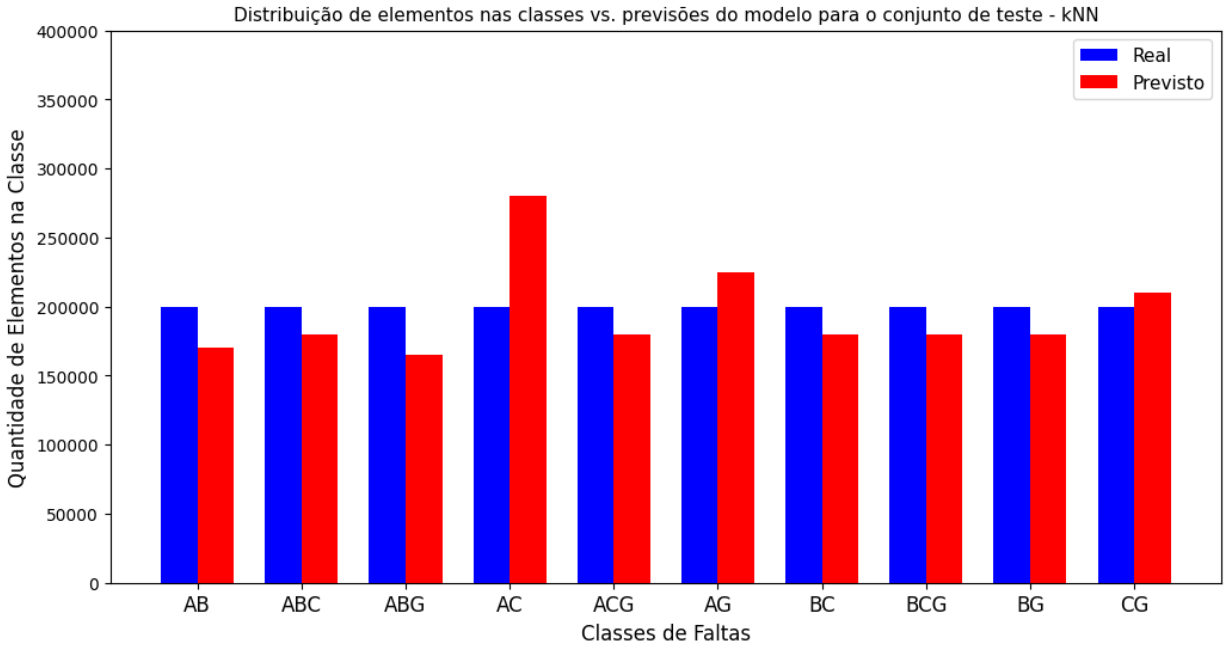
Com o modelo treinado e validado, ele é então ajustado utilizando todo o conjunto de treino balanceado, e a avaliação final é realizada no conjunto de teste separado inicialmente. Isso fornece uma avaliação final da performance do modelo em dados não vistos durante o treinamento, garantindo que o modelo generalize bem para novos dados. As métricas de desempenho no conjunto de teste são calculadas e apresentadas, incluindo acurácia, precisão, recall, F1-Score e a matriz de confusão.

O treinamento do modelo foi realizado em uma instância do Google Compute Engine com suporte a TPU (Unidade de Processamento Tensor). As especificações da máquina incluem 334.6 GB de RAM e 225.3 GB de espaço em disco.

A Figura 22 mostra o gráfico de barras comparando as previsões do modelo kNN com os valores verdadeiros para diferentes classes de falhas revela insights sobre o desempenho do modelo. Para a maioria das classes, como AB, ABC, ABG, BCG e CG, o modelo mostra uma correspondência razoavelmente boa entre as previsões e os valores verdadeiros. No entanto, classes como AC e AG destacam-se por uma superestimação significativa, onde o número de previsões é maior do que o número real de ocorrências. Essa superestimação pode indicar que o modelo está

confundindo outras classes com AC e AG, resultando em previsões inflacionadas para essas classes.

Figura 22 - Desempenho do modelo kNN no conjunto teste



Fonte: (A autora, 2024)

Após a avaliação do modelo kNN utilizando validação cruzada, foram obtidas as seguintes métricas de desempenho para o conjunto de dados: acurácia de 81,04%, precisão de 82,40%, recall de 81,04% e F1-Score de 81,39%. Essas métricas indicam um desempenho bom do modelo. A acurácia e o recall, ambos em torno de 81%, sugerem que o modelo é capaz de identificar corretamente uma boa parte dos exemplos de teste. A precisão de 82,40% indica que a maioria das previsões positivas do modelo são corretas, enquanto o F1-Score de 81,39% mostra um bom equilíbrio entre precisão e recall, conforme mostrado na Tabela 3.

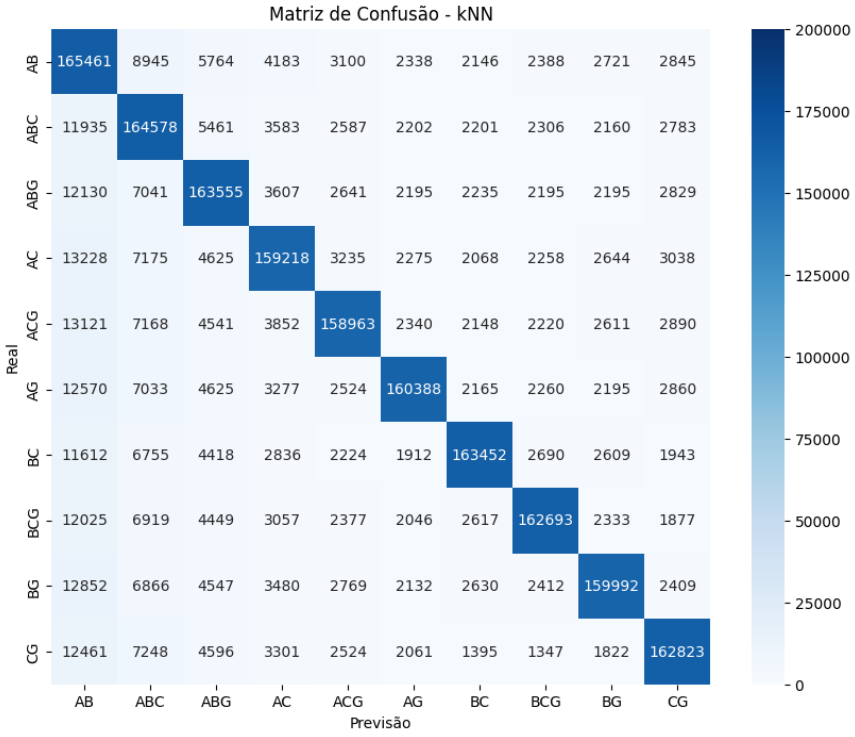
Tabela 3 - Métricas de Desempenho kNN

Métrica	**Valor**
Acurácia	0.8104
Precisão	0.8240
Recall	0.8104
F1-Score	0.8139

Fonte: (A autora, 2024)

A matriz de confusão, mostrada na Figura 23, gerada durante a validação cruzada, fornece uma visão detalhada sobre o desempenho do modelo em cada classe de falha. O modelo tem uma alta taxa de acertos para as classes AB e ABC, mas há algumas confusões com outras classes, como AB sendo confundida com ABC e vice-versa. Para a classe ABG, embora a taxa de acerto seja alta, há uma quantidade significativa de confusões com classes como AB e ABC.

Figura 23 - Matriz de Confusão – kNN



Fonte: (A autora, 2024)

Em resumo, o modelo kNN demonstra um desempenho geral bom, com métricas de acurácia, precisão, recall e F1-Score em torno de 81% a 82%. No entanto, a matriz de confusão revela que o modelo ainda enfrenta dificuldades em distinguir entre algumas classes de falhas. Essas discrepâncias sugerem que, embora o modelo kNN esteja performando bem em geral, há espaço para melhorias.

5.4 Desempenho Support Vector Machine

O modelo de SVM foi implementado utilizando a biblioteca *scikit-learn*, que fornece uma interface robusta para a construção e ajuste de modelos de aprendizado de máquina. No algoritmo implementado, o modelo SVM foi treinado e validado utilizando a técnica de validação cruzada com KFold, dividindo os dados de treinamento em cinco subconjuntos.

A implementação do SVM, envolveu a escolha de parâmetros específicos para otimizar o desempenho e garantir a eficiência do treinamento. O parâmetro max_iter=1000 define o número máximo de iterações que o otimizador deve executar. Estabelecer um limite para o número de iterações evita que o modelo fique preso em loops de treinamento extensivos e assegura que o processo de treinamento seja concluído em um tempo razoável. O valor de 1000 iterações foi selecionado com base na complexidade dos dados e na necessidade de permitir que o modelo convergisse adequadamente sem gastar recursos computacionais excessivos.

Outro parâmetro importante é tol=1e-3, que define a tolerância como critério de parada para o otimizador. Quando as mudanças no valor da função objetivo ficam abaixo deste valor, o otimizador interrompe o processo de treinamento. Um valor de 0,001 foi escolhido para garantir que o modelo alcance uma solução suficientemente precisa sem necessidade de iterações adicionais desnecessárias. O parâmetro random_state=42 foi utilizado para assegurar a reprodutibilidade dos resultados, garantindo que o processo de divisão dos dados e inicialização do modelo seja consistente em diferentes execuções.

O treinamento do modelo foi realizado em uma instância do Google Compute Engine com suporte a TPU (Unidade de Processamento Tensor). As especificações da máquina incluem 334.6 GB de RAM e 225.3 GB de espaço em disco.

O modelo de SVM foi avaliado utilizando validação cruzada, resultando nas seguintes métricas de desempenho: uma acurácia de 78,71%, precisão de 79,34%, recall de 78,71% e F1-Score de 77,65%, conforme mostrado na Tabela 4. Estas métricas indicam um desempenho moderado do modelo. O tempo total de treinamento foi de aproximadamente 11.971,3 segundos, refletindo o tempo significativo necessário para treinar um modelo SVM em um conjunto de dados complexo como este.

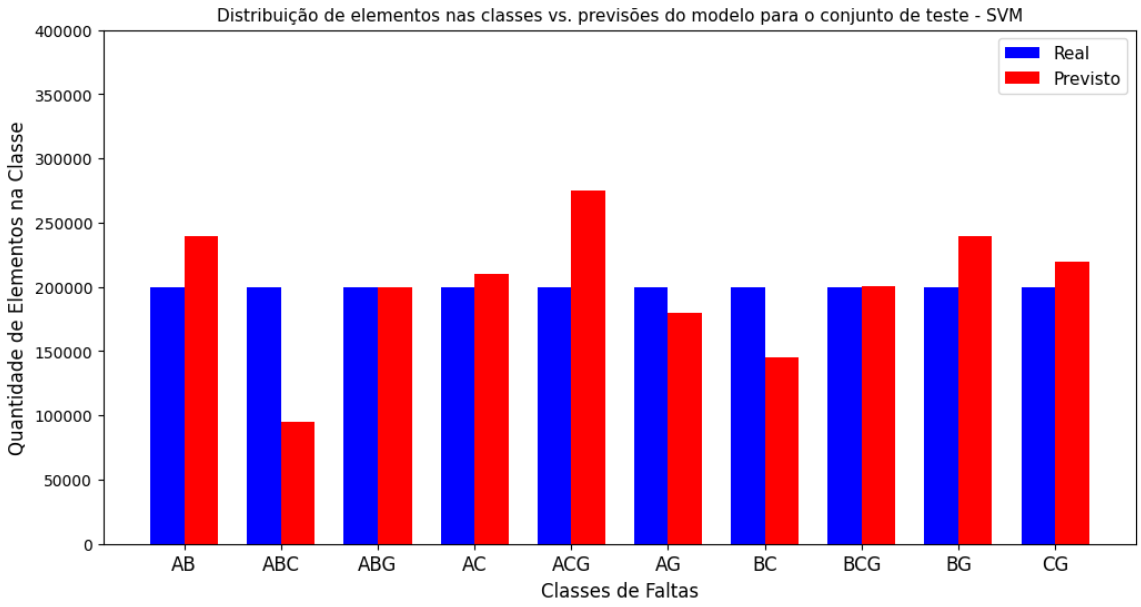
Tabela 4 - Métricas de Desempenho SVM

Métrica	Valor
Acurácia	0.7871
Precisão	0.7934
Recall	0.7871
F1-Score	0.7765

Fonte: (A autora, 2024)

O gráfico de barras compara as previsões do modelo SVM com os valores verdadeiros para diferentes classes de falhas. Observa-se que, para algumas classes, como AB, AC e CG, as previsões estão bem alinhadas com os valores verdadeiros. No entanto, há discrepâncias notáveis em outras classes. Por exemplo, a classe ABC apresenta uma superestimação significativa, enquanto a classe AG mostra uma subestimação. Essa variação na precisão das previsões sugere que o modelo SVM tem dificuldades para distinguir certas classes de falhas de maneira consistente. A análise do gráfico indica que, embora o modelo SVM consiga capturar corretamente a maioria das classes, ele ainda apresenta problemas de precisão que podem ser melhorados com ajustes adicionais nos parâmetros do modelo ou técnicas de pré-processamento de dados mais avançadas.

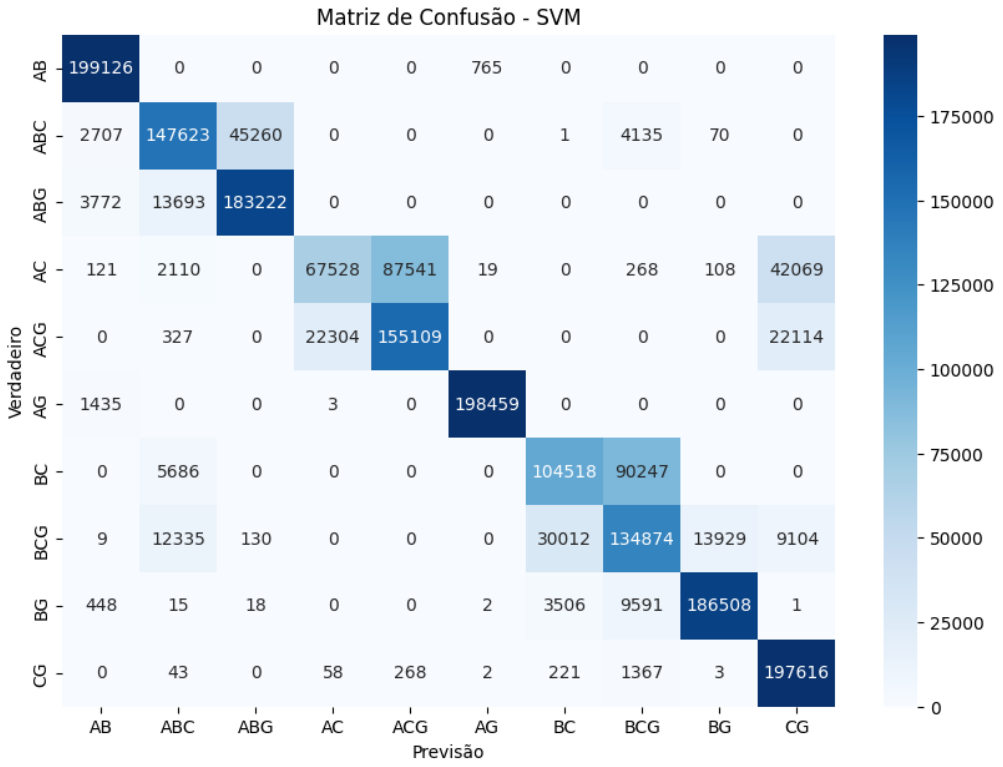
Figura 24 - Desempenho do modelo SVM no conjunto teste



Fonte: (A autora, 2024)

A matriz de confusão para o modelo SVM, exibida na figura, fornece uma visão detalhada das previsões corretas e incorretas feitas pelo modelo. Cada célula da matriz representa o número de previsões para cada combinação de classes verdadeiras e previstas. Os valores na diagonal principal indicam as previsões corretas. Observa-se que o modelo SVM tem boas previsões para algumas classes, como AB e AG, mas apresenta confusões significativas em outras classes. Por exemplo, a classe ABC tem 45.260 previsões incorretas, e a classe BG tem 18.650 previsões incorretas com outras classes. Essas confusões indicam que o modelo SVM pode estar enfrentando dificuldades em diferenciar entre classes com características semelhantes. A análise da matriz de confusão sugere que há espaço para melhorias, como o ajuste de hiperparâmetros ou o uso de técnicas de balanceamento de classes para melhorar a precisão do modelo.

Figura 25 - Matriz de Confusão – SVM



Fonte: (A autora, 2024)

Em conclusão, o modelo SVM apresenta um desempenho moderado com algumas limitações em termos de precisão e recall. As análises do gráfico de barras

e da matriz de confusão indicam que, embora o modelo consiga capturar corretamente muitas das classes, ainda há áreas onde melhorias podem ser feitas para aumentar a precisão e reduzir as confusões entre classes. Ajustes adicionais no modelo e nas técnicas de pré-processamento de dados podem ajudar a alcançar um desempenho mais robusto e confiável.

5.5 Desempenho Redes Neurais Artificiais

A rede neural utilizada foi construída com a biblioteca TensorFlow e Keras, que são ferramentas poderosas para construção e treinamento de modelos de aprendizado profundo. A estrutura da rede é composta por várias camadas densas e camadas de *dropout* para evitar *overfitting*. A camada de entrada recebe os dados de entrada com a forma específica do conjunto de dados pré-processado. A primeira camada oculta consiste em 256 neurônios com a função de ativação 'relu' e uma camada de *dropout* de 0,3 para prevenir *overfitting*. A segunda camada oculta consiste de 128 neurônios com a função de ativação 'relu' e uma camada de dropout de 0,2 para prevenir *overfitting*. A camada de saída possui um número de neurônios igual ao número de classes de faltas, com uma função de ativação '*softmax*' para fornecer a probabilidade das classes.

O KFold foi utilizado para dividir o conjunto de dados em várias partes (*folds*), permitindo uma validação cruzada robusta, o que ajuda a avaliar o desempenho do modelo de maneira mais confiável. Além disso, o *LabelEncoder* foi utilizado para transformar rótulos categóricos em rótulos numéricos, necessários para o treinamento da rede neural.

O treinamento do modelo foi realizado em uma instância do Google Compute Engine com suporte a TPU (Unidade de Processamento Tensor). As especificações da máquina incluem 334.6 GB de RAM e 225.3 GB de espaço em disco.

Utilizando validação cruzada, foram obtidas as seguintes métricas de desempenho: acurácia de 95,34%, precisão de 95,37%, recall de 95,61% e F1-Score de 95,43%, conforme mostrado na Tabela 5. O tempo médio de treinamento foi de aproximadamente 5399,15 segundos, refletindo a complexidade e o poder do modelo para a tarefa de determinar classes.

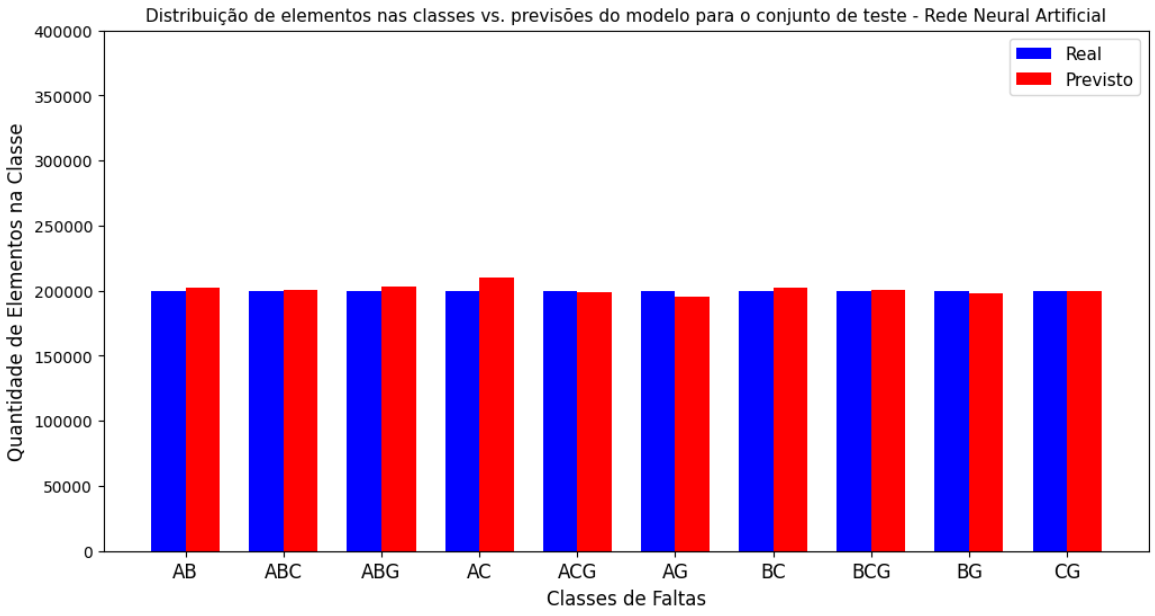
Tabela 5 - Métricas de Desempenho RNA

Métrica	Valor
Acurácia	0.9534
Precisão	0.9537
Recall	0.9561
F1-Score	0.9543

Fonte: (A autora, 2024)

O gráfico de barras apresentado na Figura 26, mostra a comparação entre as previsões do modelo de Rede Neural Artificial (RNA) e os valores verdadeiros para diferentes classes de faltas. Cada barra azul representa a contagem de ocorrências verdadeiras para cada classe, enquanto cada barra vermelha mostra as previsões do modelo. A análise revela que, em geral, o modelo está fazendo previsões bastante precisas, com as barras azul e vermelha sendo quase idênticas para a maioria das classes. Isso é indicativo de um bom desempenho do modelo, especialmente nas classes AB, ABC, ABG, AC, ACG, BC e CG, onde as previsões estão bem alinhadas com os valores verdadeiros. No entanto, algumas discrepâncias podem ser observadas. Por exemplo, para a classe AG, o modelo subestima ligeiramente as ocorrências, enquanto para a classe CG, o modelo parece superestimar as previsões.

Figura 26 - Desempenho do modelo Rede Neural Artificial no conjunto teste

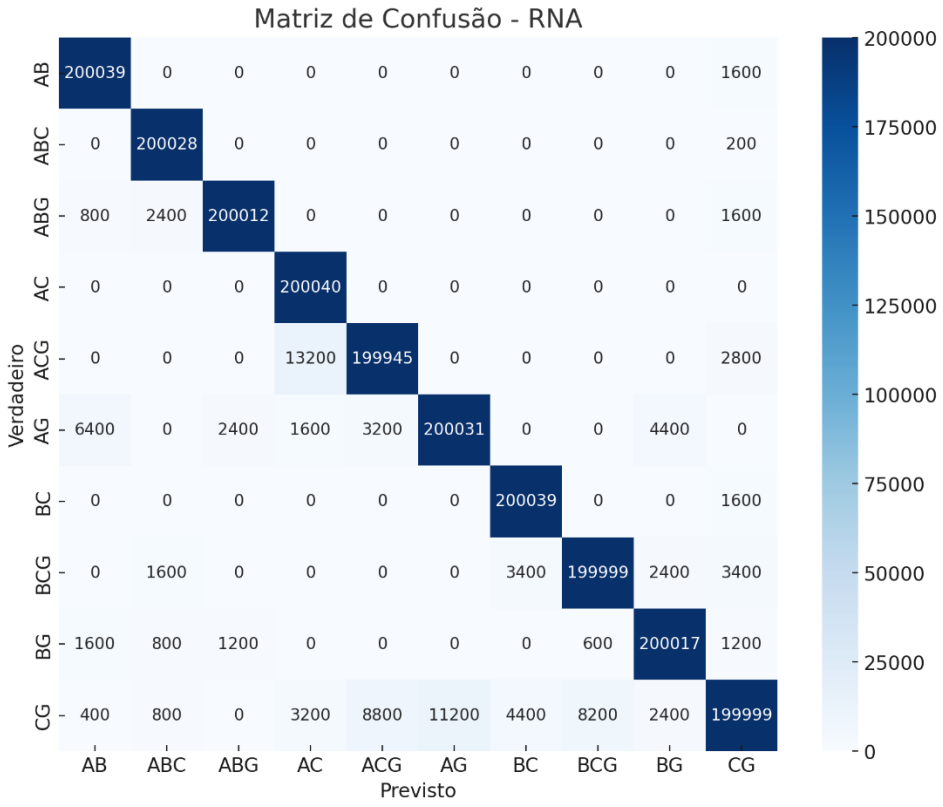


Fonte: (A autora, 2024)

A Figura 27 exibe a matriz de confusão para o modelo de RNA, ilustrando o desempenho do modelo na classificação de diferentes tipos de falhas em linhas de transmissão. Cada célula da matriz representa o número de previsões feitas pelo modelo para cada combinação de classes verdadeiras (no eixo vertical) e previstas (no eixo horizontal). Os valores na diagonal principal indicam as previsões corretas, onde a classe prevista coincide com a classe verdadeira. As células fora da diagonal indicam as previsões incorretas, mostrando onde o modelo confundiu uma classe com outra.

Observa-se que o modelo RNA tem um desempenho muito bom, com altos valores na diagonal principal para todas as classes de falha. A confusão entre classes como AG e AB ou AG e CG pode ser devido a similaridades nas características das falhas, que o modelo encontra difícil de distinguir.

Figura 27 - Matriz de Confusão – RNA



Fonte: (A autora, 2024)

A rede neural construída neste projeto demonstrou um desempenho excelente na classificação de falhas em linhas de transmissão. As ferramentas e parâmetros utilizados, foram essenciais para garantir a eficiência do processamento de dados e a precisão do modelo. O uso de camadas densas com funções de ativação *relu* e camadas de *dropout* ajudou a prevenir *overfitting*, resultando em um modelo robusto e preciso. As métricas de desempenho elevadas refletem a eficácia da abordagem adotada, destacando a capacidade da rede neural de identificar corretamente quase todas as classes de falhas.

5.6 Análise Comparativa dos Modelos

Com base na análise dos cinco modelos de Machine Learning: Regressão Logística, RNA, kNN, Random Forest e SVM para a tarefa de classificação de faltas em linhas de transmissão, foi possível identificar que o modelo de Redes Neurais Artificiais (RNA) apresentou as melhores métricas de desempenho. Realizou-se um levantamento comparativo detalhado, explicitando as métricas de desempenho de cada modelo e destacando as razões pelas quais a RNA se sobressaiu em relação aos demais. A Tabela 6, a seguir mostra todas as métricas de desempenho dos modelos analisados.

Tabela 6 – Desempenho dos modelos

Modelo	Acurácia (%)	Precisão (%)	Recall (%)	F1-Score (%)	Tempo de Treinamento (segundos)
Regressão Logística	**18,58**	**17,68**	**18,58**	**16,83**	**83,49**
Redes Neurais Artificiais (RNA)	**95,34**	**95,37**	**95,61**	**95,43**	**5399,15**
K-Nearest Neighbors (kNN)	**81,03**	**82,40**	**81,03**	**81,39**	**1346,20**
Random Forest	**92,67**	**92,88**	**92,67**	**92,63**	**2128,76**
Support Vector Machine	**78,71**	**79,37**	**78,71**	**77,65**	**11971,30**

Fonte: (A autora, 2024)

A Regressão Logística, obteve uma acurácia de 18,58%, precisão de 17,68%, recall de 18,58% e F1-Score de 16,83%. Esses resultados indicam que, embora o modelo tenha sido capaz de identificar alguns padrões nos dados, sua capacidade de classificar corretamente as diferentes classes de faltas foi limitada. A Regressão Logística mostra-se mais adequada para problemas de classificação binária ou com classes bem definidas, enfrentando, no entanto, dificuldades significativas em cenários complexos com múltiplas classes. A simplicidade e rapidez de treinamento são vantagens intrínsecas deste modelo, mas não compensam a baixa precisão observada em contextos mais complexos.

A Rede Neural Artificial (RNA), apresentaram uma acurácia de 95,34%, precisão de 95,37%, recall de 95,61% e F1-Score de 95,43%. Esses resultados demonstram a eficácia da RNA em capturar as complexidades intrínsecas aos dados. No entanto, o elevado custo computacional e o prolongado tempo de treinamento constituem desvantagens práticas. Assim, mesmo com sua alta precisão, a RNA pode não ser a escolha mais eficiente para aplicações onde tempo e recursos são limitados.

O Knn, apresentou uma acurácia de 81,03%, precisão de 82,40%, recall de 81,03% e F1-Score de 81,39%. Este modelo, embora intuitivo e de fácil compreensão, mostrou-se eficaz na captura de padrões nos dados sem a necessidade de um treinamento extensivo. No entanto, o kNN pode se tornar computacionalmente desvantajoso com grandes volumes de dados, devido ao cálculo intensivo de distâncias durante a fase de predição. O kNN é sensível à escolha do número de vizinhos, o que pode afetar sua aplicabilidade em cenários de grande escala ou quando a velocidade de predição é crucial.

O modelo Random Forest destacou-se significativamente, apresentando uma acurácia de 92,67%, assim como métricas de precisão, recall e F1-score igualmente altas. O Random Forest demonstrou uma capacidade excepcional de lidar com dados desequilibrados e uma menor propensão ao sobreajuste, graças à combinação de múltiplas árvores de decisão. Além disso, o tempo de execução foi menor do que o requerido pela RNA, tornando-o um modelo equilibrado entre desempenho e eficiência computacional. A combinação de alta precisão, robustez e eficiência torna o Random Forest uma excelente escolha para a tarefa em questão.

O SVM apresentou resultados modestos, com acurácia de 78,71%, precisão de 79,34%, recall de 78,71% e F1-Score de 77,65%. Essas métricas indicam que o SVM teve dificuldades em capturar os padrões necessários para classificar corretamente as diferentes classes de faltas. Além disso, o tempo de execução foi substancial, com mais de 3 horas para completar o treinamento, o que torna o modelo impraticável sem ajustes adicionais ou recursos computacionais substanciais.

A comparação dos resultados revela que, embora a RNA tenha apresentado as melhores métricas de desempenho, o modelo Random Forest teve um tempo de treinamento significativamente mais rápido. As métricas de desempenho do Random Forest foram ligeiramente mais baixas que as da RNA, mas ainda assim muito altas, indicando uma precisão elevada na classificação das diferentes classes de faltas. A capacidade do Random Forest de lidar com dados desequilibrados e sua menor propensão ao sobreajuste tornam-no uma escolha robusta e confiável para esta tarefa.

Portanto, considerando todas as métricas de desempenho, robustez e eficiência computacional, o modelo RNA destaca-se como a melhor escolha em situações em que o retreinamento frequente dos modelos não é necessário. No entanto, se houver a necessidade de retreinamento constante, o tempo de treinamento mais rápido do Random Forest o torna mais vantajoso em comparação com a RNA, que possui um custo computacional e tempo de treinamento mais elevados. Dessa forma, a escolha do modelo ideal deve levar em consideração tanto as métricas de desempenho quanto as necessidades operacionais e de recursos do sistema.

6 CONCLUSÕES E PROPOSTAS DE CONTINUIDADE

Neste trabalho, investigou-se a aplicação de modelos de *Machine Learning* para a classificação de faltas em linhas de transmissão, abrangendo as técnicas de Regressão Logística, Rede Neural Artificial (RNA), *K-Nearest Neighbors* (kNN), *Random Forest* e *Support Vector Machine* (SVM). A análise comparativa dos modelos foi realizada com base em métricas de desempenho, tais como acurácia, precisão, recall e F1-Score.

6.1 Conclusões

Os resultados evidenciaram que o modelo RNA apresentou o melhor desempenho, destacando-se pela sua alta precisão. No entanto, exige mais da eficiência computacional. Embora a RNA tenha apresentado um desempenho excepcional, teve um elevado custo computacional e um tempo prolongado de treinamento, tornando-se ideal para casos em que o retreinamento do modelo não é necessário. Nestas situações, a RNA pode aproveitar ao máximo sua capacidade de capturar complexidades nos dados sem a preocupação com a necessidade constante de reprocessamento. O modelo de Regressão Logística apresentou limitações significativas em cenários complexos com múltiplas classes.

O kNN mostrou-se eficaz na captura de padrões nos dados, porém, pode tornar-se computacionalmente desvantajoso em grandes volumes de dados, com um tempo de treinamento relativamente longo. O modelo Random Forest apresentou um equilíbrio ideal entre precisão e eficiência, além de um tempo de treinamento razoável, tornando-o uma escolha mais prática em cenários onde retreinamento é necessário. O modelo SVM, com os parâmetros atuais, teve um desempenho mais modesto e um tempo de treinamento excessivo.

Uma das principais dificuldades encontradas durante este estudo foi a limitação de hardware e processamento, que restringiu a quantidade de dados utilizada para treinar e testar os modelos. A insuficiência de recursos computacionais impediu a exploração de conjuntos de dados ainda maiores e mais representativos, o que poderia potencialmente melhorar a generalização e a precisão dos modelos. Esta

limitação também afetou a capacidade de realizar experimentos mais extensivos e detalhados, restringindo o escopo do estudo.

6.2 Propostas de Continuidade

Para a continuidade deste trabalho, propõem-se diversas direções a serem exploradas em futuras pesquisas. Uma dessas direções é a implementação de modelos que, além de classificar, também sejam capazes de detectar a ocorrência de falhas. A detecção de falhas constitui um passo crucial para a manutenção preventiva e a resposta rápida a problemas nas linhas de transmissão. Modelos que integrem a classificação e a detecção de falhas podem fornecer uma solução mais completa e eficaz.

Outra proposta é a aplicação de técnicas de *Machine Learning* para a localização de faltas em linhas de transmissão. A localização precisa de falhas é essencial para minimizar o tempo de inatividade e os custos associados à manutenção e reparo. Métodos avançados, como redes neurais profundas e técnicas de aprendizado por reforço, podem ser investigados para aprimorar a precisão na localização de faltas.

Adicionalmente, futuros estudos podem realizar comparações mais extensivas entre diferentes modelos de *Machine Learning*, incluindo abordagens híbridas, que combinam múltiplos algoritmos para melhorar o desempenho global. A utilização de recursos computacionais mais avançados, como clusters de computação de alto desempenho ou serviços de computação em nuvem, pode possibilitar a análise de conjuntos de dados maiores e mais complexos, proporcionando resultados mais robustos e generalizáveis.

Em síntese, este trabalho estabeleceu as bases para a aplicação de *Machine Learning* na classificação de faltas em linhas de transmissão, destacando os desafios enfrentados e as oportunidades futuras. A superação das limitações de hardware e processamento, bem como a expansão do escopo das técnicas investigadas, podem conduzir a avanços significativos na manutenção e operação de sistemas de transmissão de energia elétrica.

REFERÊNCIAS

1. MITCHELL, M. T. **Machine Learning**. [S.l.]: [s.n.], 1997.
2. BISHOP, C. M. [S.l.]: [s.n.], 2006.
3. HASTIE, T. **The Elements of Statistical Learning**. [S.l.]: [s.n.], 2008.
4. NG, A. **Machine Learning Yearning**. [S.l.]: [s.n.], 2018.
5. SUTTON, R. S. **Reinforcement Learning:** An Introduction. [S.l.]: [s.n.], 2018.
6. BARBOSA, D. C. P. Machine Learning Approach to Detect Faults in Anchor Rods of Power Transmission Lines. **IEEE**, 2019.
7. DREISEITLA, S. Logistic regression and artificial neural network classification, 2003.
8. HART, T. C. &. P. **Nearest neighbor pattern classification**. [S.l.]: IEEE Transactions on Information Theory, 1967.
9. COSTA, B. G. D. **CLASSIFICAÇÃO DE FALTAS DO TIPO CURTO CIRCUITO EM LINHAS DE TRANSMISSÃO UTILIZANDO KNN-DTW**. Pará. 2017.
10. BREIMAN, L. **"Random Forests". Machine Learning**. [S.l.]: [s.n.], 2001.
11. PATEL, R. K. **Feature selection and classification of mechanical fault of an induction motor**. [S.l.]: [s.n.], 2016.
12. VAPNIK, V. **The Nature of Statistical Learning Theory. Springer-Verlag.** [S.l.]: [s.n.], 1995.
13. EKICI, S. Support Vector Machines for classification and locating faults on transmission lines, 2012.
14. HAYKIN, S. **Neural Networks:** A Comprehensive Foundation. Prentice Hall. [S.l.]: [s.n.], 1998.
15. NIELSEN, M. **Neural Networks and Deep Learning**. [S.l.]: [s.n.], 1993.
16. YADAV, A. An Overview of Transmission Line Protection by Artificial Neural Network: Fault Detection, Fault Classification, Fault, 2014.
17. ANDERSON, P. M. **Analysis of Faulted Power Systems**. [S.l.]: IEEE, 1973.
18. FERREIRA, V. H. A survey on intelligent system application to fault diagnosis in electric, Niterói, 2016.
19. FILHO, J. M. **Proteção de Sistemas Elétricos de Potência**. [S.l.]: [s.n.], 2011.
20. KINDERMANN, G. **Curto-Circuito**. [S.l.]: [s.n.], 1997.
21. ANDERSON, P. M. **Power System Protection**. [S.l.]: [s.n.], 1998.
22. ENSINA, L. A. Localização de Faltas em Linhas de Transmissão de Energia Elétrica Utilizando Redes Neurais Recorrentes LSTM e GRU. **IEEE**, p. 7, 2022.
23. BARNETT, V.; LEWIS, T. **Outliers in Statiscal Data**. [S.l.]: [s.n.], 1994.

24. BISHOP, C. M. **Pattern Recognition and Machine Learning**. [S.l.]: [s.n.], 2006.

25. CHAWLA, N. V. **SMOTE:** Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*. [S.l.]: [s.n.], 2002.

26. GIRON, A. **Aprendizado de Máquina com Scikit-learn & Tensorflow**. Paris: [s.n.], 2017.

27. RUSSEL, S.; NORVIG, P. **Artificial Intelligence:** A Modern Approach. 4th edition. ed. Berkeley: Pearson, 2020.

28. CALDERON, J. ; MADRIGAL, G. Z.; CARRANZA, D. A. O. Comparative Analysis between Models of Neural Networks for the Classification of Faults in Electrical Systems, Medellin, 2007.

29. HAYKIN, S. **Redes Neurais:** Princípios e Prática. 2ª edição. ed. Darmstadt, Alemanha: Bookman, 2000.

30. MULLER, A.; GUIDO, S. **Introduction to Machine Learning with Python:** A Guide for Data Scientist. 1ª. ed. Bonn: O'Reilly Media, 2016.

31. K. M. SILVA, B. A. S. N. S. D. B. Fault Detection and Classification in Transmission Lines Based on Wavelet Transform and ANN. **IEEE TRANSACTIONS ON POWER DELIVERY**, 4 OCTOBER 2006. 6.

32. GARETH JAMES, D. W. T. H. R. T. **An Introduction to Statistical Learning**. [S.l.]: [s.n.], 2013.

33. [S.l.]: [s.n.].

34. ZANETTA JR., L. C. **A aprendizagem de rede neural**. [S.l.]: [s.n.], 2005.

35. BARTO, R. S. S. E. A. G. **Reinforcement Learning:** An Introduction. [S.l.]: [s.n.], 1998.

APÊNDICES

APÊNDICE 1: REPOSITÓRIO GITHUB

Para acessar todos os detalhes e códigos utilizados neste projeto, foi disponibilizado um repositório no GitHub que contém a base de dados usada, bem como os notebooks de implementação e avaliação dos cinco modelos de classificação. A página pode ser acessada pelo seguinte link: https://github.com/laurafernogueira/AnalysisFault_MachineLearning

No repositório, contém:

1. Base de Dados: Todos os arquivos de dados utilizados no projeto estão disponíveis, organizados de forma a facilitar a replicação dos experimentos. Esses dados incluem os valores de tensão e corrente registrados em diferentes condições de faltas nas linhas de transmissão.

2. Notebooks dos Modelos de Classificação:

Random Forest: Contém a implementação do algoritmo Random Forest, incluindo o pré-processamento dos dados, treinamento do modelo, geração da matriz de confusão, cálculo das métricas de desempenho e visualização dos resultados através de histogramas.

Regressão Logística: Inclui todo o código necessário para treinar e avaliar um modelo de Regressão Logística, com as mesmas etapas de pré-processamento, avaliação e visualização mencionadas acima.

SVM (Support Vector Machine): Notebook detalhando a configuração e treinamento de um modelo SVM, além das visualizações e métricas de desempenho.

Redes Neurais Artificiais (RNA): Apresenta a implementação de uma Rede Neural Artificial, cobrindo desde a configuração inicial até a avaliação do modelo, incluindo as visualizações necessárias para análise de desempenho.

kNN (k-Nearest Neighbors): Contém a implementação do modelo kNN, com todas as etapas de preparação dos dados, treinamento, avaliação e visualização dos resultados.

Esses notebooks fornecem uma visão completa e detalhada de cada etapa do processo de classificação de faltas, permitindo que outros pesquisadores e profissionais da área possam replicar os experimentos, validar os resultados ou adaptar os métodos para suas necessidades específicas. O arquivo README contém instruções de como utilizar os algoritmos.