PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Algoritmo Genético para Seleção de Variáveis na Classificação de Falhas em Processos Químicos de Larga Escala

Marcos Vinícius Porto de Sá

PROJETO FINAL DE GRADUAÇÃO

CENTRO TÉCNICO CIENTÍFICO - CTC

DEPARTAMENTO DE INFORMÁTICA

Curso de Graduação em Engenharia da Computação

Rio de Janeiro, 30 de junho de 2022

Marcos Vinícius Porto de Sá

Algoritmo Genético para Seleção de Variáveis na Classificação de Falhas em Processos Químicos de Larga Escala

Relatório de Projeto Final, apresentado ao programa **Projeto de Graduação em Engenharia de Computação II** da PUC-Rio como requisito parcial para a obtenção do título de Engenheiro de Computação.

Orientador: Marley Vellasco

Rio de Janeiro

junho de 2022.

Resumo

De Sá, Marcos Vinícius Porto. Vellasco, Marley. Algoritmo Genético para Seleção de Variáveis na Classificação de Falhas em Processos Químicos de Larga Escala. Rio de Janeiro, 2022. 34 p. Relatório Final de Projeto de Conclusão de Curso de Engenharia da Computação – Centro Técnico Científico – CTC, Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

Com o advento da indústria 4.0, observou-se um aumento expressivo na geração de informações alavancado por novas tecnologias de monitoramento. Algoritmos de *Machine Learning* são amplamente empregados nesse contexto para a inferência estatística, predição, detecção e diagnóstico de falhas. Entretanto, dados redundantes ou com baixo ganho de informação a respeito do processo que se deseja ter uma predição ou diagnóstico podem significar um custo computacional desnecessário, gerando modelos inadequados. Um grande desafio, portanto, é filtrar esses dados a fim de capturar apenas uma parcela que possua relevância significativa, com o objetivo de otimizar recursos investidos nos sistemas de monitoramento. Este projeto propõe um método de seleção de variáveis do tipo *Wrapper* baseado em Algoritmos Genéticos para obter um subconjunto suficiente de atributos de entrada para proporcionar uma acurácia satisfatória no treinamento e validação de modelos de classificação de falhas em processos químicos industriais. Foi empregado para a classificação de falhas e também para avaliar as soluções geradas pelo Algoritmos Genético, o *Random Forrest,* da classe dos algoritmos do tipo *Ensemble*. Este projeto utilizou como objeto de estudo o caso do Processo *Tennessee Eastman*. Os resultados foram considerados promissores, obtendo-se uma melhora na acurácia de 1,72% e com uma redução de aproximadamente 50% do número de variáveis em relação ao caso base sem seleção de variáveis.

Palavras-chave

Seleção de Variáveis, *Wrapper*, Algoritmos Genéticos, Classificação, Processo *Tennessee Eastman*

Abstract

De Sá, Marcos Vinícius Porto. Vellasco, Marley. Genetic Algorithm for Feature Selection in Large Scale Chemical Processes Fault Classification. Rio de Janeiro, 2022. 34 p. Final Report for Computer Engineering Course Conclusion – Centro Técnico Científico – CTC, Department of Informatics. Pontifical Catholic University of Rio de Janeiro.

With the advent of industry 4.0, there was a significant increase in the generation of information leveraged by new monitoring technologies. Machine Learning algorithms are widely used in this context for statistical inference, prediction, detection and fault diagnosis. However, redundant data or with low gain of information about the process that is desired to have a prediction or diagnosis, can mean an unnecessary cost, generating slow models. A major challenge, therefore, is to filter this data to capture only a portion that has significant relevance, to optimize resources invested in monitoring systems. This project proposes a Wrapper Feature Selection Method based on a Genetic Algorithm to obtain a subset of the input attributes to provide a satisfactory accuracy in the training and validation of classification models in Industrial Chemical Processes Fault Classification. It was used for Fault Classification and to evaluate the solutions generated by the Genetic Algorithm, a Random Forrest Classifier, from the Ensemble algorithm class. This project used the Tennessee Eastman Process as its object of study. The results were considered promising, with an improvement in accuracy of 1.72% and a reduction of approximately 50% in the number of variables in relation to the base case without selection of variables.

Keywords

Feature Selection, *Wrapper*, Genetic Algorithm, Classification, *Tennessee Eastman Process*

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

1.1 Apresentação

Em processos químicos industriais, diversos tipos de falhas podem afetar a eficiência do processo, comprometer a produtividade e a segurança. (CACCAVALE et al, 2009) o que torna essencial a detecção e o diagnóstico de falhas em sistemas de larga escala nesse contexto. Além disso, com o advento da indústria 4.0, observou-se um aumento expressivo na geração de informações alavancado por novas tecnologias de monitoramento, o que dá maior importância à eficiente utilização desses dados para o diagnóstico de falhas. (XIAO et al, 2021).

A geração de dados gerados pode alcançar *terabytes*, ou até *petabytes*, portanto, é um desafio não só gerenciar esses dados, mas também interpretá-los e extrair informações relevantes (YIN et al, 2015). Para se beneficiar dessa grande quantidade de informação Algoritmos de *Machine Learning* são amplamente empregados nesse contexto para a inferência estatística, predição, detecção e diagnóstico de falhas. (SOARES, 2017)

Num processo de aprendizado de máquina, estratégia comumente usada na detecção de falhas é: quanto maior o volume de dados de entrada utilizados para o treinamento do modelo, maior tende a ser a sua acurácia, desde que os dados fornecidos sejam fidedignos ao processo que se deseja classificar (ROH et al., 2021), (ADADI, 2021). Entretanto, informações redundantes ou sem relevância podem resultar em um desperdício de tempo de treino, além do desperdício dos recursos despendidos para coletar e processar esses dados (WANG et al., 2017). Para evitar esse desperdício é bastante interessante que se consiga manter uma taxa de acertos satisfatória no modelo utilizando somente parte dos dados de entrada e com pouca - ou nenhuma - perda de informação sobre o processo que se deseja classificar.

1.2 Motivação

Para resolver o problema de otimizar os dados de entrada na classificação de falhas há uma vasta gama de métodos de seleção de variáveis utilizados na fase de pré-processamento dos dados, como visto em CHEBEL-MORELLO et al. (2011), que propõe a avaliação da correlação dos atributos de entrada com as classes para seleção destes atributos na classificação de falhas no *Processo*

Tennessee Eastman. Já em HUI et al. (2017), os autores obtiveram um aumento do desempenho no diagnóstico de falhas em rolamentos, reduzindo as informações consumidas a respeito das vibrações.

A seleção de variáveis, ou também conhecido como seleção de atributos, é o processo que visa selecionar um subconjunto das variáveis originais baseados em algum critério, sem reduzir a capacidade de previsão ou inferência do modelo. Este é um importante passo de pré-processamento e tem um papel fundamental na construção de modelos de detecção mais acurados (AHUJA e RATNOO, 2015). As metodologias de seleção de variáveis utilizadas atualmente podem ser classificadas como *wrapper-based*, *filter-based* e *embedded-based* (PRAMOCKCHON e PIAMSA-NGA, 2016),

Os métodos de seleção de variáveis *filter-based* não dependem do algoritmo de aprendizado de máquinas empregado (DU et al., 2019) e são comumente aplicados como uma etapa de pré-processamento, onde as variáveis analisadas são classificadas por relevância de acordo com propriedades intrínsecas aos dados. (CLAVIJO et al., 2021)

Os métodos *wrapper-based* utilizam o desempenho do classificador durante a busca pelo subconjunto ideal dos atributos (ZORIC et al, 2020). Um dos métodos básicos é a seleção *forward* ou *backward*: um método sequencial, iniciando a seleção de variáveis a partir de um conjunto vazio (ou cheio), e acrescenta variáveis (ou remove variáveis) avaliando o desempenho do modelo de Aprendizado de Máquina utilizado no processo de classificação ou regressão (WAH et al., 2018). Outro método básico também explorado é o método heurístico, quando os subconjuntos são gerados com a ajuda de algoritmos estocásticos e o desempenho do modelo classificação ou regressão é avaliado. (CLAVIJO et al., 2021)

Os métodos *embedded-based* combinam o modelo de aprendizado com um problema de otimização, permitindo que a seleção de variáveis seja feita simultaneamente com o ajuste do modelo de detecção (CHEN et al., 2019).

Os métodos *filter-based* necessitam de menor poder computacional, dada a não dependência do ajuste do modelo de aprendizado de máquina no processo de seleção, entretanto, em alguns casos o resultado dessa seleção não maximiza a performance do modelo de classificação e detecção (ZHANG et al., 2019). Por outro lado, os modelos *wrapper-based* e *embedded-based* possuem melhor desempenho geral, porém necessitam de mais tempo de processamento. (WANG et al., 2015)

Para encontrar uma opção viável para este projeto é preciso considerar que existem dois fatores que dificultam a descoberta do subconjunto ideal de atributos: 1) a inviabilidade de se testar todas as combinações possíveis de atributos a serem utilizados, principalmente em conjuntos de dados com uma grande quantidade de atributos, devido ao custo computacional impeditivo, resultado da busca exaustiva. 2) a complexidade de avaliar tecnicamente o subconjunto que ao mesmo tempo excluísse atributos irrelevantes ou redundantes e mantivesse atributos que oferecessem um ganho de informação expressivo. Uma abordagem possível para cenários, onde se têm pouco ou nenhum conhecimento sobre a natureza dos processos geradores dos dados de entrada, é a utilização da metodologia *wrapper-based* de seleção de variáveis compostas por Algoritmo Genético assim como em JACK e NANDI (2000), ALI et al. (2019), CERRADA et al. (2015) e em OLULEYE et al. (2014), visando identificar os atributos que melhoram a capacidade de detecção de falhas em sistemas de monitoramento industriais de larga escala. Portanto, uma solução equilibrada, tanto em termos de acurácia dos resultados obtidos com a seleção das variáveis, quanto de custo de processamento, é a implementação de um método wrapper-based de seleção de variáveis heurístico utilizando Algoritmo Genético.

Há diversos estudos sobre o tema como visto em CHIESA et al. (2020), GARCÍA-DOMINGUEZ et al. (2020) e WULTZ et al. (2019), entretanto, até onde se observou, não foi identificado especificamente a utilização de Algoritmos Genético em métodos de seleção de variáveis aplicado à detecção de falhas em sistemas de monitoramento de processos industriais em larga escala.

1.3 Objetivo

Com base no estudo da abordagem citada, este projeto tem como objetivo implementar um modelo baseado em algoritmo genético para selecionar um subconjunto suficiente de variáveis de controle, presentes na base de dados do benchmark *Tennessee Eastman*. Não será objetivo desta seleção o de maximizar a acurácia na detecção de falhas, mas achar um conjunto de atributos de tal modo a estabelecer um equilíbrio entre acurácia satisfatória e redução do número de variáveis a serem utilizadas pelo algoritmo de detecção e classificação de falhas. O modelo desenvolvido terá como entrada uma base de dados contendo todas as variáveis geradas por simulação de um processo químico de larga escala, baseado o modelo *Tennessee Eastman.* O produto resultante é o subconjunto de variáveis que obteve o melhor desempenho, tendo como função objetivo a

acurácia e a redução no número de variáveis, considerando como modelo de classificação de falhas o algoritmo *Random Forest Classifier,* englobando todas as possíveis falhas a serem detectadas. Como este método de seleção de variáveis depende diretamente do modelo de classificação empregado, deve-se incluí-lo na categoria *wrapper-based*. Cabe ressaltar que os resultados obtidos neste projeto serão utilizados no trabalho de pesquisa conjunta de professores de engenharia elétrica e química denominado *Gerenciamento das Situações Anormais: Detecção e Diagnóstico de Falhas na Produção de Petróleo e Gás Natural Baseado em Machine Learning.*

1.4 Descrição do Trabalho

O restante deste documento é organizado da seguinte maneira: O capítulo 2 apresenta a fundamentação teórica a respeito dos conceitos abordados neste trabalho. A seção 2.1 detalha o que é o processo Tennessee Eastman e como o mesmo oferece um benchmark para simulação de monitoramento de falhas em processos químicos industriais. A seção 2.2 descreve o conceito de Algoritmo Genético e apresenta a biblioteca que será utilizada para facilitar a sua implementação. A seção 2.3 resume brevemente o que são Árvores de Decisão e como os algoritmos de classificação *ensemble* às utiliza nas chamadas Florestas Aleatórias (*Random Forest*). No capítulo 3 são listadas as etapas do processo de implementação do Algoritmo Genético para Seleção de Variáveis Wrapper, objetivo deste trabalho. Os resultados dos testes são apresentados e comentados no capítulo 4. O capítulo 6 contém as conclusões obtidas a partir dos resultados dos testes, bem como a proposição de trabalhos futuros pertinentes ao tema deste trabalho.

2 Fundamentação Teórica

2.1 Processo Tennessee Eastman (TEP)

Para fornecer um método realista de estudar o monitoramento de processos industriais em larga escala, um simulador de uma planta de produção foi proposto em DOWNS e VOGEL (1993). TEP é um modelo de processo industrial baseado em um processo químico, que consiste em cinco unidades: um reator de duas fases, onde uma reação exotérmica ocorre, um separador, um stripper, um compressor e um misturador. Este é um processo instável, aberto e não-linear que

tem sido usado como objeto de estudo em diversas pesquisas para simular controle em larga escala, monitoramento estatístico de processos, detecção de falhas e modelos de identificação orientados a dados (MARTIN-VILLALBA et al., 2018).

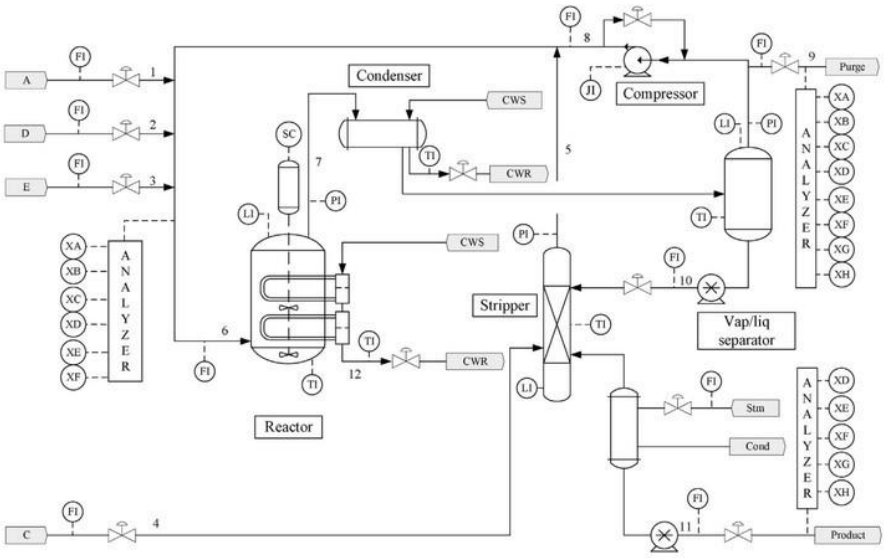

Figura 1 - Processo Tennessee Eastman

O simulador utiliza 12 variáveis manipuladas (XMV) e 41 variáveis medidas de processo (XMEAS), totalizando 53 variáveis. As variáveis manipuladas se encontram na Tabela 1. A Tabela 2 apresenta as medições contínuas do processo (como temperaturas, pressões, vazão etc.), enquanto as Tabelas 3 e 4 trazem as medidas de composição dos componentes.

Tabela 1 - Variáveis manipuladas do processo

Identificação	Descrição da variável	Unidade
XMV (1)	Vazão de reagente D (fluxo 2)	kg/h
XMV (2)	Vazão do subproduto E (fluxo 3)	kg/h
XMV (3)	Vazão de reagente A (fluxo 1)	kscmh
XMV (4)	Vazão dos reagentes A e C (fluxo 4)	kscmh
XMV (5)	Abertura da Válvula do Compressor	%
XMV (6)	Abertura da Válvula de purga (fluxo 9)	%
XMV (7)	Vazão de líquido para o separador (fluxo 10)	m^3/h
XMV (8)	Vazão de produto do separador (fluxo 11)	m^3/h
XMV (9)	Abertura da Válvula de vapor do stripper	%
XMV (10)	Vazão de refrigerante do reator	m^3/h
XMV (12)	Vazão de refrigerante do condensador	m^3/h
XMV (11)	Velocidade do agitador	rpm

Fonte: Adaptado de Downs e Vogel (1993)

Tabela 2 - Variáveis medidas continuamente do processo

Identificação	Descrição da variável	Unidade
XMEAS (1)	Vazão de reagente A (fluxo 1)	kscmh
XMEAS (2)	Vazão do regente D (fluxo 2)	kg/h
XMEAS (3)	Vazão do subproduto (fluxo 3)	kg/h
XMEAS (4)	Vazão dos reagentes A e C (fluxo 4)	kscmh
XMEAS (5)	Vazão de reciclagem (fluxo 8)	kscmh
XMEAS (6)	Alimentação do reator (fluxo 6)	kscmh
XMEAS (7)	Pressão do reator	kPa
XMEAS (8)	Nível do reator	%
XMEAS (9)	Temperatura no reator	^{O}C
XMEAS (10)	Taxa de purga (fluxo 9)	kscmh
XMEAS (11)	Temperatura no separador	^{O}C
XMEAS (12)	Nível do separador	%
XMEAS (13)	Pressão no separador	kPa
XMEAS (14)	Fluxo do separador (fluxo 10)	m^3/h
XMEAS (15)	Nível do stripper	%
XMEAS (16)	Pressão do stripper	kPa
XMEAS (17)	Fluxo do stripper (fluxo 11)	m^3/h
XMEAS (18)	Temperatura do stripper	^{O}C
XMEAS (19)	Vazão de gás do stripper	kg/h
XMEAS (20)	Trabalho do compressor	kw
XMEAS (21)	Temperatura de saída do refrigerante do reator	^{O}C
XMEAS (22)	Temperatura de saída do refrigerante do separador	^{O}C

Fonte: Adaptado de Downs e Vogel (1993)

Tabela 3 - Variáveis medidas no fluxo de alimentação

Identificação	Descrição da variável	Unidade
XMEAS (23)	Concentração de A	% molar
XMEAS (24)	Concentração de B	% molar
XMEAS (25)	Concentração de C	% molar
XMEAS (26)	Concentração de D	% molar
XMEAS (27)	Concentração de E	% molar
XMEAS (28)	Concentração de F	% molar

Fonte: Adaptado de Downs e Vogel (1993)

Tabela 4 - Variáveis medidas no gás de purga

Identificação	Descrição da variável	Unidade
XMEAS (29)	Concentração de A	% molar
XMEAS (30)	Concentração de B	% molar
XMEAS (31)	Concentração de C	% molar
XMEAS (32)	Concentração de D	% molar
XMEAS (33)	Concentração de E	% molar
XMEAS (34)	Concentração de F	% molar
XMEAS (35)	Concentração de G	% molar
XMEAS (36)	Concentração de H	% molar

Fonte: Adaptado de Downs e Vogel (1993)

Tabela 5 - Variáveis medidas no fluxo de produto

Identificação	Descrição da variável	Unidade
XMEAS (37)	Concentração de D	% molar
XMEAS (38)	Concentração de E	% molar
XMEAS (39)	Concentração de F	% molar
XMEAS (40)	Concentração de G	% molar
XMEAS (41)	Concentração de H	% molar

Fonte: Adaptado de Downs e Vogel (1993)

As 20 falhas definidas por Downs e Vogel (1993) estão listadas na Tabela 6.

Tabela 6 - Falhas do processo

Identificação	Descrição da variável	Tipo
IDV (1)	Razão de alimentação A/C (fluxo 4)	Degrau
IDV (2)	Composição de B (fluxo 4)	Degrau
IDV (3)	Temperatura de alimentação (fluxo 2)	Degrau
IDV (4)	Temperatura de entrada do refrigerante do reator	Degrau
IDV (5)	Temperatura de entrado do refrigerante no condensador	Degrau
IDV (6)	Perda de alimentação de A (fluxo 1)	Degrau
IDV (7)	Redução na disponibilidade de C (fluxo 4)	Degrau
IDV (8)	Composição de alimentação de A, B e C (fluxo 4)	Variação Randômica
IDV (9)	Temperatura de alimentação de D (fluxo 2)	Variação Randômica
IDV (10)	Temperatura de alimentação de C (fluxo 4)	Variação Randômica
IDV (11)	Temperatura de entrada do refrigerante do reator	Variação Randômica
IDV (12)	Temperatura de entrado do refrigerante no condensador	Variação Randômica
IDV (13)	Cinética das reações	Desvio lento
IDV (14)	Válvula do refrigerante do reator	Agarramento
IDV (15)	Válvula do refrigerante do condensador	Agarramento
IDV (16)	Desconhecido	
IDV (17)	Desconhecido	
IDV (18)	Desconhecido	
IDV (19)	Desconhecido	
IDV (20)	Desconhecido	

Fonte: Adaptado de Downs e Vogel (1993)

Maiores detalhes podem ser obtidos em RUSSEL et al. (2000).

2.2 Algoritmo Genético

2.2.1 Base Biológica e Aplicações

Algoritmos Evolucionários são baseados no modelo de seleção natural proposto por Charles Darwin (1859). A teoria da evolução de Darwin explica a evolução das espécies pelo princípio de seleção natural, que favorece a sobrevivência das espécies que melhor se adaptam às condições ambientais às quais são expostas. (BÄCK, 1995). Algoritmos Genéticos é uma variante dos Algoritmos Evolucionários, introduzidos por Holland (1975) e popularizados por Goldberg (1989), têm sucesso na solução de problemas de otimização e configuração, onde o espaço de busca é muito grande (BAJPAI e KUMAR, 2010) ou quando o problema não pode ser matematicamente descrito (SLOSS, GUSTAFSON, 2019). A seção a seguir descreve brevemente o funcionamento de um Algoritmo Genético.

2.2.2 Algoritmo Genético Clássico

Inicialmente se tem as possíveis soluções (indivíduos) para o problema, geradas inicialmente de forma aleatória, constituindo uma população. Os indivíduos, também chamados de cromossomos, que melhor se adaptarem ao problema serão as mais bem avaliadas numa função fitness. Esta função avalia os indivíduos com base em parâmetros que aumentam sua pontuação, quanto melhor for seu desempenho na resolução do problema proposto (KATOCH et al., 2020). As melhores soluções têm maior potencial em serem escolhidas para a reprodução (ou crossover), onde se combina características de cada solução a fim de encontrar uma solução nova, que pode ser melhor ou pior que as soluções selecionadas para reprodução (SPEARS e ANAND, 1991). Após o crossover pode-se aplicar uma mutação nessas soluções, que serão pequenas alterações aleatórias também visando uma possível melhor adaptação desse novo indivíduo como solução para o problema (GABRIEL e DELBERN, 2008). O processo é repetido gerando novas gerações até que se atinja o nível de adaptabilidade desejado ou até que o número de gerações atinja um limite. A Figura 2 ilustra o fluxograma geral de um algoritmo genético.

Figura 2 - Fluxograma de um Algoritmo Genético Clássico

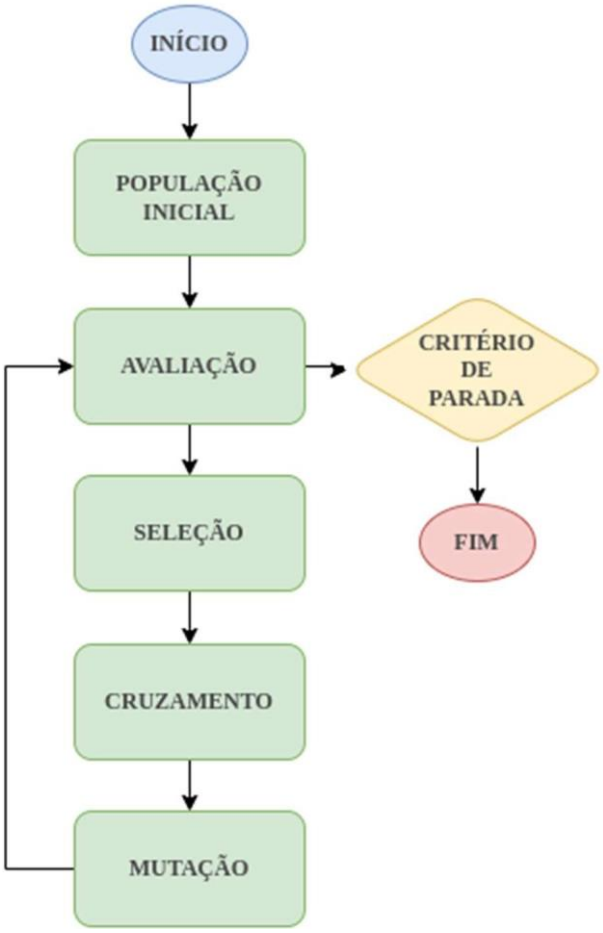

Figura 2 - Fluxograma de um Algoritmo Genético Clássico

Tanto o crossover quanto a mutação ocorrem dependendo de uma probabilidade específica, utilizando métodos que podem variar segundo a conveniência do problema (MICHALEWICZ e SCHOENAUER, 1996). Há algumas alternativas de métodos de crossover, mutação, seleção de indivíduos para o cruzamento, além de parâmetros (taxa de crossover, mutação etc.) mais adequados que devem ser investigados experimentalmente para se identificar a as melhores soluções representadas como indivíduos. (HILDAYANTI et al., 2018)

2.2.3 Biblioteca DEAP

Um dos grandes benefícios da utilização da linguagem Python para a implementação de soluções em *Data Science* e *Machine Learning* é a vasta gama de bibliotecas que apoiam no processo de manipulação dos dados e treinamento de modelos. Por consequência surgem, com cada vez maior frequência,

bibliotecas que apoiam os desafios dentro desse universo, inclusive o de implementar um modelo baseado em Algoritmos Genético. A biblioteca DEAP (FORTIN et al., 2012) oferece uma solução completa para a implementação de Algoritmos Evolucionários, entre eles o Algoritmo Genético, com um alto nível de abstração.

O paradigma da biblioteca DEAP é baseada em uma caixa de ferramentas, onde devem ser registradas as definições sobre as principais entidades pertencentes ao modelo. Por exemplo, na caixa de ferramenta deve ser definido se a otimização buscada pelo modelo é traduzida na maximização de um valor X, na minimização de outro valor Y ou na combinação entre maximização e minimização de vários valores X, Y, Z, etc. Nessa caixa de ferramentas deve ser registrado qual é a representação do indivíduo, qual é o método de cruzamento a ser utilizado, a probabilidade de mutação e assim em diante. A biblioteca ainda oferece relatórios para o acompanhamento das gerações, como a máxima pontuação de aptidão dos indivíduos em cada geração.

2.3 Arvores de Decisão e Florestas Aleatórias

Árvores de Decisão são métodos de classificação de dados no contexto da chamada Mineração de Dados (*Data Mining*). Com o uso de uma árvore de decisão, um problema complexo é dividido em problemas mais simples. (SANTI, 2018). O processo de indução de árvores de decisão tem a função de particionar recursivamente um conjunto de treinamento até que cada subconjunto obtido deste particionamento contenha casos de uma única classe. Uma árvore de decisão toma como entrada um objeto ou situação descrito por um conjunto de atributos e retorna uma decisão, ou seja, o valor de saída previsto, de acordo coma entrada. A árvore de classificação é o resultado de se fazer uma sequência ordenadas de perguntas, e as perguntas feitas a cada passo na sequência dependem das respostas às perguntas anteriores. A sequência termina com a previsão da classe. (BARBOSA et al., 2012). Essa sequência de perguntas e respostas é representada na forma de um grafo acíclico, conexo e dirigido em que cada nó pode ser definido como:

- Um nó folha que está associado a uma classe
- Um nó de decisão que contém um teste num atributo, cada ramo descendente corresponde a um possível valor deste atributo

(MICHEL, 1997)

Uma forma de entender como funcionam as árvores de decisão é por meio de exemplos gráficos. Supondo que em uma base de dados estejam contidas informações sobre o clima de uma determinada região e se tenha a necessidade de, com base nessas informações históricas inferir se uma determinada família foi para a praia ou se ficou em casa. O modelo baseado em árvores de decisão aprenderá a observar quando a família vai ou não para a praia com base em valores das variáveis do clima, como por exemplo se está um dia ensolarado ou se está ventando por exemplo, e criará uma espécie de fluxograma semelhante ao da Figura 3.

Figura 3 - Exemplo de Árvore de Decisão para exemplo de apoio à decisão de ir ou não para a praia a partir de condições climáticas

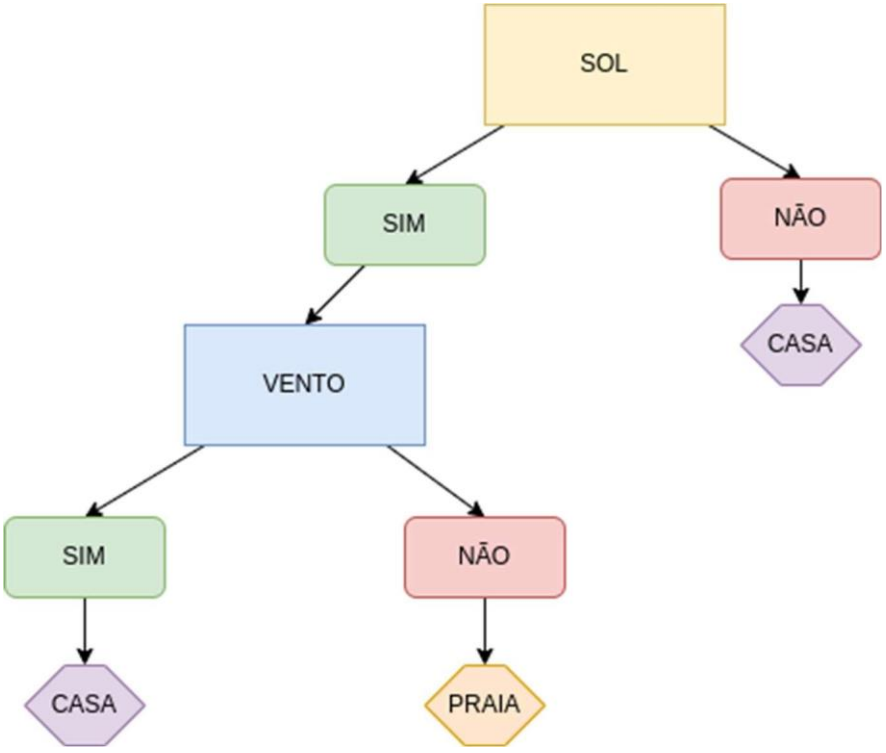

Figura 3 - Exemplo de Árvores de Decisão para exemplo de apoio à decisão de ir ou não para a praia a partir de condições climáticas

Em particular, as árvores quando muito profundas tendem a aprender padrões altamente irregulares: sobre-ajustam os seus conjuntos de treino, ou seja, têm um baixo viés, mas uma variância muito alta. (HASTIE et al., 2017) As florestas aleatórias são um conjunto de árvores de decisão, portanto um organizadas em “*ensemble*”, treinadas em diferentes partes do mesmo conjunto de treino, com o objetivo de reduzir a variância (SAGI e ROKACH, 2018). A modelagem ensemble é um processo em que se criam diversos modelos com o objetivo de ter uma predição ou classificação do mesmo processo. O modelo

ensemble então combina a predição ou classificação de todos os demais modelos treinados para obter um resultado final (SAINI e GOSH, 2017). A motivação do uso dessa abordagem de ensemble do tipo *bagging* é reduzir a variância e minimiza o *overfitting* (erro de generalização). Contanto que os modelos sejam independentes e diferentes, o erro de generalização é reduzido quando a estratégia de *ensemble* é utilizada. (BREIMAN, 2001)

3 Metodologia

Considerando que este trabalho tem o objetivo de apoiar um outro projeto de pesquisa ainda em desenvolvimento, algumas das decisões tomadas na metodologia foram baseadas na metodologia deste outro projeto. Desta forma os resultados obtidos poderão ser utilizados como referência em seu desenvolvimento.

De modo a estar alinhado com o que está sendo desenvolvido no projeto de pesquisa, que envolve esse estudo de caso, foram definidos: o processo de aquisição e análise dos dados, detalhada na seção 3.1, o pré-processamento dos dados, seção 3.2, bem como a estratégia escolhida para desenvolver o modelo de classificação, explorado na seção 3.3. O restante do capítulo, portanto, detalha aquilo que se propõe este trabalho.

3.1 Aquisição e Análise dos Dados

As bases de dados, geradas por meio de simulação, foram disponibilizadas pelos integrantes do projeto de pesquisa. Porém, como são objeto de estudo de diversos artigos na literatura são facilmente encontrados na internet. O site kaggle.com disponibiliza algumas bases de dados[1] para treino e para teste semelhantes as utilizadas neste projeto.

As bases de dados originais obtidas são armazenadas em arquivos com extensão *RData*, comumente utilizadas pela linguagem R. São duas bases de dados de treino e duas bases de dados de testes. Uma base de treino contém apenas registros onde não foi registrado qualquer tipo de falha e na outra base de treino estão contidos registros onde se observaram todas as possíveis falhas,

[1] https://www.kaggle.com/code/afrniomelo/tennessee-eastman-fault-detection-with-pca-and-lgb/data

sendo que em cada registro apenas uma falha foi detectada. O mesmo vale para as bases de teste.

- *TEP_FaultFree_Training.RData*
- *TEP_Faulty_Training.RData*
- *TEP_FaultFree_Testing.RData*
- *TEP_Faulty_Testing.RData*

As 20 falhas, descritas na seção 2.2 são representadas tanto nos registros usados para modelagem como para o teste.

- Nas bases de dados de treinamento (*TEP_FaultFree_Training.RData* e *TEP_Faulty_Training.RData*), para cada falha detectada existem 500 simulações com 500 amostras cada. Ou seja:
- para a base sem falha (*TEP_FaultFree_Training.RData)* existem 500 x 500 = 250.000 registros
- temos 20 possíveis falhas portanto:
    - para a base com falha (*TEP_Faulty_Training.RData*), existem 20 x 500 x 500 = 5.000.000 registros
- Nas bases de dados de teste (*TEP_FaultFree_Testing.RData e TEP_Faulty_Testing.RData*), para cada falha detectada existem 500 simulações com 960 amostras cada. Ou seja:
- para a base sem falha (*TEP_FaultFree_Testing.RData*) existem 500 x 960 = 480.000 registros
- temos 20 possíveis falhas portanto:
- para base com falha (*TEP_Faulty_Testing.RData*) existem 20 x 500 x 960 = 9.600.000 registros

Todos as bases possuem os mesmos atributos:

- Número da falha
    - No caso das bases sem falha o número será 0
    - Nas bases com falhas, os números vão de 1 a 20
- Número da Simulação (1 a 500)
- Número da Amostra
    - No caso das bases de treinamento de 1 a 500
    - Nas bases de teste de 1 a 960
- 53 variáveis

- 'xmeas_1', 'xmeas_2', 'xmeas_3', 'xmeas_4', 'xmeas_5', 'xmeas_6', 'xmeas_7', 'xmeas_8', 'xmeas_9', 'xmeas_10', 'xmeas_11', 'xmeas_12', 'xmeas_13', 'xmeas_14', 'xmeas_15', 'xmeas_16', 'xmeas_17', 'xmeas_18', 'xmeas_19', 'xmeas_20', 'xmeas_21', 'xmeas_22', 'xmeas_23', 'xmeas_24', 'xmeas_25', 'xmeas_26', 'xmeas_27', 'xmeas_28', 'xmeas_29', 'xmeas_30', 'xmeas_31', 'xmeas_32', 'xmeas_33', 'xmeas_34', 'xmeas_35', 'xmeas_36', 'xmeas_37', 'xmeas_38', 'xmeas_39', 'xmeas_40', 'xmeas_41',
- 'xmv_1','xmv_2', 'xmv_3', 'xmv_4', 'xmv_5', 'xmv_6', 'xmv_7', 'xmv_8', 'xmv_9', 'xmv_10', 'xmv_11', 'xmv_12']

As amostras foram coletadas a cada 3 minutos, totalizando 25 horas na base de treino e 48 horas na base de teste. As falhas foram introduzidas após 1 hora na base de treinamento, e após 8 horas na base de teste. A Figura 4 mostra o comportamento de 5 das 53 variáveis (a variável XMEAS (1) corresponde ao primeiro gráfico e a variável XMEAS (5) é o quinto gráfico, em ordem crescente de cima para baixo), onde: a faixa azul representa os valores assumidos pelas variáveis nas 500 amostras das 500 simulações na base de dados de treino. onde foi detectada a falha IDV (1); já a faixa laranja representa os valores assumidos pelas variáveis nas 500 amostras das 500 simulações na base de dados de treino onde não foi detectada falha alguma.

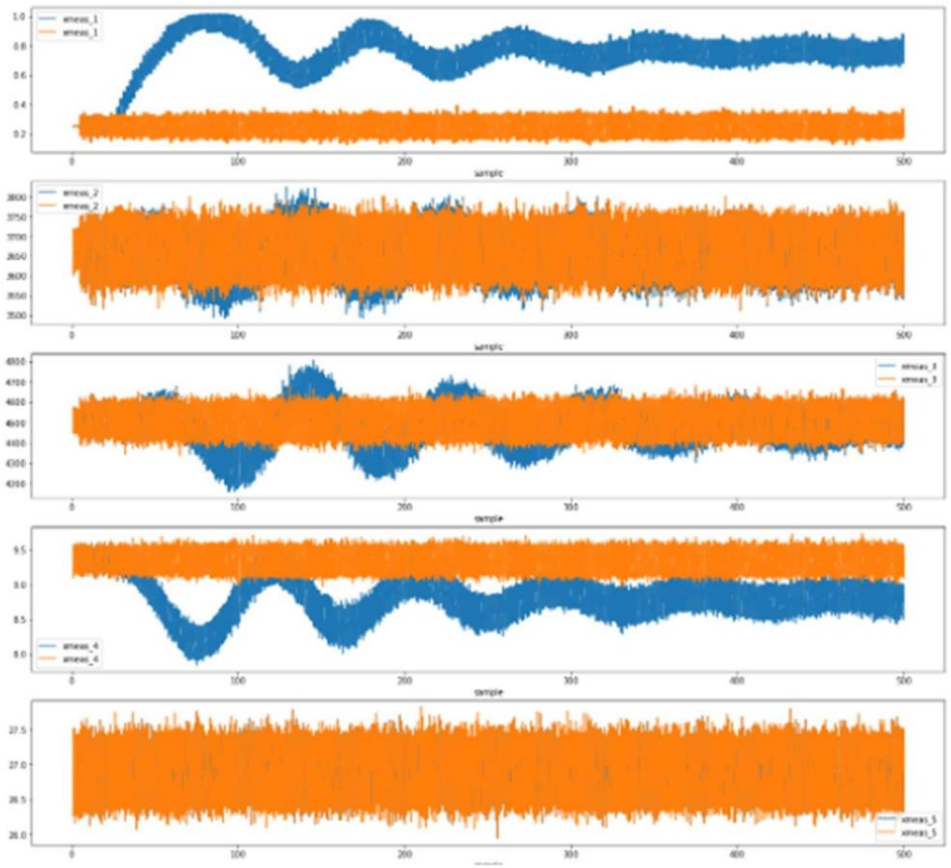

Figura 4 - Comportamento das variáveis XMEAS (1), XMEAS (2), XMEAS (3), XMEAS (4) e XMEAS (5) em todas as 500 simulações com a presença da falha IDV (1), em azul, e na ausência de falha, em laranja.

É possível observar que nas variáveis XMEAS (1), XMEAS (2), XMEAS (3) e XMEAS (4), a faixa de valores que as variáveis assumem em cada tempo das simulações na presença da falha (faixa azul) é bem diferente de quando o sistema está em funcionamento normal (faixa laranja). Já na variável XMEAS (5), a faixa de valores na presença da falha IDV (1) é semelhante à faixa de valores no funcionamento normal, ou seja, não é possível distinguir em que momento a falha IDV (1) ocorre observando apenas a variável XMEAS (5). Ambos os comportamentos podem ser observados nas demais variáveis e varia de acordo com a falha observada. Desta forma pode-se concluir que em determinadas falhas, apenas algumas das variáveis tem alterações em seus valores em relação ao comportamento normal, em parte das amostras, enquanto as demais variáveis não apresentam alterações em seus valores em relação ao comportamento normal.

3.2 Pré-processamento

Devido à natureza dos dados e com base em análises realizadas sob a ótica de engenharia química, nas quais esse projeto não tem a pretensão de aprofundar-se, foi identificada a necessidade de se realizar o pré-processamento dos dados com o intuito de melhorar o desempenho dos modelos aplicados.

3.2.1 Codificação Binária

Considerando o comportamento das simulações nas bases onde foram detectadas falhas, descrito na seção 3.1, foi constatada a possibilidade de identificar a ocorrência de falha em algumas das variáveis apenas observando se a faixa de valores alcançadas nas amostras de cada simulação eram diferentes da faixa de valores alcançadas nas amostras das bases onde não foram detectadas falhas. Tomando como base a Figura 4, por exemplo, nas variáveis XMEAS (1) e XMEAS (4) a faixa de valores assumidos na presença da falha IDV (1), em azul, é muito diferente da faixa de valores assumidos na ausência de falha, em laranja. Já nas variáveis XMEAS (2) e XMEAS (3), as faixas de valores na presença da falha IDV (1) são levemente diferentes das faixas na ausência de falha. Enquanto na variável XMEAS (5) a faixa de valores assumidos na presença da falha IDV (1) se assemelha à faixa de valores na ausência de falha. Portanto, foi decidido transformar os registros das variáveis em valores binários representando o fato de os valores das variáveis estarem ou não semelhantes aos valores observados na ausência de falha. Essa transformação se deu da seguinte maneira:

Foi definido um intervalo de normalidade para cada variável considerando a média e desvio padrão dos valores que a variável assume em todas as amostras de todas as simulações presentes na base de dados sem falha, tendo como limite inferior e limite superior o valor da média subtraída e somada, respectivamente, de três vezes o valor do desvio padrão.

Os valores das variáveis, portanto, sofreram uma codificação binária seguindo a regra:

- 0 – valor da variável dentro do intervalo de normalidade
- 1 – valor da variável fora do intervalo de normalidade

3.2.2 Janelamento Deslizante

Com a codificação binária realizada, é feita uma transformação na base de dados na forma de um janelamento deslizante. O janelamento é realizado de forma diferente entre as variáveis de entrada (atributos) e a variável classe. O janelamento dos atributos é realizado da seguinte maneira:

Considere uma simulação S

$$\boldsymbol{S} = \{\boldsymbol{A_1}, \boldsymbol{A_2}, \boldsymbol{A_3}, \ \dots, \boldsymbol{A_{500}}\} \quad \text{(Eq. 1)}$$

Sendo A_i amostras já codificadas, onde:

$$\boldsymbol{A_i} = [\boldsymbol{0}, \boldsymbol{1}], \ \boldsymbol{i} \in (\boldsymbol{1}, \boldsymbol{2}, \boldsymbol{3}, \dots, \boldsymbol{500}) \quad \text{(Eq. 2)}$$

O janelamento é uma função transformação com base na seguinte fórmula:

$$\boldsymbol{J} : \boldsymbol{S} \rightarrow \boldsymbol{S'} \quad \text{(Eq. 3)}$$

Onde:

$$\boldsymbol{S'} = \{\boldsymbol{W_1}, \boldsymbol{W_2}, \boldsymbol{W_3}, \ \dots, \boldsymbol{W_{499}}\} \quad \text{(Eq. 4)}$$

$$\boldsymbol{W_i} = \sum_{\boldsymbol{j=i}}^{\boldsymbol{i+50}} \boldsymbol{A_j}, \ \boldsymbol{A_j} \in \boldsymbol{S} \quad \text{(Eq. 5)}$$

O janelamento da variável classe é realizado da seguinte maneira:

Considere uma simulação S

$$\boldsymbol{S} = \{\boldsymbol{A_1}, \boldsymbol{A_2}, \boldsymbol{A_3}, \ \dots, \boldsymbol{A_{500}}\} \quad \text{(Eq. 6)}$$

O janelamento é uma função transformação com base na seguinte fórmula:

$$\boldsymbol{J} : \boldsymbol{S} \rightarrow \boldsymbol{S'} \quad \text{(Eq. 7)}$$

Onde:

$$\boldsymbol{S'} = \{\boldsymbol{W_1}, \boldsymbol{W_2}, \boldsymbol{W_3}, \ \dots, \boldsymbol{W_{499}}\} \quad \text{(Eq. 8)}$$

$$\boldsymbol{W_i} = \{\boldsymbol{A_i}, \boldsymbol{A_{i+1}}, \boldsymbol{A_{i+2}}, \ \dots, \boldsymbol{A_{i+50}}\} \quad \text{(Eq. 9)}$$

$$\boldsymbol{C_{W_i}} = \boldsymbol{0}, \ \boldsymbol{se\ Moda}(\boldsymbol{W_i}) = \boldsymbol{0} \quad \text{(Eq. 10)}$$

$$\boldsymbol{C_{W_i}} = \boldsymbol{X}, \ \boldsymbol{se\ Moda}(\boldsymbol{W_i}) = \boldsymbol{1} \quad \text{(Eq. 11)}$$

Onde X é o número referente a falha detectada na simulação. E C_{Wi} é a classe da falha referente a janela W_i.

3.3 Redução da Escala

Após o pré-processamento, descrito na seção 3.2.2, a base de dados está pronta para o treinamento do modelo de classificação e testes. Entretanto, considerando que as bases de dados possuem uma dimensão muito grande, como mostrado na seção 3.1, o que tornaria o tempo de processamento muito lento e proibitivo para este projeto, algumas estratégias foram adotadas para reduzir a escala dos testes e acelerar o processo de desenvolvimento. As estratégias foram:

- Descarte das Bases de Dados de Teste (*TEP_FaultFree_Testing.RData e TEP_Faulty_Testing.RData)*
- Concatenação das Bases de Treinamento (*TEP_FaultFree_Training.RData* e *TEP_Faulty_Training.RData)* para formar uma única base contendo registros com falhas e sem falhas (*TEP.RData*)
- Descarte de 80% dos registros da nova base (*TEP.RData*) de forma estratificada, ou seja, mantendo a proporção entre os tipos de falha.
- Separação de 90% da nova base (*TEP.RData)* para treinamento e validação (*TEP_TrainVal.RData*)
- Separação de 10% da nova base (*TEP.RData*) para testes (*TEP_Testing.RData*).
- Separação de 20% da nova base de treino e validação (*TEP_TrainVal.RData*) para validação (*TEP_Validating.RData*)
- Separação de 80% da nova base de treino e validação (*TEP_TrainVal.RData*) para treinamento (*TEP_Training.RData*)

3.4 Classificação das Falhas

O método escolhido para a classificação das falhas foi o *Random Forest Classifier* (seção 2.5). Os hiperparâmetros do classificador foram definidos pelos integrantes do projeto de pesquisa ao qual este projeto está apoiando, com base em diversos testes experimentais, e são os seguintes:

- Max_depth: 12
- Min_sample_split: 6
- N_estimators: 100
- Criterion: Gini

- Min_samples_leaf:20

Para a validação, foi utilizado o método de validação cruzada K-Fold com K=5.

3.5 Seleção das Variáveis Wrapper com Algoritmo Genético

Como explicado na seção 1.2 o método wrapper de seleção de variáveis utiliza o desempenho do classificador no processo de obtenção do subconjunto de variáveis de entrada. Seguindo a proposta deste projeto foi implementado um método de seleção de variáveis do tipo Wrapper utilizando Algoritmo Genético. O processo de seleção de variáveis é representado em formato de fluxograma na Figura 5.

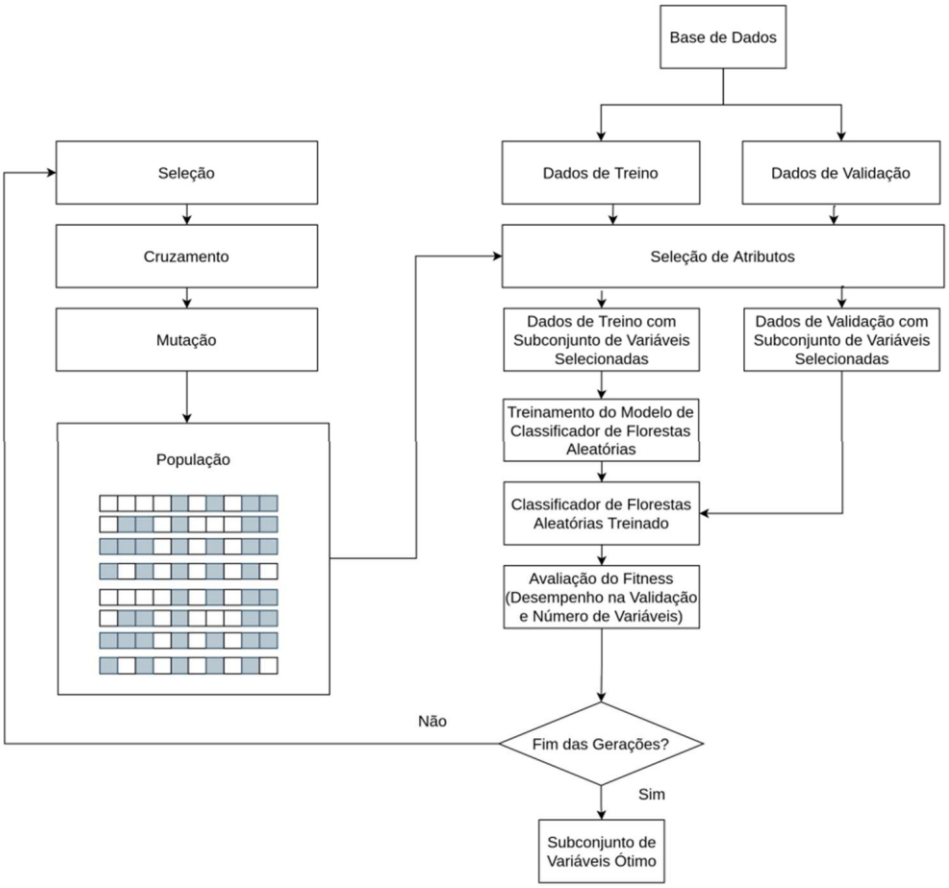

Figura 5 - Fluxograma de Algoritmo Genético para Seleção de Variáveis

Os cromossomos gerados aleatoriamente na população inicial serão utilizados para gerar diferentes subconjuntos de variáveis a serem utilizadas para treinar diversos modelos do classificador. Esses modelos serão utilizados na função de avaliação (*Fitness*) para avaliar a aptidão dos cromossomos, ou seja, o quão relevante aquele subconjunto de variáveis é para a classificação de falhas

no processo *Tennessee Eastman,* tanto em termos de acurácia na classificação de falhas, utilizando a base de validação, quanto em termos de redução do número de variáveis.

3.5.1 Função *Fitness*

A função de *Fitness* recebe como entrada as bases de dados de treino e de validação além do indivíduo que se deseja avaliar na forma de uma lista de booleanos indicando o subconjunto das variáveis selecionadas. Então a base é transformada da seguinte forma:

Considere um indivíduo I sendo:

$$\boldsymbol{I} = [\mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \dots, \mathbf{1}] \quad \text{(Eq. 12)}$$

Considere o indivíduo como uma lista de 52 posições.

Índice	0	1	2	3	4	5	6	7	8	9	10	...	51
Gene	1	0	1	1	0	0	0	0	1	0	1	...	1

Nesse cenário as variáveis cujos índices são 0, 2, 3, 8, 10 e 51 estarão presentes nas bases de dados de treino e validação transformadas, enquanto as variáveis cujos índices são 1, 4, 5, 6, 7 e 9 serão descartadas das bases.

As bases de treino e validação então são separadas de forma estratificada, i.e., mantendo a proporção das falhas em todas as divisões, em 5 partes para a validação cruzada 5-Fold. Essas partes são utilizadas para treinar o modelo e depois avaliar a sua acurácia. A função *Fitness* então retorna a média das acurácias dos 5 modelos nas 5 partes da base de validação, e o número de variáveis consideradas no indivíduo.

3.6 Hiperparâmetros do Algoritmo Genético

Durante o processo de desenvolvimento do protótipo de algoritmo genético, baseado na biblioteca DEAP (seção 2.4.3), tinha-se como meta a definição do conjunto de hiperparâmetros que melhor se adequassem ao contexto do projeto, entretanto não há na literatura uma definição específica de quais são os melhores métodos a serem utilizados caso a caso, possivelmente pela natureza heurística dos algoritmos genéticos. Desta forma é possível concluir que a melhor maneira de se definir os hiperparâmetros que oferecem ao algoritmo genético a melhor performance é experimentalmente. Portanto o desenvolvimento do algoritmo genético foi feito pensando na customização dos hiperparâmetros. Isso também

favoreceria a reutilização do código, dado que ofereceria ao utilizador do código selecionar diferentes abordagens ao gerar um algoritmo genético sem a necessidade de se aprofundar na documentação desta biblioteca.

Os hiperparâmetros escolhidos para os testes foram:

- Target – objetivo do algoritmo genético:
  - Maximização da Média da Acurácia do Classificador e Minimização do Número de Variáveis Utilizadas
- Crossover - Método Utilizado para cruzamentos dos indivíduos do GA:
  - Crossover de 1 Ponto
  - Crossover de 2 Pontos
- Número de Indivíduos da População
  - 20
  - 50
- Número de Gerações como Critério de Parada
  - 20
  - 50
- Método de Seleção:
  - Roleta
  - Torneio
- Taxa de Mutação:
  - 10%
  - 5%
- Taxa de Crossover:
  - 100%
  - 50%
- Elitismo

4 Resultados dos Testes

4.1 Melhor Conjunto de Hiperparâmetros

Considerando todas as combinações dos valores possíveis dos hiperparâmetros, foram executados 64 algoritmos genéticos diferentes para selecionar o conjunto de hiperparâmetros com o melhor desempenho na base de validação. Com o intuito de observar as melhores combinações de

hiperparâmetros, criou-se Tabela 7 onde estão apresentadas as 5 combinações de hiperparâmetros do algoritmo genético que obtiveram os melhores desempenhos em termos de acurácia na base de teste.

Tabela 7 - Maiores acurácias e número de variáveis obtida com a base de validação relacionada às características dos hiperparâmetros dos melhores indivíduos da última geração dos algoritmos genéticos executados.

Seleç.	Cruz.	Popul.	Geraç.	Mut%	Cruz%	Vars.	Acur.
Roleta	2 pts.	50	50	10%	50%	27	78,1%
Torn.	1 pt.	50	50	10%	100%	31	78,0%
Torn.	2 pts.	20	50	10%	50%	29	77,9%
Roleta	2 pts.	50	50	10%	100%	33	77,9%
Roleta	1 pt.	50	50	5%	100%	29	77,8%

4.1 Desempenho do Melhor Indivíduo

O cromossomo que representa a linha 1 da Tabela 1 tem o seguinte formato e variáveis selecionadas:

[0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1]

['xmeas_5', 'xmeas_6', 'xmeas_7', 'xmeas_10', 'xmeas_14', 'xmeas_16', 'xmeas_18', 'xmeas_19', 'xmeas_20', 'xmeas_21', 'xmeas_22', 'xmeas_23', 'xmeas_25', 'xmeas_28', 'xmeas_31', 'xmeas_33', 'xmeas_34', 'xmeas_38', 'xmeas_40', 'xmeas_41', 'xmv_1', 'xmv_2', 'xmv_4', 'xmv_7', 'xmv_9', 'xmv_10', 'xmv_11']

Na Figura 6 está representada a matriz confusão obtido a partir do conjunto de testes. Para a grande maioria das classes de falhas, o modelo conseguiu ter uma boa previsão (quadrados claros). Porém para as falhas de classe 3,9,10,15 e 16, o modelo não teve um bom desempenho. A seção seguinte apresentará o resultado dos testes de desempenho do modelo treinado com todas as variáveis contidas nas simulações para avaliar se essa falta de acurácia foi resultado da seleção de variáveis

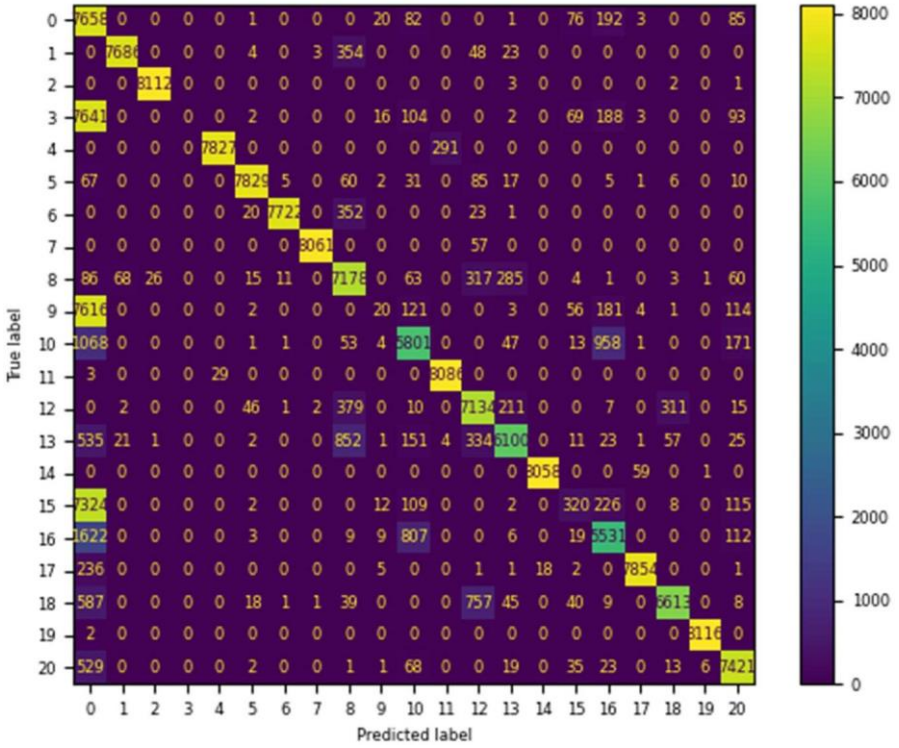

Figura 6 - Matriz de confusão da base de teste considerando o melhor conjunto de variáveis escolhido a partir do conjunto de validação.

A Figura 7 apresenta o acompanhamento da performance de indivíduos com maior pontuação de aptidão em cada geração do modelo baseado em Algoritmos Genético. O eixo horizontal apresenta o número de gerações enquanto o vertical apresenta a acurácia obtida pelo melhor indivíduo de cada geração. Pode-se observar que entre as gerações 30 e 40 o Algoritmos Genético entra em ótimo local, onde não é obtida nenhuma melhora na performance. Porém a partir da geração 40 há uma melhora na aptidão máxima alcançada.

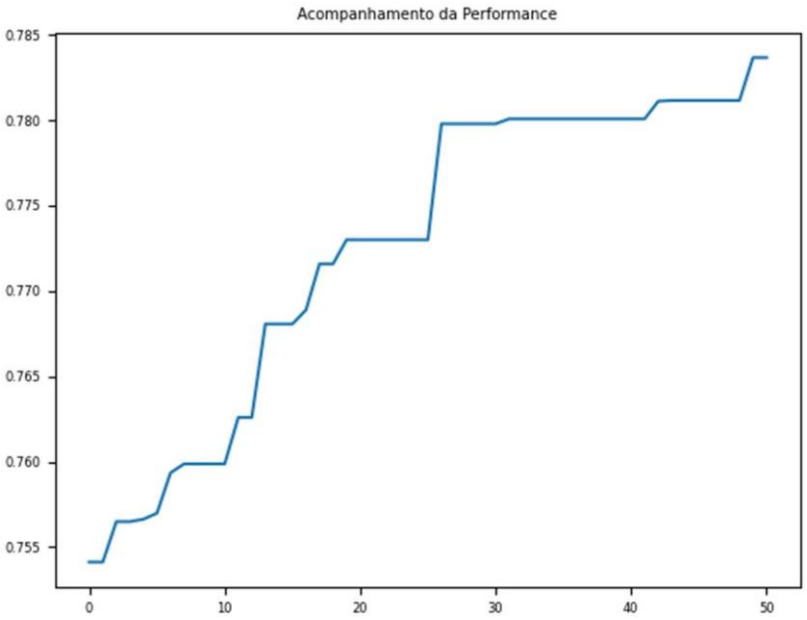

Figura 7 - Gráfico de avaliação dos melhores indivíduos por geração.

4.1 Comparação dos testes com e sem Seleção de Variáveis

Após os testes para definir os hiperparâmetros que gerassem o melhor desempenho no modelo com as variáveis selecionadas, o planejamento era de comparar esse desempenho com o modelo com todas as variáveis, ou seja, sem seleção de variáveis. Em dois testes realizados com modelos treinados utilizando todas as variáveis foi obtido uma acurácia média de 76,233% e 76,258%. A primeira Figura 9 mostra a matriz de confusão de um dos testes realizados com todas as variáveis (53).

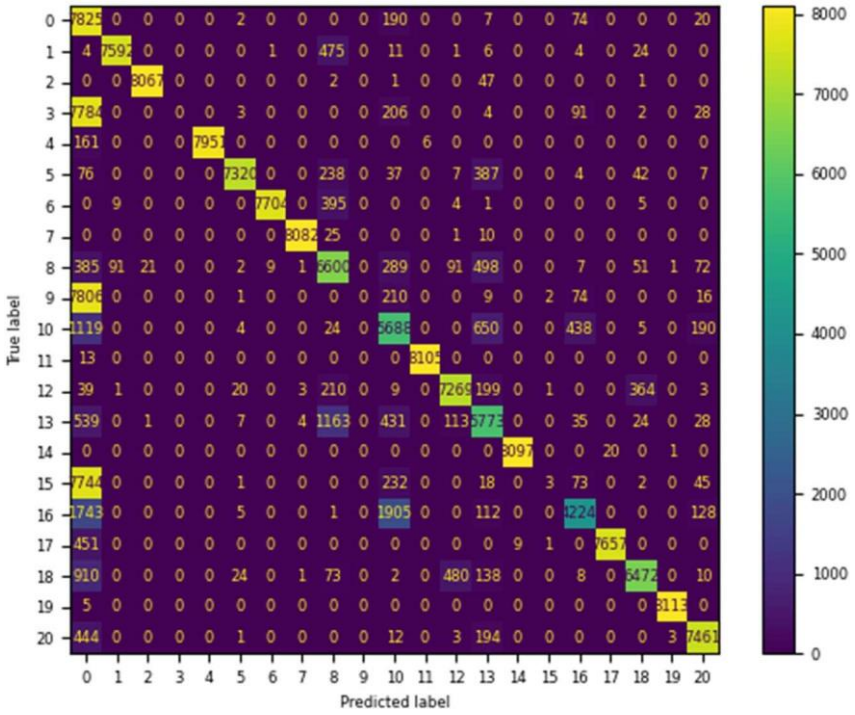

Figura 8 - Matriz de confusão da base de teste considerando o conjunto completo de variáveis.

Analisando a matriz é possível identificar que o modelo treinado utilizando todas as variáveis também obteve um desempenho ruim na predição dos valores onde a real classificação das falhas era das falhas 3,9,10,15 e 16, assim como no teste realizado com o subconjunto de variáveis de melhor desempenho. Dessa forma é possível afirmar que o desempenho ruim nas classificações dessas falhas não foi introduzido pela redução no número das variáveis. Este fenômeno pode estar associado ao pré-processamento realizado ou pela própria natureza dos fenômenos ocorridos na simulação do processo *Tennessee Eastman.*

Desconsiderando-se o baixo desempenho do modelo em classificar as falhas 3,9,10,15 e 16 e comparando as acurácias dos modelos treinados com os subconjuntos de melhor desempenho e de menor número de variáveis com a acurácia do modelo treinado com todas as variáveis é possível afirmar que o método de seleção de variáveis do tipo *Wrapper* baseado em Algoritmo Genético obteve um desempenho melhor e com uma considerável redução dos recursos necessários para a detecção de falhas em processos químicos industriais de larga escala, tendo como caso de estudo o processo *Tennessee Eastman*.

5 Conclusão e Trabalhos Futuros

O emprego da estratégia de seleção de variáveis *wrapper-based* baseado em Algoritmo Genético apresentou uma melhora pouco significativa no desempenho do modelo de classificação de falhas baseado em *Random Forest*. Foi constatado que este algoritmo *ensemble* já realiza a seleção de variáveis de forma intrínseca, ou seja, é um método de seleção de variáveis *embedded-based*.

Houve uma redução significativa no número de variáveis, em torno de 50%, o que indica que a seleção de variáveis *wrappe-based* baseada em algoritmo genético se torna relevante para a redução do número de variáveis, remoção das variáveis pouco relevantes e a consequente sintetização dos dados necessários para classificar falhas em processos químicos de larga escala. Entretanto, é necessário validar essa redução utilizando outro algoritmo de *Machine Learning* para treinamento do modelo de classificação de falhas no intuito de comparar esse método de seleção de variáveis com o próprio *Random Forest*.

Um ponto pertinente de se avaliar em trabalhos futuros é o fato de não observar uma convergência no gráfico de evolução da performance do melhor indivíduo do Algoritmo Genético com o desempenho entre os experimentos. Um novo conjunto de hiperparâmetros pode ser avaliado bem como o aumento no número de gerações, para que se dê mais tempo para o Algoritmo Genético atingir a convergência.

Outro aspecto que deve ser levado em consideração em trabalhos futuros é a grande taxa de erro em determinadas falhas, onde o modelo classificava as falhas 3, 9, 10, 15 e 16 como falha 0, ou seja, ausência de falhas. Uma nova forma de pré-processamento na base de dados do modelo *Tennessee Eastman* deve ser desenvolvida de modo que todas as falhas sejam classificadas com acurácia satisfatória.

6 Referências

Caccavale, F., et al. **An integrated approach to fault diagnosis for a class of chemical batch processes.** Dipartimento di Ingegneria e Fisica dell'Ambiente, Università degli Studi della Basilicata, Basilicata, Italy, Disponível em: https://www.sciencedirect.com/science/article/pii/S0959152408001583 2009. Acesso em: 24/05/2022

Xiao, B., et al. **Decentralized PCA modeling based on relevance and redundancy variable selection and its application to large-scale dynamic process monitoring**.School of Automation, Central South Univeristy, Changsha, China, Disponível em:

https://www.sciencedirect.com/science/article/pii/S095758202100299#bib0170
2021. Acesso em: 23/05/2022

Ying, S., et al. **Data-Based Techniques Focused on Modern Industry: An Overview**. in IEEE Transactions on Industrial Electronics, vol. 62,
Disponível em:
https://ieeexplore.ieee.org/document/6748057
2015. Acesso em: 24/05/2022

Soares, F., **Técnicas de Machine Learning aplicadas a inferência e detecção e diagnóstico de falhas de processos químicos industriais em contexto de Big Data.** Escola de Química, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil. Disponível em:
https://minerva.ufrj.br/F/?func=direct&doc_number=000858624&local_base=UFR01#.YqCIblzMKA0
2017. Acesso em: 05/06/2022

Roh, Y., Heo, G. and Whang, S.E., **A Survey on Data Collection for Machine Learning: A Big Data – AI Integration Perspective**.IEEE Transactions on Knowledge and Data Engineering, vol. 33. Disponível em:
https://ieeexplore.ieee.org/document/8862913
2021. Acesso em: 23/05/2022

Adadi, A., **A Survey on data-efficient algorithms in big data era**. Journal of Big Data. Disponível em:
https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00419-9
2021. Acesso em: 23/05/2022

Wang, Y., Jiang, Q. and Fu, J. **Data-Driven Optimized DIstributed Dynamic PCA for Efficient Monitoring of Large-Scale Dynamic Processes.** School of Eletric Engineering, Shanghai Dianji University, Shanghai, China. Disponível em:
https://ieeexplore.ieee.org/document/8026142
2017. Acesso em: 05/06/2022

Chebel-Morello, B., Malinowski, S. and Senoussi, H. **Feature Selection for fault detection systems: application to the Tennessee Eastman Process.** Applied Intelligence, Springer Verlag, Germany. Disponível em:
https://hal.archives-ouvertes.fr/hal-01303484/document
2016, Acesso em: 14/06/2021

Hui, K. H. et al., **Na improved Wrapper-based feature selection method for machinery fault diagnosis.** Institute of Noise and Vibration, University Teknologi Malaysia, Kuala Lumpur, Malaysia. Disponível em:
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0189143
2017, Acesso em: 14/06/2021

Ahuja, J. and Ratnoo, S. D., **Feature Selection using Multi-objective Genetic Algorithm: A Hybrid Approach**. Guru Jambheshwar University of Science and Technology, Hisar, Haryana, India. Disponível em:
https://www.semanticscholar.org/paper/Feature-selection-using-multi-objective-genetic-a-Ahuja/1506b1d9baa76cf313959a6cde67bb2652c513f8
2015. Acesso em: 16/04/2021

Pramonckchon, P, and Piamsa-Nga, P., **Effective threshold estimation for filter-based feature selection**. Disponível em:

https://ieeexplore.ieee.org/document/7859919
2016 Acesso em: 05/06/2022

Du, L. et al., **A Filter-based Unsupervised FeatureSelection Method via Improved Local Structure Preserving.** School of Computer and Infomration Technology, Shanxi University, Taiyuan, China. DIsponível em:
https://ieeexplore.ieee.org/document/8802793
2019 Acesso em: 05/06/2022

Clavijo, N., et al. **Variable Selection for Fault Detection Based on Causal Discovery Methods: Analysis of an Actual Industrial Case**. Universidade Federal do Rio de Janeiro, RJ. Brasil. Disponível em: https://www.mdpi.com/2227-9717/9/3/544/htm.
2021. Acesso em: 16/04/2021

Zoric, B., Drazen, B. and Dudjak, M. **Wrapper-based feature selection via differential evolution: Benchmarking different discretisation techniques.** Faculty of Eletrical Engineering, Computer Science and Information Technology Osijek, Osijek, Croatia. Disponível em:
https://ieeexplore.ieee.org/document/9263700
2020, Acesso em 05/06/2022

Wah, Y.B. et al., **Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy.** Petanika Journal of Science and Tecnology. Disponível em
https://www.researchgate.net/publication/322920304_Feature_selection_methods_Case_of_filter_and_wrapper_approaches_for_maximising_classification_accuracy
2018 Acesso em 05/06/2022

Chen, J. et al., **Ensemble Feature Selection Method for Short-Term Electrical Load Forecasting.** State Grid Human Eletric Power Company Limited Research Institute Changsha, China. DIsponível em:
https://ieeexplore.ieee.org/document/9062042
2019 Acesso em: 05/06/2022

Zhang J., Xiong Y. and Min S., **A new hybrid filter;wrapper algorithm for feature selection in classification.** College of Science, China Agricultural University, Beijing, China. DIsponível em:
https://www.sciencedirect.com/science/article/pii/S0003267019307810
2019, Acesso em 05/06/2022

Wang, A. et al., **Accelerating wrapper-based feature selection with K-nearest-neighbour.** School of Computer and Information, Hefei University of Technology, Hefei, China. DIsponível em:
https://www.sciencedirect.com/science/article/pii/S0950705115001033?via%3Dihub
2015, Acesso em: 05/06/2022

Jack, L.B., and Nandi, A., **Genetic Algorithms for feature selection in Machine condition monitoring with vibration signals.** Signal Processing and Communication Division Department of Eletronic and Eletrical Engineering, University of Liverpool, Brownlow Hill, Liverpool, UK. Disponível em:
https://www.researchgate.net/publication/3359098_Genetic_algorithms_for_feature_selection_in_machine_condition_monitoring_with_vibration_signals

2000, Acesso em: 29/06/2021

Ali, T., Nawaz, A. and Sadia H., **Genetic Algorithm Based Feature Selection Technique for Eletroencephalography Data.** Applied Computer Systems, PMAS Arid Agriculture University, Rawalpindi, Pakistan. Disponível em: https://www.researchgate.net/publication/339408755_Genetic_Algorithm_Based_Feature_Selection_Technique_for_Electroencephalography_Data
2019, Acesso em: 29/06/2021

Cerrada, M. et al., **Multi-Stage Feature Selection by Using Genetic Algorithms for Fault Detection in Gearboxes on Vibration Signal.** Sensors
Disponível em: https://www.researchgate.net/publication/281863266_Multi-Stage_Feature_Selection_by_Using_Genetic_Algorithms_for_Fault_Diagnosis_in_Gearboxes_Based_on_Vibration_Signal
2015, Acesso em: 29/06/2021

Oluleye, B. et al., **A Genetic Algorithm-Based Feature Selection.** International Journal of Eletronics Communication and Computer Engineering. Disponível em: https://www.researchgate.net/profile/Oluleye-Babatunde/publication/264545487_A_Genetic_Algorithm-Based_Feature_Selection/links/5557580b08ae6fd2d824efad/A-Genetic-Algorithm-Based-Feature-Selection.pdf
2014, Acesso em: 29/06/2021

Chiesa, M., et al. **GARS: Genetic Algorithm for the identification of a Robust Subset of Features in high-dimensional datasets.** BMC Bioinformatics. Disponível em: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-020-3400-6
2020, Acesso em: 14/06/2021

García-Dominguez, A., et al. **Feature Selection Using Genetic Algorithms for the Generation of a Recognition and Classification of Children Activities Model Using Enironmental Sound.** Unidad Académica de Ingeniería Elétrica, Universidad Autónoma de Zacatecas, México. Disponível em: https://www.hindawi.com/journals/misy/2020/8617430/
2020. Acesso em: 14/06/2021

Wutzl, B., et al. **Genetic Algorithms for Feature Selection when Classifying Severe Chronic Disorder of Consciousness.** US National Library of Mediciene National Institutes os Health. Disponível em: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6622536/
2019. Acesso em: 14/06/2021

Down, J.J., and Vogel. E.F. **A plant-wide industrial process control problem.** Eastman Chemical Company, Kingsport, Tennessee, U.S.A. Disponível em: https://www.sciencedirect.com/science/article/pii/009813549380018I
1993. Acesso em: 14/06/2021

Martin-Villalba, C., Urquia, A. and Shao, G., **Implementations of the Tennessee Eastman Process in Modelica**. Departamento de Informática y Automática, UNED, Madrid, Espanha. Disponível em: https://www.sciencedirect.com/science/article/pii/S2405896318301095

2018. Acesso em: 16/04/2021

Russel E., Chiang, L. and Braatz R., **Data-Driven Techniques for Fault Detection and Diagnosis in Chemical Processes (Advances in Industrial Control).** Cap. 8. Páginas 99-105. Disponível em: https://link.springer.com/chapter/10.1007/978-1-4471-0409-4_8
2000 Acesso em: 05/06/2022

Bäck, T. **Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms**. Páginas 7, 8 e 9. Disponível em: https://books.google.com.br/books
1995, Acesso em: 16/05/2022

Goldberg, D.E. **Genetic Algorithms in Search, Optimization and Machine Learning**. Addinson-Wesley. 1989

Holland, J. **Adaptation in natural and artificial systems**. 1975

Bajpai P. and Kumar M., **Genetic Algorithm - na Approach to Solve Global Optimization Problems.** Amity Institute of Information Technology, Amity University. Lucknow, Uttar Pradesh, India. Disponível em: https://www.researchgate.net/publication/283361244_Genetic_Algorithm_-_an_Approach_to_Solve_Global_Optimization_Problems
2010 Acesso em: 05/06/2022

Sloss, A. and Gustafson, S., **2019 Evolutionary Algorithms Review**. ArXiv.org, Cornell University, New York, United States. Disponível em: https://arxiv.org/abs/1906.08870
2019, Acesso em: 16/05/2022

Gabriel, P. and Delbem, A., **Fundamentos de Algoritmos Evolutivos.** ICMC-USP São Carlos, Disponível em: https://repositorio.usp.br/item/001687219
2008. Acesso em: 05/06/2022

Kantoch, S., Chaunhan, S. and Kumar, V., **A review on genetic algorithm: past, present, and future.** Computer Science and Engineering Department, National Institute of Technology, Hamipur, India. Disponível em: https://link.springer.com/article/10.1007/s11042-020-10139-6
2020. Acesso em: 05/06/2022

Spears, W. and Anand, V., **A Study of Crossover Operators in Genetic Algorithms.** Navy Centrer for Applied Research in AI. Naval Research Laboratory, Washington, D.C. United States of America. Disponível em: https://link.springer.com/chapter/10.1007/3-540-54563-8_104
1991. Acesso em: 05/06/2022

Michalewicz, Z. and Schoenauer, M., **Evolutionary Algorithms for COntrained Parameter Optimization Problems.** Department of COmputer Science, University of North Carolina, Charlotte, United States of America. Disponível em: https://ieeexplore.ieee.org/document/6791784
1996. Acesso em: 05/06/2022

Hildayanti, I., Soesanti, I. and Permanasari, A., **Performance Comparison of Genetic Algorithms Operator COmbinations for Optmization Problems.** Dept. Of Eletrical Engineering and Information Technology. Univeritas Gadjah Mada, Yogyakarta, Indonesia. Disponível em:
https://ieeexplore.ieee.org/document/8864469
2018. Acesso em: 05/06/2022

Fortin, F. et al. DEAP: **Evolutionary Algorithms Made Easy. Laboratoire de vision et systèmes numériques**. Département de génie életrique et de génie informatique. Université Laval. Québec, Canada. DIsponível em:
https://www.researchgate.net/publication/235707001_DEAP_Evolutionary_algorithms_made_easy
2012 Acesso em: 05/06/2022

Santi, G., **Diagnóstico de Falhas em Processos Industriais Usando Classificadores Locais Avaliados com Diferentes Características.** Universidade do Espírito Santo. Centro Tecnológico Departamento de Engenharia Elétrica. Disponível em:
http://repositorio.ufes.br/bitstream/10/9563/1/tese_10890_Disserta%c3%a7%c3%a3o%20-%20Gustavo%20Boina%20Santi.pdf
2018. Acesso em: 05/06/2022

Barbosa, J., Carneiro, T. and Tavares, A., **Métodos de Classificação por Árvores de Decisão.** Programa de Pós-Graduação em Ciência da Computação do Departamento de Computação da Universidade Federal de Ouro Preto. Disponível em:
http://www.decom.ufop.br/menotti/paa111/files/PCC104-111-ars-11.1-JulianaMoreiraBarbosa.pdf
2012. Acesso em: 05/06/2022

Mitchel, T., **Machine Learning.** Cap. 3. Disponível em:
https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf
1997. Acesso em: 05/06/2022

Hastie, T., Tibshirani, R. and Friedman, J., **The Elements of Statistical Learning: Data Mining, Inference and Prediction.** Second Edition. Springer. ISBN 0-387-95284-5. Disponível em:
https://hastie.su.domains/ElemStatLearn/
2009. Acesso em: 05/06/2022

Sagi, O. and Rokach, L., **Ensemble Learning: A Survey.** Department of Software and Information Systems Engineering, Bem-Gurion University, Beersheba, Israel. Disponível em:
https://wires.onlinelibrary.wiley.com/doi/10.1002/widm.1249
2018. Acesso em: 05/06/2022

Saini, R. and Gosh, S., **Ensemble classifiers in remote sensing: A review.** Indian Institute of Technology (IIT) Roorkee, India. Disponível em:
https://ieeexplore.ieee.org/document/8229969
2017. Acesso em: 05/06/2022

Breiman, L., **Random Forests.** Machine Learning**.** V. 45,5-32. Disponível em:
https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf
2001. Acesso em: 05/06/2022

Jovic, A., Brkic, K. and Bogunovic., **A Review of feature selection methods with applications**. Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. Disponível em: https://bib.irb.hr/datoteka/763354.MIPRO_2015_JovicBrkicBogunovic.pdf 2015, Acesso em: 14/06/2021