

## UMA ABORDAGEM BASEADA EM APRENDIZADO DE MÁQUINA PARA DIAGNÓSTICO DE FALHAS NO SISTEMA APS – SCANIA<sup>1</sup>

Marcus F. S. Meneses<sup>a</sup>, Rogério C. N. Rocha<sup>a</sup>, Laércio I. Santos<sup>b</sup>, Murilo O. Camargos<sup>a</sup>,  
Marcos F. S. V. D'Angelo<sup>c</sup> \*

<sup>a</sup> Programa de Pós-graduação em Modelagem Computacional e Sistemas  
Universidade Estadual de Montes Claros (UNIMONTES), Montes Claros-MG, Brasil

<sup>b</sup> Campus Montes Claros  
Instituto Federal do Norte de Minas (IFNMG), Montes Claros-MG, Brasil

<sup>c</sup> Departamento de Ciência da Computação  
Universidade Estadual de Montes Claros (UNIMONTES), Montes Claros-MG, Brasil

Recebido 28/05/2023, aceito 29/12/2023

### RESUMO

O gerenciamento de frotas de caminhões é crucial para garantir a segurança dos motoristas e prevenir riscos financeiros e materiais devido a falhas operacionais. A tecnologia de pressão a ar, APS, oferecida pela SCANIA, é uma solução capaz de coletar dados precisos e fornecer informações valiosas para a manutenção e operação adequadas do veículo. Para aprimorar a detecção de falhas, propõe-se uma abordagem de aprendizado de máquina por meio de Árvores de Decisão induzidas por Programação Genética. Foram utilizados dados públicos para treinar o modelo. Os resultados obtidos mostram que o método performa melhor na classe positiva (sistema com falha) em comparação com trabalhos do estado da arte. Além disso, esta abordagem fornece modelos de previsão mais interpretáveis, o que é importante para identificar sistemas com falhas, corrigi-las, prevenir acidentes e aumentar a segurança no trânsito.

**Palavras-chave:** detecção de falhas, árvore de decisão, programação genética.

### ABSTRACT

Truck fleet management is crucial to ensure driver safety and prevent financial and material risks due to operational failures. The Air Pressure System (APS) technology offered by SCANIA is a solution that can collect accurate data and provide valuable information for proper vehicle maintenance and operation. To enhance fault detection, a machine learning approach using Genetic Programming-based Decision Trees is proposed. Public data was used to train the model. The results show that the method performs better in the positive class (system with faults) compared to state-of-the-art works. Additionally, this approach provides more interpretable prediction models, which is important to identify systems with faults, correct them, prevent accidents, and increase road safety.

**Keywords:** fault detection, decision tree, genetic programming.

---

\* Autor para correspondência. E-mail: marcos.dangelo@unimontes.br  
DOI: <https://doi.org/10.4322/PODes.2024.001>

## 1. Introdução

A tecnologia de pressurização a ar (APS – Sistema de Processamento de Ar) é um tipo de equipamento para veículos pesados, que trabalha a ar comprimido, auxiliando a renegeração da frenagem e administrando operacionalmente elementos mecânicos. As informações são repassadas ao condutor por meio de sensores (Rafsunjani et al., 2019). As falhas na unidade APS são na maioria das vezes em função de condições de funcionamento ou erros operacionais nos componentes mecânicos, como em válvulas de retenção, válvulas solenoides e anéis de vedação. Os dados dessas falhas são operacionais e diários, recolhidos dos sensores de caminhões Scania, que apresentaram erros mecânicos em serviço. O conjunto de dados é disponibilizado pelo fabricante e permite isolar os componentes que causaram ou causam falha. É importante que haja uma melhor compreensão do sistema completo da unidade APS, pois avisos da unidade APS nem sempre significam que existem falhas no sistema. Portanto, é necessário sempre verificar todos os circuitos do sistema para eliminar situações redundantes e garantir uma boa condição operacional do motor e dos elementos mecânicos (Scania, 2017).

Existem poucos estudos que utilizam modelos interpretáveis de Aprendizagem de Máquina (ML, do inglês *Machine Learning*) na previsão de falhas em Unidades de APS. A maioria dos estudos mencionados utiliza modelos como Floresta Aleatória, SVM (Support Vector Machine) e KNN (K-Nearest Neighbors), que são modelos de ML mais tradicionais. Um exemplo de estudo que utiliza um modelo interpretável é o trabalho de Gondek et al. (2016), que utiliza histogramas e um classificador baseado em Floresta Aleatória para prever falhas em unidades APS. Esta abordagem permite a criação de novas características interpretáveis com base nos histogramas e, junto com o modelo de Floresta Aleatória, é capaz de identificar corretamente as falhas. Outro estudo, que aborda o problema, é o trabalho de Ozan et al. (2016). Neste estudo, além de utilizar Floresta Aleatória, SVM e KNN, os autores propõem uma nova estratégia de imputação de valores ausentes para melhorar os resultados da previsão de falhas. O estudo realizado por Rafsunjani et al. (2019) também compara diferentes técnicas de imputação de valores ausentes, juntamente com os classificadores SVM e KNN. O objetivo deste estudo é avaliar os efeitos das técnicas de imputação em relação ao problema de desbalanceamento de dados, que é comum em conjuntos de dados de falhas. Já o trabalho de Syed et al. (2021) apresenta uma abordagem de aprendizado de máquina projetada para lidar com o desbalanceamento de dados que utiliza uma fronteira de decisão linear maximizando o critério de Área Sob a Curva. Embora esses estudos sejam relevantes e utilizem técnicas de ML, é importante destacar a falta de trabalhos que explorem modelos interpretáveis de ML na previsão de falhas em unidades APS. Esses modelos interpretáveis podem ajudar a entender o processo de tomada de decisão dos modelos de ML e fornecer informações valiosas para os especialistas em manutenção das unidades APS.

Técnicas de ML são interessantes, porque emulam a maneira humana de pensar e tomar decisões, analisam grandes conjuntos de dados contendo muitas características em um tempo razoável e podem lidar com relacionamentos complexos entre dados (Barros et al., 2012). O ML pode ajudar em uma avaliação mais ágil, com a possibilidade de diminuição das tentativas de identificação de uma falha operacional mecânica, tendo como consequência uma redução do custo de manutenções corretivas e preventivas. No entanto, o desempenho dessas técnicas é reduzido por existirem valores ausentes e o problema de dados desbalanceados (Santos et al., 2022).

Dentre as principais técnicas de ML, as Árvores de Decisão se destacam devido à natureza compreensível de seus modelos de previsões que se assemelha ao raciocínio humano. Além disso, algoritmos de indução de árvore de decisão apresentam várias vantagens sobre outros algoritmos de aprendizado, como robustez ao ruído, baixo custo computacional para a construção e utilização do modelo e a capacidade de lidar com atributos redundantes (Quinlan, 1987).

Entretanto, tais algoritmos geralmente utilizam uma estratégia gulosa e tendem ficar presos em ótimos locais. Ademais, a política de particionamento recursivo na fase de construção pode resultar em conjuntos de dados com baixa cardinalidade para o processo de seleção de atributos em nós mais profundos da árvore, causando um super-ajuste dos dados (Barros et al.,

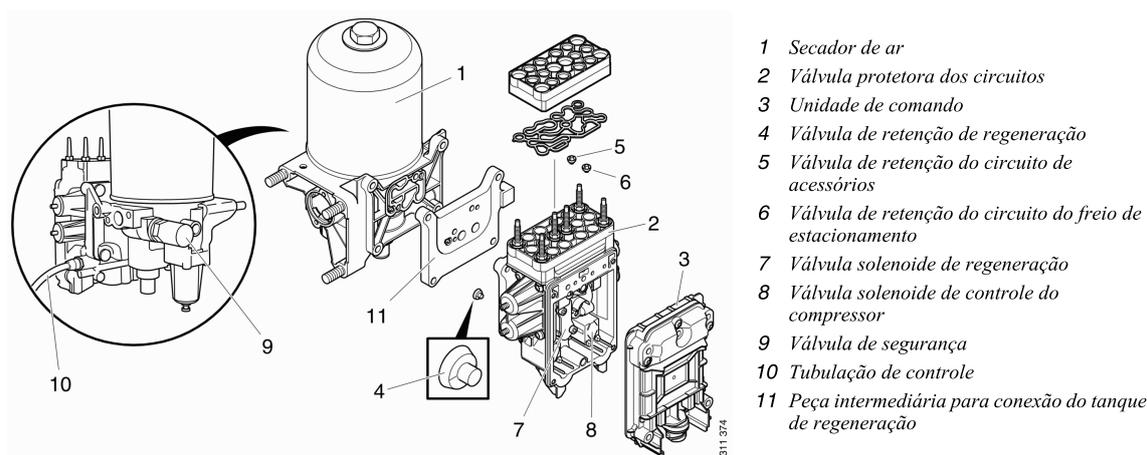
2012). Uma boa prática tem sido a aplicação de algoritmos evolutivos para induzir árvores de decisão, especificamente, por meio de Programação Genética (PG). PG é um algoritmo evolutivo que tem como característica evoluir um conjunto de indivíduos representados na forma de árvores. Quando PGs são aplicadas na indução de árvores de decisão, é possível lidar com vários atributos ao mesmo tempo, reduzindo a dependência de métodos de seleção de características no pré-processamento e ainda prover uma busca global (Kubat et al., 1998).

Portanto, esse estudo consiste na contribuição de utilizar uma abordagem de Programação Genética (PG) para induzir Árvores de Decisão e posteriormente aplicar tais árvores na identificação de falhas relacionadas com o APS. Essa identificação pode agilizar as manutenções corretivas, periodizando adequadamente as manutenções preventivas, consequentemente provendo maior segurança a veículos pesados que utilizam o APS e uma melhor condição de faturamento pelo uso mais adequado da frota de ônibus ou caminhões.

O restante desse trabalho é organizado da seguinte maneira: na Seção 2 é apresentado o *Benchmark* de detecção de falhas da *Scania*. Seção 3 apresenta a metodologia baseada em árvores de decisão induzidas por programação genética, metodologia essa usada como detector de falhas. A Seção 4 apresenta a aplicação da nova metodologia proposta o *Benchmark* de detecção de falhas da *Scania* e seus resultados. Finalmente, as conclusões são apresentadas na Seção 5.

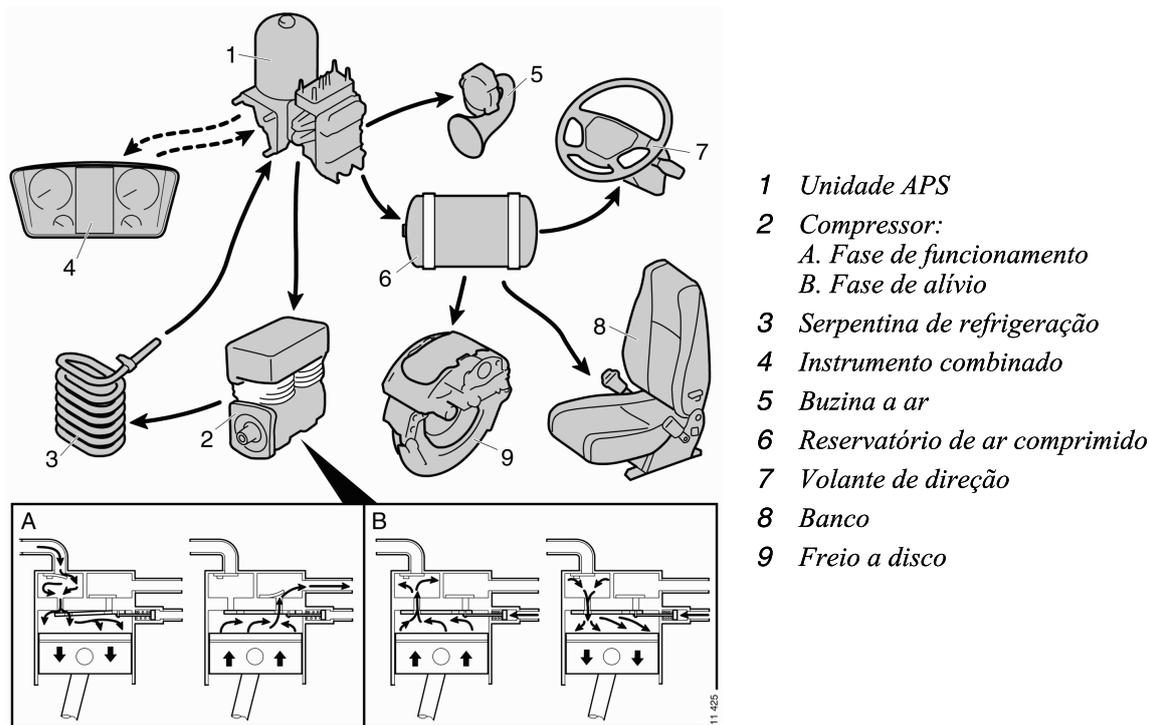
## 2. Benchmark do Sistema de Processamento de Ar (APS) da Scania

Figura 1: APS - Detalhamento do APS



Fonte: Scania (2017)

Figura 2: APS - Sequência de Utilização do Compressor



Fonte: Scania (2017)

A Scania oferece duas versões da unidade APS: a avançada e a de alta capacidade, sendo a diferença entre elas a quantidade de reservatórios do dessecante. A primeira contém um reservatório, enquanto a segunda possui dois, indicada para veículos com alto consumo de ar. A válvula protetora do circuito é a mesma nas duas versões, porém, a regeneração é a única diferenciadora funcional. Os detalhes da unidade APS podem ser encontrados nos trabalhos de referência (Biteus e Lindgren, 2017a) e (Biteus e Lindgren, 2017b).

A unidade APS (Figura 1) tem a função de gerenciar a pressão do ar no sistema pneumático do veículo, drenando e limpando o ar proveniente do compressor, além de secar e limpar cerca de  $250 \text{ dm}^3$  de ar antes que o filtro esteja saturado. O processo de regeneração é ativado pelo controle do APS quando o volume de ar atinge o nível do filtro saturado, com o uso da válvula solenoide de regeneração que sopra o ar de volta para o recipiente dessecante (Figura 2). Durante a regeneração, não é possível pressurizar o sistema pneumático.

O veículo Scania é composto por três redes de comunicação coloridas, denominadas verde, laranja e vermelha (Figura 3), que indicam a condição de funcionamento em que se encontram. Os sensores de pressão na unidade APS registram e transmitem informações para outros sistemas por meio da rede CAN (do inglês, *Controller Area Network*), indicando a pressão no freio de estacionamento, frenagem dianteira e traseira. A lâmpada conectada ao sensor de frenagem e ao APS começa a piscar e a cigarra soa quando há queda de pressão para 6 bar ou inferior.

A unidade APS usa a válvula protetora de circuito para regular a ordem de pressurização dos circuitos, o compressor e a regeneração são controlados pelo coordenador (unidade de comando), levando em consideração parâmetros como carga do motor. A unidade de comando se comunica com toda a rede CAN, gerenciando o APS e verificando e regulando a função de regeneração e compressão. O secador de ar varia de acordo com o modelo do caminhão, podendo ter um ou dois reservatórios com o dessecante.

Por fim, a eletrônica de controle do APS é a mesma nas duas versões, porém, o coordenador é configurado de maneira diferente para cada uma delas. O APS monitora todos os sensores e controla as válvulas solenoide, sendo conectado à rede CAN, de cor laranja do veículo, o que

permite a recuperação das informações necessárias pelos sensores do APS e de outros sistemas interligados à rede (Figura 4).

## **2.1. Equipamentos interligados pela rede CAN ao APS**

Esta Seção apresenta alguns dos principais equipamentos interligados à rede CAN do caminhão, que se comunicam com o APS.

**Válvula de segurança:** a válvula de segurança tem uma pressão de abertura de 13 a 14,5 bar, e limita a pressão do ar, esvaziando o sistema de ar comprimido quando ela está muito alta;

**Reservatório do dessecante:** o reservatório do dessecante, diminui a quantidade de umidade e a impede que entre no sistema de ar comprimido. O dessecante é drenado durante a regeneração, isso acontece quando o ar comprimido passa através da válvula de drenagem;

**Válvulas de retenção:** as válvulas de retenção denominadas CVG1 e CVG2 impedem a fuga do ar comprimido da válvula de proteção e retorno para o secador e as válvulas de retenção denominadas CVR1 e CVR2 garantem que o ar comprimido não vaze para o tubo de controle e para a válvula de drenagem (pressurização do sistema pneumático durante a regeneração). As válvulas de retenção também impedem, que o ar do circuito do freio de estacionamento e do acessório flua para os circuitos do freio de serviço;

**Válvula solenoide V196:** possui a função de controlar qual filtro ficará ativo e qual será regenerado;

**Válvula solenoide de alívio do compressor:** esta válvula solenoide atua no alívio do compressor, uma vez que haja pressurização;

**Válvula de Desvio:** válvula que atua priorizando a frenagem dianteira em detrimento da frenagem de estacionamento, caso a pressão no freio dianteiro seja menor que a de estacionamento, à vazão de 33 a 55 litros por minuto, a uma pressão de 6,0 bar;

**Válvula de proteção:** controla a carga do sistema de ar comprimido. Assegura, que a pressurização do freio de estacionamento seja aplicada antes mesmo do atendimento aos freios dianteiro e traseiro. Se o circuito dianteiro for despressurizado, o circuito do freio de estacionamento será drenado pela válvula de desvio. Quando a válvula solenoide está aberta, o ar comprimido flui por meio desta, abrindo a válvula de drenagem. Ainda com relação a válvula de proteção, que atua no circuito do freio de estacionamento, a pressão de abertura da válvula prioritária é 7,8 bar. Como a válvula protetora é alimentada a partir dos circuitos do freio de serviço, o freio de estacionamento não pode ser liberado, enquanto os freios de serviço não tiverem pressão suficiente, a válvula de prioridade fecha a pressão  $\geq 3,1$  bar.

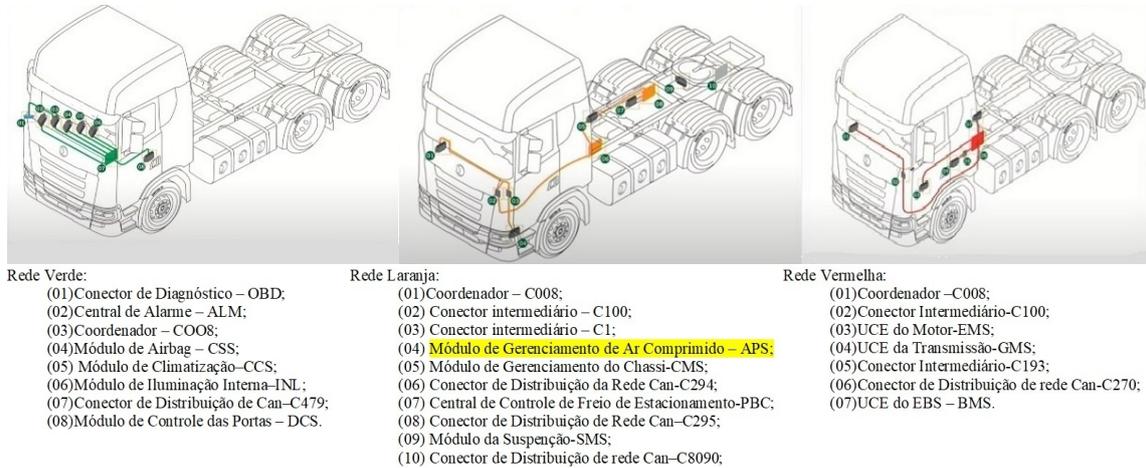
**Rede CAN (Rede de Comunicação):** É a rede de sensores de pressão do circuito que estão presentes em todo veículo, incluso o de estacionamento e trem de força, exemplificados na figura 3. O coordenador é o principal instrumento da Rede CAN.

**Conector de Diagnóstico (OBD):** Neste conector as informações dos sensores de pressão dos circuitos são transmitidas através da comunicação CAN e disponibilizadas no instrumento de modo combinado.

Resumindo, aqui foram apresentados alguns dos principais equipamentos interligados à rede CAN do caminhão, que se comunicam com o APS. A válvula de segurança, por exemplo, tem como função limitar a pressão do ar, esvaziando o sistema de ar comprimido quando ela atinge níveis muito altos. Já o reservatório do dessecante reduz a quantidade de umidade no sistema de ar comprimido e a válvula de retenção impede a fuga de ar comprimido e vazamentos nos circuitos dos freios dianteiro, traseiro e de estacionamento. A válvula solenoide V196 controla

a regeneração dos filtros e a válvula de desvio prioriza a frenagem dianteira em detrimento da de estacionamento, desde que a pressão no freio dianteiro seja menor. Além disso, temos a rede CAN de comunicação, que conta com sensores de pressão presentes em todo o veículo, incluindo o circuito de estacionamento e o trem de força, e o conector de diagnóstico (OBD), que disponibiliza informações dos sensores de pressão dos circuitos ao instrumento combinado.

Figura 3: Rede Can dos caminhões Scania



Fonte: Scania (2017)

Figura 4: Painel de instrumentos dos caminhões Scania



Fonte: Scania (2017)

### 3. Abordagem Baseada em Árvore de Decisão Gerada por Programação Genética

A Programação Genética (PG) é uma técnica de computação evolutiva que simula o princípio de seleção natural de Darwin por meio de operadores genéticos como reprodução, recombinação e mutação (Banzhaf, 1998). Os sistemas baseados em PG podem representar a solução candidata para um problema de várias maneiras, sendo as árvores de decisão bastante frequentes. Nessa representação, cada indivíduo da população possui ramificações ordenadas em que os nós internos são funções enquanto as folhas da árvore são os terminais do problema. Cada árvore é uma solução

candidata para o problema e, como em outros algoritmos em computação evolutiva, elas são avaliadas com uma medida de qualidade (ou fitness) que reflete o quão boa é uma solução em relação às outras na mesma população (Zhao, 2007).

Neste trabalho, a predição utilizando árvores de decisão ocorre da seguinte maneira. A instância  $(\mathbf{x}_i, c_i) \in \mathcal{X}$ , tal que  $\mathbf{x}_i \in \mathbb{R}^d$ , possui  $d$  atributos que serão pré-processados para assumir um conjunto finito de valores. As variáveis contínuas serão discretizadas em  $N$  categorias usando seus pontos percentuais, enquanto as variáveis categóricas podem ser recategorizadas da mesma forma quando os valores possíveis que ela pode assumir forem muitos, diminuindo a complexidade das soluções finais e o super-ajuste (Saremi e Yaghmaee, 2014). Foi adotada a estratégia de limitar o número de percentis ou categorias a 4, ou seja,  $N \in \{2, 3, 4\}$ , escolhidos aleatoriamente segundo uma distribuição uniforme. Portanto, o conjunto de dados de treinamento  $\mathcal{X}$  é composto pelos pares  $(\tilde{\mathbf{x}}_i, c_i) \in \mathcal{A}_1 \times \dots \times \mathcal{A}_d \times \mathcal{C}$ , onde  $\mathcal{A}_k = \{a_1^k, \dots, a_{n_k}^k\}$  é o conjunto finito de valores  $n_k$  que o atributo  $k$ -th pode assumir.

No uso da PG para induzir árvores de decisão, os nós internos representam os atributos do conjunto de dados de treinamento, enquanto as folhas representam as classes. A função de teste de atributo  $\phi_k: \mathcal{A}_k \rightarrow \mathcal{Y} \subseteq \mathcal{X}$  particiona o conjunto de dados de treinamento de forma que cada partição contenha todas as instâncias de  $\mathcal{X}$  onde o atributo  $k$ -th é igual a um determinado valor  $a_j^k$

$$\phi_k(a_j^k) = \{(\tilde{\mathbf{x}}_i, c_i) \mid \tilde{x}_{i,k} = a_j^k, (\tilde{\mathbf{x}}_i, c_i) \in \mathcal{X}\},$$

onde  $\tilde{x}_{i,k}$  é o  $k$ -ésimo atributo do vetor  $\tilde{\mathbf{x}}_i$ . Quando uma instância precisa ser avaliada, a função na raiz da árvore testa o atributo correspondente, e se o argumento for um terminal, será retornada a decisão (classificação) para esta instância; caso contrário, um novo atributo será avaliado.

O Algoritmo 1 mostra o pseudo-código para a árvore de decisão induzida pela PG, onde os parâmetros de entrada são: o conjunto de dados de treinamento  $\mathcal{X}$ , o número máximo de gerações  $\zeta \in \mathbb{N}^*$ , o limite de simplificação  $\varepsilon \in [0, 1] \subset \mathbb{R}$  e a frequência de simplificação  $\tau \in \mathbb{N}^*$ , onde  $\mathbb{N}^*$  é o conjunto dos números naturais sem zero; e o parâmetro de saída é a melhor Árvore de Decisão (Árvore de Decisão\*). O método INICIALIZA (Linha 1) produz a população inicial com base no conjunto de dados de treinamento, conforme descrito na Seção 3.1; AVALIA (Linha 2 e Linha 11) mede a aptidão de cada indivíduo na população em relação aos dados de treinamento; o método RECOMBINA (Linha 6) constrói uma população filha usando a seleção pai do método SELECIONA (Linha 5); O método MUTAÇÃO (Linha 7) insere variabilidade na população de árvores, conforme descrito na Seção 3.3; a instrução RESTO (Linha 8) garante a periodicidade dos testes de simplificação a cada  $\tau$  gerações; o método SIMPLIFICAÇÃO (Linha 9) irá podar árvores com nós internos não expressivos, controlando seu crescimento, conforme descrito na Seção 3.5; finalmente,  $\varepsilon$  representa o limite predefinido para eliminar sub-árvores.

---

**Algoritmo 1** Programação Genética(Entrada:  $\mathcal{X}$ ,  $\zeta$ ,  $\varepsilon$ ,  $\tau$ ; Saída: Árvore de Decisão\*)

---

```

1:  $P \leftarrow$  INICIALIZA( $\mathcal{X}$ )
2:  $F_P \leftarrow$  AVALIA( $\mathcal{X}$ ,  $P$ )
3: gen  $\leftarrow$  1
4: while gen <  $\zeta$  do
5:    $P_S \leftarrow$  SELECIONA( $P$ ,  $F_P$ )
6:    $P_C \leftarrow$  RECOMBINA( $P_S$ )
7:    $P \leftarrow$  MUTAÇÃO( $P_C$ )
8:   if RESTO(gen,  $\tau$ ) = 0 then
9:      $P \leftarrow$  SIMPLIFICAÇÃO( $P$ ,  $\varepsilon$ )
10:  end if
11:   $F_P \leftarrow$  AVALIA( $\mathcal{X}$ ,  $P$ )
12:  gen  $\leftarrow$  gen + 1
13: end while

```

---

### 3.1. Geração da População Inicial

Considerando a tupla de todos os atributos no conjunto de dados  $\mathbf{A} = (\mathcal{A}_1, \dots, \mathcal{A}_d)$ , seja  $\mathbf{A}_p$  o conjunto de todas as  $d!$  permutações de  $\mathbf{A}$  onde  $\mathbf{A}_p(k)$  é a  $k$ -ésima permutação. A população inicial da árvore é construída escolhendo uma permutação aleatória  $\mathbf{A}^* \in \mathbf{A}_p$ , tal que

$$\mathbf{A}^* = \mathbf{A}_p(u), \quad \text{com } u \sim \mathcal{U}\{1, d!\}, \quad (1)$$

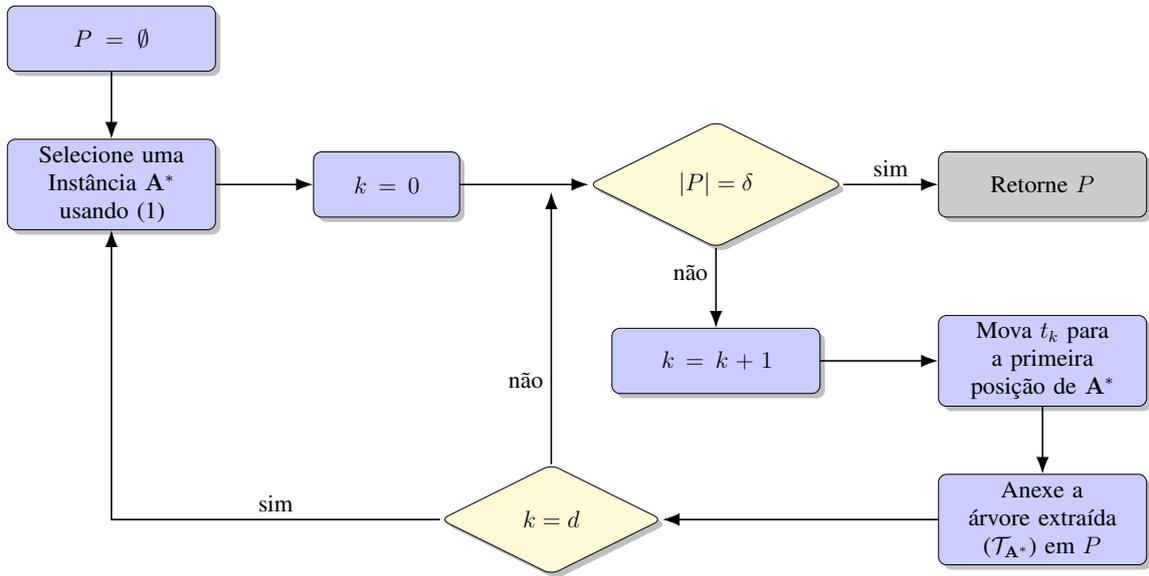
onde  $\mathcal{U}\{a, b\}$ , com  $b > a$ , é uma distribuição uniforme discreta com suporte  $s \in \{a, a + 1, \dots, b - 1, b\}$ . A tupla escolhida aleatoriamente em (1) pode ser usada para construir uma árvore  $\mathcal{T}_{\mathbf{A}^*} = (t_1, \dots, t_d)$  cujos nós  $t_n$  são elementos de  $\mathbf{A}$ . O primeiro elemento  $t_1$  é a raiz da árvore enquanto os filhos esquerdo e direito do elemento  $t_n$  são dados por:

$$t_{n,\text{left}} = \begin{cases} t_{2n}, & \text{se } 2n \leq d \\ \emptyset, & \text{caso contrário,} \end{cases} \quad (2a)$$

$$t_{n,\text{right}} = \begin{cases} t_{2n+1}, & \text{se } 2n + 1 \leq d \\ \emptyset, & \text{caso contrário.} \end{cases} \quad (2b)$$

As seguintes árvores na população inicial  $P$  (Linha 1 do Algoritmo 1) serão criadas movendo os elementos de  $\mathcal{T}_{\mathbf{A}^*}$  para a primeira posição (nó raiz) um por um. Após todas as rotações para criar  $d$  árvores diferentes de  $\mathcal{T}_{\mathbf{A}^*}$ , uma nova instância de  $\mathbf{A}^*$  será desenhada se o número máximo de árvores  $\delta \in \mathbb{N}^*$ , na população inicial, não é atingido. Este número máximo é definido empiricamente para cada conjunto de dados e o processo para criar a população inicial é descrito na Figura 5. A população inicial é o conjunto  $P = \{p(1), \dots, p(\delta)\}$ , onde  $p(k)$  é a representação em árvore de uma tupla  $\mathbf{A}^* \in \mathbf{A}_p$  cujo *fitness*  $f_p(k)$  compõe o vetor *fitness*  $F_P = \{f_p(1), \dots, f_p(\delta)\}$ .

Figura 5: Fluxograma da Geração da População Inicial.



Fonte: Autores.

### 3.2. Operador de Seleção

Para seleccionar quais pais serão responsáveis por criar a próxima geração, um torneio binário é feito. A ideia principal é executar múltiplas simulações para seleccionar pais com maiores valores

de aptidão e controlar a pressão seletiva. Seja  $\mathcal{I} = \{1, 2, \dots, \delta\}$  e  $\mathcal{I}_p$  o conjunto de todas as  $\delta!$  permutações de  $\mathcal{I}$  onde  $\mathcal{I}_p(k)$  é a  $k$ -ésima permutação. Uma permutação aleatória  $\mathcal{I}^* = \mathcal{I}_p(v)$  é escolhida, com  $v \sim \mathcal{U}\{1, \delta!\}$ , tal que

$$\mathcal{I}^* = (i_1^*, \dots, i_\delta^*), \quad (3)$$

onde  $i_k^* \in \mathcal{I}$ . A população selecionada é dada como

$$P_S = \{p_s(1), \dots, p_s(\delta_s)\}, \quad (4)$$

onde os indivíduos são escolhidos por um torneio binário, tal que

$$p_s(k) = \begin{cases} p(i_{2k-1}^*), & \text{se } f_p(i_{2k-1}^*) \geq f_p(i_{2k}^*) \\ p(i_{2k}^*), & \text{caso contrário.} \end{cases} \quad (5)$$

O número de indivíduos em  $P_S$  é dado por

$$\delta_s = \begin{cases} \frac{\delta}{2}, & \text{se } \delta \text{ é par} \\ \frac{\delta-1}{2}, & \text{caso contrário.} \end{cases} \quad (6)$$

### 3.3. Operadores de Variabilidade

O procedimento de recombinação combina os pais na população selecionada em pares. Cada par gera dois filhos que irão compor  $P_C$ , dados como

$$P_C = \{p_s(1), \dots, p_s(\delta_s), p_c^1(1), p_c^2(1), \dots, p_c^1(\delta_c), p_c^2(\delta_c)\}, \quad (7)$$

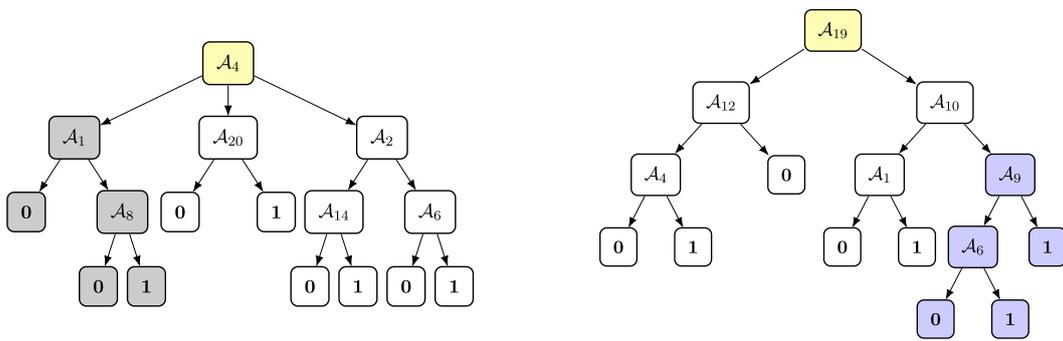
onde  $p_s(k)$  são indivíduos da população previamente selecionada e  $(p_c^1(k), p_c^2(k))$  são os filhos gerados pelo cruzamento de  $p_s(k)$  e  $p_s(r_\delta - k + 1)$  com  $1 \leq k < \frac{r_\delta + 1}{2}$ . O operador de cruzamento escolhe um nó aleatório em ambos os pais  $p_s(k)$  e  $p_s(r_\delta - k + 1)$ , exceto suas raízes; então, as subárvores, cujas raízes são os pontos de corte aleatórios, são trocadas, criando dois filhos:  $p_c^1(k)$  e  $p_c^2(k)$ . Um exemplo do procedimento é mostrado na sequência da Figura 6, o primeiro pai  $p_s(1) \in P_S$  é escolhido e um nó de atributo  $\mathcal{A}_1$  é selecionado para ser um ponto de corte Figura 6(a); na Figura 6(b), o primeiro pai  $p_s(\delta_s) \in P_S$  é escolhido e um nó de atributo  $\mathcal{A}_9$  é selecionado para ser um ponto de corte; Figura 6(c) e a Figura 6(d) mostram a troca de subárvores entre os pais criando dois filhos.

O operador de mutação é aplicado apenas a atributos previamente contínuos a uma taxa de  $\eta \in [0, 1] \subset \mathbb{R}$ , ou seja, o atributo sofrerá mutações com probabilidade  $\eta$ . Após decidir que uma mutação deve ocorrer, um atributo previamente contínuo será selecionado aleatoriamente para ter seus limites de discretização alterados; de acordo com Saremi e Yaghmaee (2018), este mecanismo de mutação é competitivo com outros em termos de simplicidade e eficiência. Por exemplo, suponha um nó que represente o atributo idade em anos e a população possua indivíduos cuja idade esteja no intervalo  $[0, 98]$ ; uma possível discretização desse atributo é criar três faixas categóricas, como  $\{[0, 25], [26, 57], [58, 98]\}$ . Então uma mutação neste atributo discretizado poderia deslocar a categoria do intervalo intermediário em 4 unidades como, por exemplo,  $\{[0, 25], [26, 61], [62, 98]\}$ , criando uma nova discretização. A nova população é atribuída a  $P$  como a população da nova geração.

### 3.4. Função de Aptidão (*Fitness*)

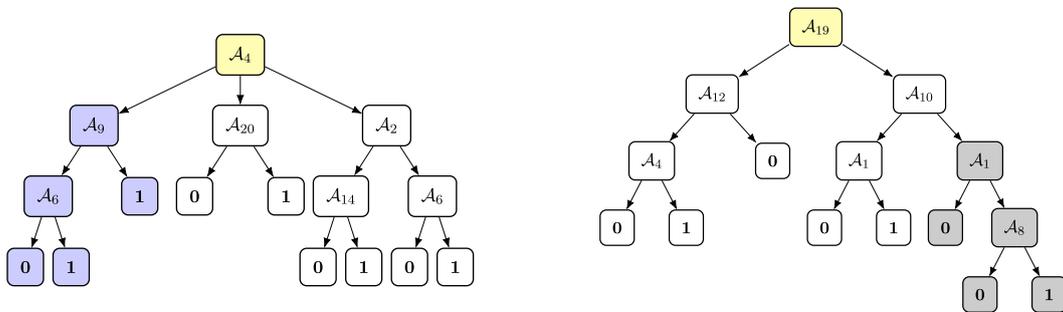
A função de aptidão avalia a qualidade de um determinado indivíduo em relação a um objetivo pré-definido. É usado no processo de treinamento para guiar a população em uma determinada

Figura 6: Exemplo do comportamento do operador de cruzamento aplicado a um par de pais  $p_s(1), p_s(\delta_s) \in P_S$  gerando dois filhos  $p_c^1(1)$  e  $p_c^2(1)$ .



(a) Selecione o pai  $p_s(1) \in P_S$  e o nó  $A_1$  para cortar.

(b) Selecione o pai  $p_s(r_\delta) \in P_S$  e o nó  $A_9$  para cortar.



(c) Filho  $p_c^1(1)$  do primeiro e do último pai.

(d) Filho  $p_c^2(1)$  do primeiro e do último pai.

Fonte: Autores.

direção, auxiliando no processo de treinamento (Devarriya et al., 2020). No contexto das tarefas de classificação, uma solução pode ser avaliada usando diferentes métricas, como precisão, *F1-score*,  $G_{\text{mean}}$ , *Cohen's kappa*, etc (Zhou et al., 2015). Embora a precisão e *Cohen's kappa* sejam métricas de avaliação muito comuns, em conjuntos de dados desbalanceados, eles podem ser tendenciosos em relação à classe majoritária, levando a resultados incorretos (Devarriya et al., 2020). Para conjuntos de dados desbalanceados, as métricas *F1-score* e  $G_{\text{mean}}$  minimizam o viés de precisão, no entanto, *F1-score* é preferível quando a classe minoritária é mais importante, enquanto  $G_{\text{mean}}$  é usado para maximizar a sensibilidade de ambas classes (Al-Badarneh et al., 2022). Assim, neste trabalho, os indivíduos são representados por árvores de decisão e a função de aptidão é a média geométrica entre especificidade (SPC) e sensibilidade (SEN), definida como:

$$\text{SPC} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \quad (8a)$$

$$\text{SEN} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (8b)$$

$$G_{\text{mean}} = \sqrt{\text{SPC} \cdot \text{SEN}}, \quad (8c)$$

onde TP, FP, TN e FN denotam, respectivamente, verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos. Essa métrica é utilizada devido a sua propriedade de independência à distribuição de exemplos de diferentes classes, obrigando o sistema de aprendizado a produzir classificações corretas em uma fração significativa dos exemplos positivos (Kubat et al., 1998).

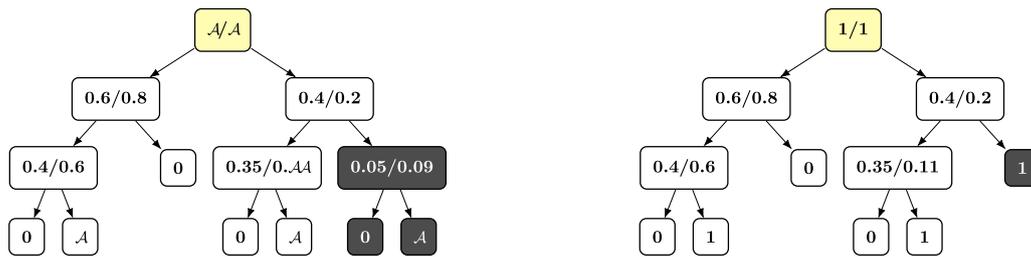
### 3.5. Operador de Simplificação

O método de inicialização cria árvores com uma quantidade fixa de nós internos; no entanto, o procedimento de recombinação pode fazer com que algumas árvores cresçam sem limites ao longo de novas gerações. À medida que o tamanho da árvore aumenta, os resultados são propensos a *overfitting* e podem sofrer uma diminuição em suas capacidades de interpretabilidade (DeLisle e Dixon, 2004). Doerr et al. (2019) mostra que sistemas baseados em PG para evoluir funções booleanas simples, formadas pela conjunção de algumas variáveis, exigirão um limite logarítmico no tamanho da árvore. Em Lissovoi e Oliveto (2019), recomenda-se o uso de estratégias para reduzir o crescimento da árvore usando limites pré-estabelecidos. Assim, um limiar ( $\varepsilon$ ) é definido para remover sub-árvores inexpressivas, definidas como as sub-árvores cujo nó raiz contém menos que  $\varepsilon\%$  das instâncias de treinamento de ambas as classes; cada nó armazena a quantidade de dados de cada classe que ele representa em relação a todo o conjunto de dados. Este procedimento é executado periodicamente com um período  $\tau$  definido empiricamente. Um exemplo deste procedimento considerando  $\varepsilon = 0.1$  é mostrado na Figura 7, onde a sub-árvore cinza na Figura 7(a) representa 5% da classe 0 e 9% da classe 1. Figura 7(b) mostra a substituição desta sub-árvore por um nó folha com a classe 1 mais representativa.

## 4. Resultados

Este estudo utiliza um conjunto de dados público disponível no repositório de aprendizado de máquina UCI (Biteus e Lindgren, 2017a). Os dados são divididos em conjunto de treinamento e conjunto de testes, contendo 60.000 e 16.000 instâncias, respectivamente. Desse total, 59.000 instâncias pertencem à classe negativa e 1.000 à classe positiva no conjunto de treinamento, enquanto no conjunto de testes 375 instâncias são de classe positiva e 15.625 são de classe negativa. Os dados foram obtidos do sistema APS de caminhões Scania, após ocorrer ou não uma falha do

Figura 7: Exemplo do operador de simplificação com  $\varepsilon = 0.1$ .



(a) Seleciona uma subárvore cujo nó raiz representa menos que  $\varepsilon\%$  dos dados de treinamento.

(b) Substitui a subárvore por um nó folha representado pela classe mais comum nela.

Fonte: Autores.

sistema. As duas classes presentes nos dados são classe positiva, que indica falhas relacionadas aos componentes do APS, e classe negativa, que indica falhas não relacionadas aos componentes do APS.

Como principal resultado, este trabalho tem o intuito de garantir uma previsão precisa de falhas relacionadas ao sistema APS, a fim de que se possa identificar o componente com problema rapidamente. Tal resultado é benéfico para o operador, motorista, mecânico e fabricante envolvidos na operação do veículo. Devido a sua conexão com diversos outros componentes, o APS pode dificultar a identificação direta do motivo da falha. Prever falhas relacionadas ao APS se torna um desafio, devido ao desbalanceamento inerente ao conjunto de dados. Além disso, a quantidade de valores ausentes preocupa - somente 2% das instâncias apresentam dados completos para todos os atributos. Assim, optou-se pela eliminação de 2,33% dos dados que não possuem qualquer valor no conjunto de dados de teste e treinamento, em ambas as classes positivas e negativas.

#### 4.1. Configuração Experimental

A abordagem proposta foi executada 32 vezes utilizando os dados de treinamento para calibrar os parâmetros do modelo e os dados de teste para avaliar os modelos treinados. Os resultados das 32 avaliações são apresentados na Tabela 3. Foram utilizadas 4 métricas para avaliar os modelos treinados: Especificidade (SPC), Sensibilidade (SEN) e G-média (Gmean) (mencionadas na Seção 3.4), e ainda, Acurácia (ACC) que é definida pela Equação 9:

$$ACC = \frac{VP + VN}{VP + FP + VN + FN} \quad (9)$$

Onde:  $VP$ ,  $FP$ ,  $VN$  e  $FN$  denotam, respectivamente, verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos.

#### 4.2. Análises e Resultados

Esta seção aborda a análise de métricas de predição para avaliação e classificação de dados em falhas mecânicas do APS. A Sensibilidade (SEN) se refere aos erros inerentes ao APS corretamente identificados como falhas mecânicas, enquanto a Especificidade (SPC) identifica elementos mecânicos sem problemas. A média obtida para a Sensibilidade foi de 0,8110, indicando que 81,10% dos dados classificam falhas inerentes ao APS, e para a Especificidade foi de 0,8939, ou seja, 89,39% dos dados sem falhas relacionadas ao APS como apresentado na Tabela 1. Na Tabela 1 ainda é possível avaliar a quantidade média de nós internos gerados pela abordagem (um

valor entre 6 a 7 nós). Esse valor é baixo o que mostra que a abordagem de indução por PG é capaz de gerar árvores com baixa complexidade e com boa interpretabilidade e isso justifica a utilização do mecanismo de simplificação.

Uma comparação com abordagens de estado da arte é apresentado na Tabela 2. A comparação é realizado com os modelos KNN e SVM descritos em Rafsunjani et al. (2019). Os resultados obtidos com os modelos do KNN e SVM apresentaram uma Acurácia maior, porém com menor Sensibilidade, de acordo com os valores de  $p$  na Tabela 2, computados com o uso do teste Kruskal Wallis. Esses resultados estão relacionados ao fato da PG utilizar o *G-Mean* como função de aptidão, o que faz o processo de evolução buscar por árvores que maximizam essa métrica. O mesmo não ocorre nos treinamentos dos modelos baseados em SVM ou KNN que têm como propósito maximizar a acurácia. Além disso, no contexto da evolução de soluções por meio de algoritmos evolutivos, como a Programação Genética (PG), é inerente a capacidade de escapar de ótimos locais, permitindo, dessa forma, uma busca abrangente em nível global. Importante destacar que, em problemas de predição em falhas mecânicas com desbalanceamento de classes, a Sensibilidade é a métrica mais relevante, visto que a classe de menor prevalência é o desfecho de maior interesse, pois uma falha mecânica pode causar acidentes graves. Contudo, é essencial mencionar que os erros de classificação, que resultam em valores mais baixos de acurácia, não devem ser negligenciados. Quando ocorre um falso negativo, significa que um componente com falha não foi identificado pelo modelo, o que pode levar à manutenção inadequada desse componente. Por outro lado, um falso positivo indica que um componente perfeitamente funcional foi erroneamente considerado defeituoso, resultando na sua substituição desnecessária. Essa situação pode aumentar significativamente os custos de manutenção do sistema. Portanto, é crucial considerar esses erros de classificação ao avaliar a eficácia do modelo de detecção de falhas (Rodrigues et al., 2023; Berrade et al., 2013).

Diante disso, a abordagem proposta apresentou bons resultados proporcionalmente considerando a Sensibilidade e Especificidade. Outro ponto importante que deve ser considerado, é a natureza interpretável dos modelos baseados em árvores de decisão (solução proposta neste estudo). A interpretabilidade é importante em modelos de previsão pois auxilia na compreensão do problema e na tomada de decisão. No problema abordado neste trabalho, modelos interpretáveis podem ajudar a identificar o componente causador da falha e isso possibilita uma manutenção mais eficaz. Nesse caso, as variáveis mais importantes na determinação da falha foram: 11, 39, 115, 155 e 156; isso significa que a implementação de um sistema automático de detecção de falhas no sistema APS precisa rastrear somente 5 variáveis e não todas, resultando em economia financeira, por exemplo na redução do número de sensores.

Tabela 1: Média e Desvio Padrão - Métricas (ACC, SEN, SPC, G-Media).

Med.	AC	SEN	SPC	GMean	Nós
Media	0,8118	0,8110	0,8939	0,8493	6,34
DEV PAD	0,0622	0,0654	0,0650	0,0273	1,38

Fonte: Autores.

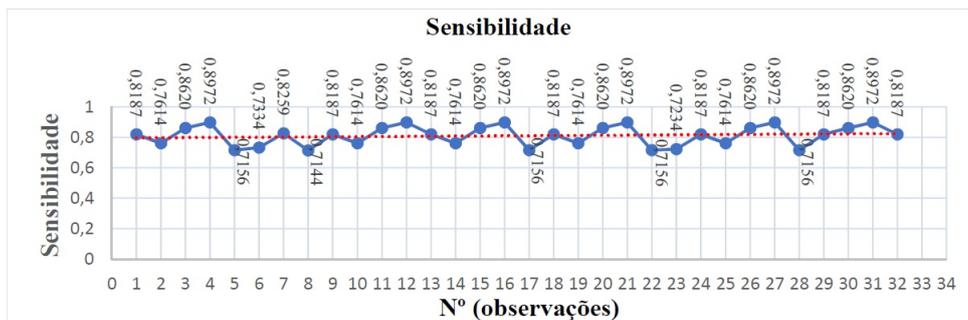
Tabela 2: Resultado por Classificadores (KNN, SVM, PG).

Class.	ACC	SEN	SPC	GMean
KNN	0,9794	0,1867	0,9984	0,4317
SVM	0,9843	0,4987	0,9960	0,7047
PG	0,8118	0,8110	0,8939	0,8493
p-valor	< 0.001	< 0.001	< 0.001	< 0.001

Fonte: Autores.

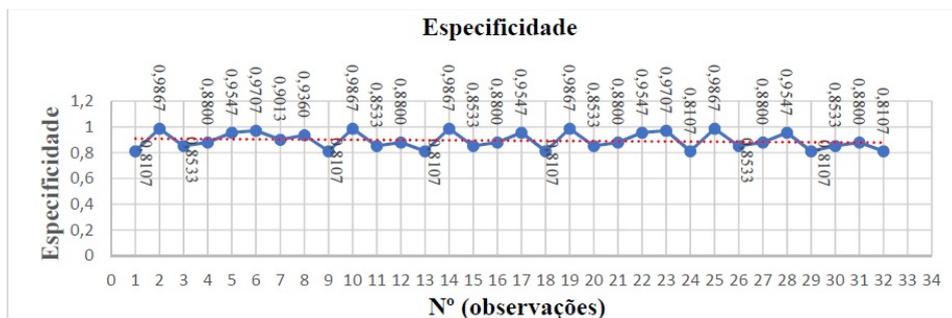
Os resultados obtidos a partir das 32 execuções da abordagem são mostrados nas Figuras 8 9 e 10, apresentando as métricas de sensibilidade, especificidade e acurácia, respectivamente. A Sensibilidade (SEN) (Figura 8) é a métrica responsável por identificar os resultados classificados como falhas, ou verdadeiros positivos (falhas reais do APS), em relação ao total de casos de falhas observados. Dessa forma, a linha de tendência foi elevada, permitindo uma melhor visualização dos resultados. Por outro lado, a Acurácia (ACC) (Figura 10) apresentou uma linha de tendência ascendente, permitindo uma avaliação mais completa do resultado, já que mede o total de acertos considerando o total de observações.

Figura 8: Sensibilidade.



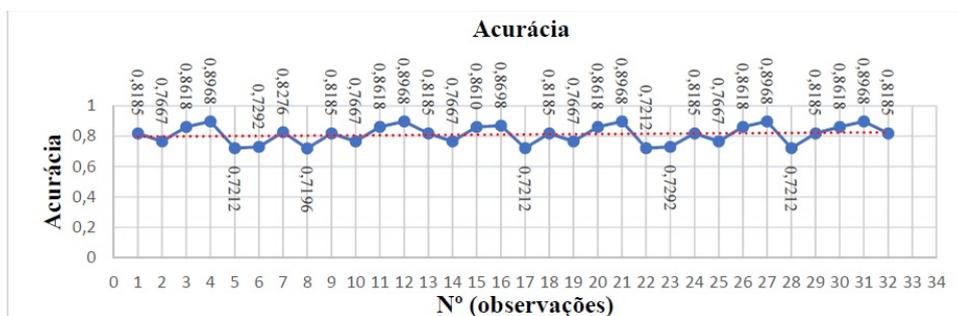
Fonte: Autores.

Figura 9: Especificidade.



Fonte: Autores.

Figura 10: Acurácia.



Fonte: Autores.

## 5. Conclusões

O objetivo principal deste trabalho consistiu em identificar corretamente se a causa da falha era relacionada ao APS ou não, levando em consideração que este sistema está vinculado a diversos outros componentes. Constatou-se que o conjunto de dados do APS é muito desequilibrado, com apenas 1.000 ocorrências de falhas do APS dentre as 60.000 instâncias presentes no conjunto de dados.

Os resultados obtidos através da análise de Sensibilidade (SEN) e Acurácia (ACC) indicam que há casos de falsos positivos de falhas relacionados ao APS. Apesar de os valores de acurácia obtidos neste estudo serem menores que os valores relatados em estudos anteriores obtidos pelos classificadores KNN e SVM, a média dos valores calculados para a sensibilidade foi maior. Isso significa que os valores de sensibilidade indicam maior precisão em identificar falhas relacionadas ao APS através da nossa abordagem do que através dos classificadores KNN e SVM.

No entanto, observou-se em trabalhos anteriores que os valores obtidos para especificidade (SPC) foram superiores aos obtidos neste estudo, o que indica um aumento na imprecisão das falhas não relacionadas ao APS. A métrica *G-Mean* minimiza este tipo de viés, maximizando a sensibilidade. Desta forma, os resultados apresentados comprovam que a abordagem adotada pela Árvore de Decisão induzida por Programação Genética foi eficaz para o estudo de classes desbalanceadas do sistema APS da Scania. Os resultados obtidos pela abordagem proposta podem ser aplicados em qualquer unidade APS de características semelhantes à unidade APS do presente estudo e em casos onde as características se diferem, existe a necessidade de retreinar o modelo.

Este trabalho apresenta bons resultados no que se refere à aplicação de uma técnica de Aprendizado de Máquina (Machine Learning - ML) na identificação de casos reais de falhas fornecidos pelo sistema APS-Scania. A capacidade de identificar essas falhas de maneira segura e consistente é de extrema importância no contexto do problema abordado. Os modelos matemáticos de detecção de falhas que foram propostos e implementados podem ser integrados diretamente ao sistema APS ou ao Conector de Diagnóstico (OBD). Isso permite que, durante a operação normal dos caminhões ou veículos pesados, as falhas sejam isoladas de forma eficiente. Como resultado, essa abordagem contribui para o aumento da vida útil dos veículos e a redução de acidentes relacionados ao sistema APS.

Em trabalhos futuros pode-se avaliar a inserção de estratégias de manutenção de diversidade de população de árvores por meio de técnicas de *niching* e/ou busca novidades, e ainda, a aplicação dessa abordagem em outros conjuntos de dados de falhas APS. Uma limitação significativa deste estudo é que nossa abordagem foi desenvolvida exclusivamente para problemas com duas classes. Portanto, é essencial avaliá-la em cenários envolvendo mais de duas classes.

**Agradecimentos.** Os autores agradecem o apoio da FAPEMIG, CAPES e CNPq.

## Referências

Al-Badarneh, I., Habib, M., Aljarah, I., e Faris, H. Neuro-evolutionary models for imbalanced classification problems. *Journal of King Saud University - Computer and Information Sciences*, v. , 2022. ISSN 1319-1578 <https://www.sciencedirect.com/science/article/pii/S1319157820305309>.

Banzhaf, W. *Genetic programming : an introduction on the automatic evolution of computer programs and its applications*. San Francisco, Calif. Heidelberg: Morgan Kaufmann Publishers Dpunkt-verlag. ISBN 978-1558605107, 1998.

Barros, R. C., Basgalupp, M. P., de Carvalho, A. C. P. L. F., e Freitas, A. A. A survey of evolutionary algorithms for decision-tree induction. *IEEE Transactions on Systems, Man, and Cybernetics, Part*

*C (Applications and Reviews)*, v. 42, n. 3, p. 291–312, 2012. ISSN 1558-2442<http://dx.doi.org/10.1109/TSMCC.2011.2157494>.

Berrade, M., Scarf, P., Cavalcante, C., e Dwight, R. Imperfect inspection and replacement of a system with a defective state: A cost and reliability analysis. *Reliability Engineering System Safety*, v. 120, p. 80–87, 2013. ISSN 0951-8320.

Biteus, J. e Lindgren, T. *APS Failure at Scania Trucks*, 2017a. UCI Machine Learning Repository. Disponível em: . DOI: <https://doi.org/10.24432/C51S51>. Acesso em: 20/05/2023.

Biteus, J. e Lindgren, T. Planning flexible maintenance for heavy trucks using machine learning models, constraint programming, and route optimization. *SAE International Journal of Materials and Manufacturing*, v. 10, n. 3, p. 306–315, 2017b.

DeLisle, R. K. e Dixon, S. L. Induction of decision trees via evolutionary programming. *Journal of chemical information and computer sciences*, v. 44, n. 3, p. 862–870, 2004. <https://doi.org/10.1021/ci034188s>.

Devarriya, D., Gulati, C., Mansharamani, V., Sakalle, A., e Bhardwaj, A. Unbalanced breast cancer data classification using novel fitness functions in genetic programming. *Expert Systems with Applications*, v. 140, p. 112866, 2020. <https://doi.org/10.1016/j.eswa.2019.112866>.

Doerr, B., Lissovoi, A., e Oliveto, P. S. Evolving boolean functions with conjunctions and disjunctions via genetic programming. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. In: *Proceedings of the Genetic and Evolutionary Computation Conference*, 2019. p. 1003–1011.

Gondek, C., Hafner, D., e Sampson, O. R. Prediction of failures in the air pressure system of scania trucks using a random forest and feature engineering. In: *Advances in Intelligent Data Analysis XV: 15th International Symposium, IDA 2016, Stockholm, Sweden, October 13-15, 2016, Proceedings 15*. In: *Advances in Intelligent Data Analysis XV: 15th International Symposium, IDA 2016, Stockholm, Sweden, October 13-15, 2016, Proceedings 15*. Springer, 2016. p. 398–402.

Kubat, M., Holte, R. C., e Matwin, S. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, v. 30, n. 2-3, p. 195–215, 1998. <http://dblp.uni-trier.de/db/journals/ml/ml30.html#KubatHM98>.

Lissovoi, A. e Oliveto, P. S. On the time and space complexity of genetic programming for evolving boolean conjunctions. *Journal of Artificial Intelligence Research*, v. 66, p. 655–689, 2019. <https://doi.org/10.1613/jair.1.11821>.

Ozan, E. C., Riabchenko, E., Kiranyaz, S., e Gabbouj, M. An optimized k-nn approach for classification on imbalanced datasets with missing data. In: *Advances in Intelligent Data Analysis XV: 15th International Symposium, IDA 2016, Stockholm, Sweden, October 13-15, 2016, Proceedings 15*. In: *Advances in Intelligent Data Analysis XV: 15th International Symposium, IDA 2016, Stockholm, Sweden, October 13-15, 2016, Proceedings 15*. Springer, 2016. p. 387–392.

Quinlan, J. R. Simplifying decision trees. *International journal of man-machine studies*, v. 27, n. 3, p. 221–234, 1987.

Rafsunjani, S., Safa, R. S., Imran, A. A., Rahim, S., e Nandi, D. An empirical comparison of missing value imputation techniques on aps failure prediction. *International Journal of Information Technology and Computer Science*, v. p. 21–29, 2019. <https://j.mecs-press.net/ijitcs/ijitcs-v11-n2/IJITCS-V11-N2-3.pdf>.

Rodrigues, A. J. d. S., Cavalcante, C. A. V., e Alberti, A. R. A multicriteria model to support the selection of inspection service providers based on the delay time model. *International Transactions in Operational Research*, v. 30, n. 6, p. 3554–3577, 2023. ISSN 1475-3995<http://dx.doi.org/10.1111/itor.13274>.

Santos, L. I., Camargos, M. O., D'Angelo, M. F. S. V., Mendes, J. B., de Medeiros, E. E. C., Guimarães, A. L. S., e Palhares, R. M. Decision tree and artificial immune systems for stroke prediction in imbalanced data. *Expert Systems with Applications*, v. 191, p. 116221, 2022. ISSN 0957-4174<https://www.sciencedirect.com/science/article/pii/S0957417421015347>.

Saremi, M. e Yaghmaee, F. Evolutionary decision tree induction with multi-interval discretization. In: *2014 Iranian Conference on Intelligent Systems (ICIS)*. In: *2014 Iranian Conference on Intelligent Systems (ICIS)*. IEEE, 2014. p. 1–6.

Saremi, M. e Yaghmaee, F. Improving evolutionary decision tree induction with multi-interval discretization. *Computational Intelligence*, v. 34, n. 2, p. 495–514, 2018. <https://doi.org/10.1111/coin.12153>.

Scania. *Scania CVAB*, 2017. Scania. Disponível em: . <https://www.scania.com/br/pt/home.html>. Acesso em: 20/05/2023.

Syed, M. N., Hassan, M. R., Ahmad, I., Hassan, M. M., e De Albuquerque, V. H. C. A novel linear classifier for class imbalance data arising in failure-prone air pressure systems. *IEEE Access*, v. 9, p. 4211–4222, 2021.

Zhao, H. A multi-objective genetic programming approach to developing pareto optimal decision trees. *Decision Support Systems*, v. 43, n. 3, p. 809–826, 2007. <https://doi.org/10.1016/j.dss.2006.12.011>.

Zhou, J., Li, X., e Mitri, H. S. Comparative performance of six supervised learning methods for the development of models of hard rock pillar stability prediction. *Natural Hazards*, v. 79, n. 1, p. 291–316, 2015. <https://doi.org/10.1007/s11069-015-1842-3>.

## Apêndices

Tabela 3: Resultados das Métricas (ACC, SEN, SPC, GMean) para as 32 avaliações realizadas.

N	AC	SEN	SPC	GMean	Nós
1	0,8185	0,8187	0,8107	0,8147	6
2	0,7667	0,7614	0,9867	0,8668	8
3	0,8618	0,8620	0,8533	0,8576	4
4	0,8968	0,8972	0,8800	0,8885	7
5	0,7212	0,7156	0,9547	0,8266	6
6	0,7292	0,7334	0,9707	0,8380	8
7	0,8276	0,8259	0,9013	0,8628	8
8	0,7196	0,7144	0,9360	0,8177	7
9	0,8185	0,8187	0,8107	0,8147	6
10	0,7667	0,7614	0,9867	0,8668	8
11	0,8618	0,8620	0,8533	0,8576	4
12	0,8968	0,8972	0,8800	0,8885	7
13	0,8185	0,8187	0,8107	0,8147	6
14	0,7667	0,7614	0,9867	0,8668	8
15	0,8610	0,8620	0,8533	0,8576	4
16	0,8698	0,8972	0,8800	0,8885	7
17	0,7212	0,7156	0,9547	0,8266	6
18	0,8185	0,8187	0,8107	0,8147	6
19	0,7667	0,7614	0,9867	0,8668	8
20	0,8618	0,8620	0,8533	0,8576	4
21	0,8968	0,8972	0,8800	0,8885	7
22	0,7212	0,7156	0,9547	0,8266	6
23	0,7292	0,7234	0,9707	0,8380	8
24	0,8185	0,8187	0,8107	0,8147	6
25	0,7667	0,7614	0,9867	0,8668	8
26	0,8618	0,8620	0,8533	0,8576	4
27	0,8968	0,8972	0,8800	0,8885	7
28	0,7212	0,7156	0,9547	0,8266	6
29	0,8185	0,8187	0,8107	0,8147	6
30	0,8618	0,8620	0,8533	0,8576	4
31	0,8968	0,8972	0,8800	0,8885	7
32	0,8185	0,8187	0,8107	0,8147	6

Fonte: Autores.