

## Introdução

No cenário competitivo atual, otimizar a eficiência dos equipamentos é fundamental para garantir uma produção contínua e manter a competitividade das empresas. Interrupções inesperadas podem gerar inatividade e comprometer a qualidade dos produtos. A manutenção preditiva, baseada na análise exploratória de dados e extração de características, é uma abordagem promissora para identificar padrões e anomalias que indicam falhas. Tecnologias como IoT e Machine Learning viabilizam essas estratégias, otimizando a performance dos equipamentos, prevenindo falhas e assegurando a continuidade e eficiência da produção.

## Objetivo

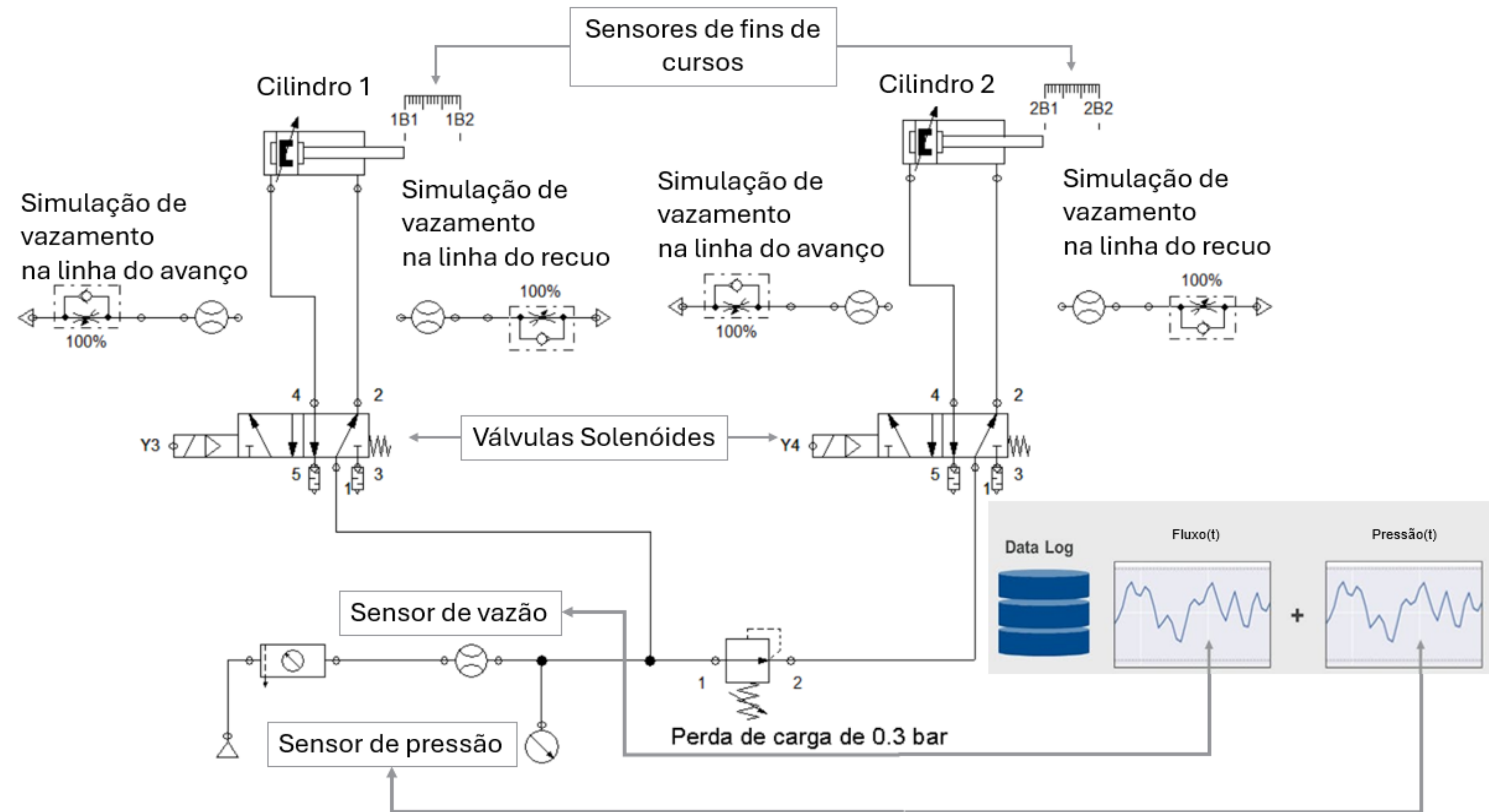
O principal objetivo deste estudo é realizar uma análise exploratória de dados e a extração de características para aprimorar a compreensão das condições operacionais dos equipamentos industriais, visando a implementação de estratégias de manutenção preditiva eficazes. Este estudo busca identificar padrões e anomalias nos dados coletados, facilitando a construção de modelos preditivos robustos que possam antecipar falhas e otimizar a performance dos sistemas. Os objetivos específicos incluem a coleta e organização dos dados, visualização dos dados, extração de características, desenvolvimento de modelos preditivos e avaliação da eficiência dos modelos.

## Fundamento Teórico

As técnicas de aprendizado de máquina são cruciais para a manutenção preditiva, facilitando a identificação de padrões e a previsão de necessidades futuras de manutenção, como destacado por [1] e [2]. A análise exploratória de dados é vital para compreender as condições operacionais dos equipamentos, destacando tendências, outliers e anomalias [2] e utilizando métodos avançados como a transformada de Fourier e análise de Wavelet [1]. No presente trabalho, consideramos que o processo de manutenção preditiva envolve monitoramento contínuo através de sensores que registram vazão e pressão, permitindo o uso de classificadores como Árvore de Decisão (DT), Random Forest (RF) e Perceptron de Múltiplas Camadas (MLP) para prever falhas de forma eficiente. Essas técnicas permitem que equipes de manutenção atuem proativamente, otimizando recursos e minimizando inatividades, apoiadas por métodos consolidados na literatura [2], [3], [4], [5], [6], [7], [8].

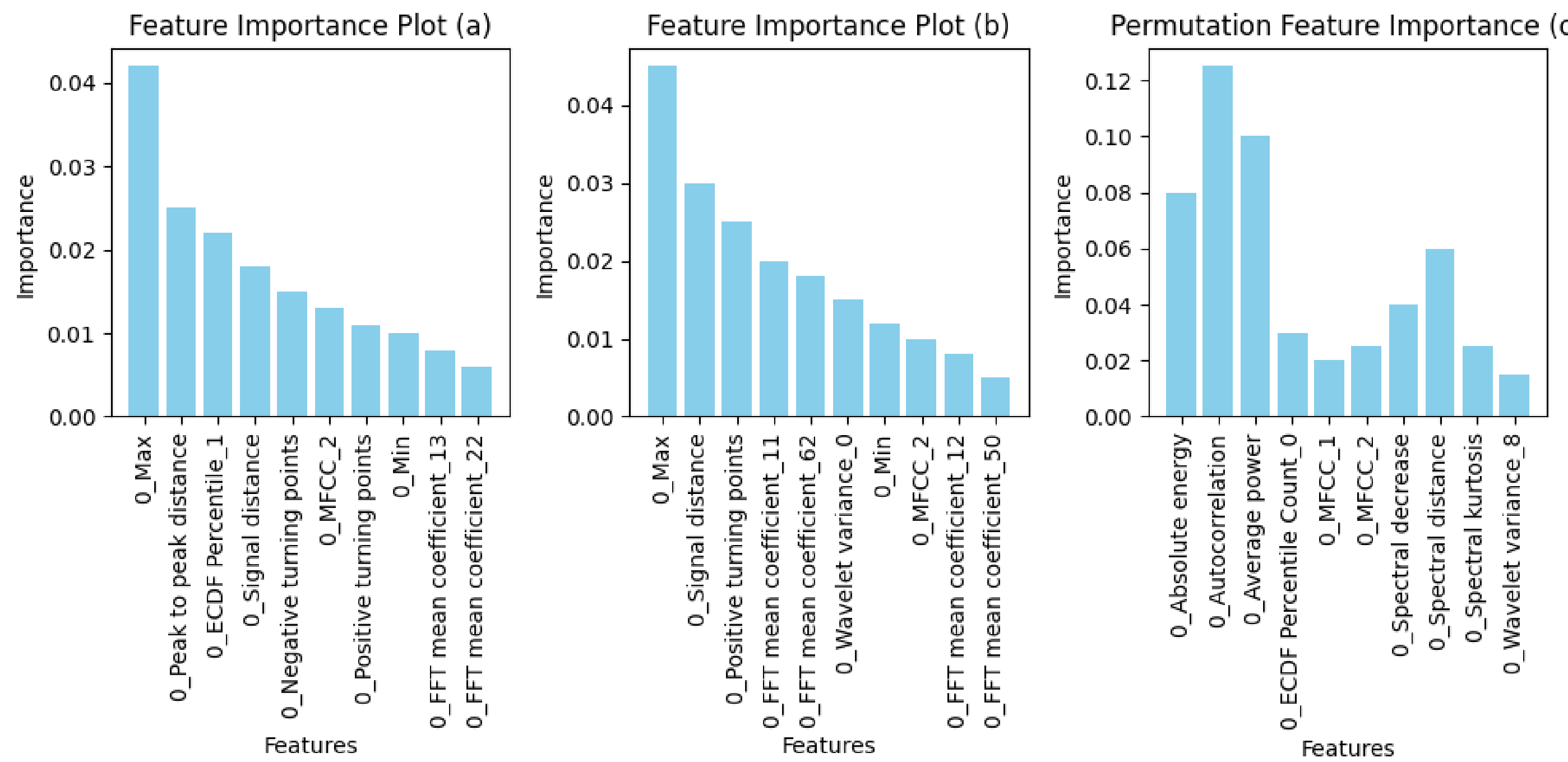
## Desenvolvimento

Este estudo desenvolve um sistema de predição com dados simulados gerados pelo FluidSIM, um software renomado da Art-Systems (2024) para simulação de sistemas pneumáticos, eletropneumáticos, hidráulicos e automação industrial. O FluidSIM permite a construção, simulação e análise de circuitos com um editor intuitivo e recursos visuais detalhados. A Figura 1 ilustra o sistema estudado, destacando sua aplicação prática.



**Figura 1** - Circuito Eletropneumático construído no FluidSIM. Fonte: Elaborado pelo autor (2024).

O sistema simulado permitiu a coleta de dados de pressão (em bar) e vazão (em l/min) em uma faixa de pressão entre 5,4 e 6,6 bar, comumente utilizado em sistemas pneumáticos [10], esta variação é devido a característica do sistema ser ideal e sem perdas. Para desenvolver o modelo preditivo, utilizou-se a biblioteca TSFEL [11] para a extração de 200 características nos domínios temporais, estatísticos e espectrais dos dados coletados. O objetivo foi classificar dados em sete categorias: Funcionamento Normal, Vazamento no Avanço/Recuo da Válvula 1, Vazamento Interno do Cilindro 1, Vazamento no Avanço/Recuo da Válvula 2, e Vazamento Interno do Cilindro 2. Os classificadores foram implementados com a biblioteca PyCaret [12], utilizando 80% dos dados para treinamento e 20% para teste. A técnica [13] da biblioteca scikit-learn, foi empregada para otimizar hiperparâmetros. Após o treinamento, a função evaluate\_model facilita a análise do desempenho do modelo. A técnica de feature importance [14], Figura 2(a) e Figura 2(b), foi aplicada para identificar as características mais influentes, e a permutation importance, Figura 2(c), da scikit-learn foi utilizada para a rede neural MLP. As características extraídas ajudam a analisar e classificar defeitos comuns em sistemas eletropneumáticos. As características que foram vistas de forma expressiva na Figura 2 estão apresentadas na Tabela 1.



**Figura 2** - Feature Importance: (a) Árvore de Decisão; (b) Random Forest; (c) MLP. Fonte: Elaborado pelo autor (2024).

**Tabela 1 – FEATURE IMPORTANCE X POSSÍVEIS DEFEITOS.**

Feature	Descritivo da Importância	Possíveis Defeitos
0_Max	Indicador crucial de picos de pressão ou vazão que podem sinalizar condições críticas no sistema.	Sobrecarga, vazamentos, obstrução ou resistência no sistema.
0_Autocorrelation	Fundamental para identificar padrões repetitivos e anomalias cíclicas que podem levar a falhas.	Falhas cíclicas ou persistentes.
0_Average power	Essencial para avaliar o consumo energético e a eficiência, indicando possíveis sobrecargas.	Desgaste, sobrecarga, consumo energético e eficiência do sistema.
0_Peak to peak distance	Importante para detectar flutuações na pressão ou fluxo, sugerindo instabilidades operacionais.	Instabilidade de controle, vibrações excessivas, problemas de controle de pressão.

Fonte: Elaborado pelo autor (2024)

## Resultados

Os resultados foram divididos em duas fases: treinamento e teste. Avaliou-se o desempenho dos modelos Árvore de Decisão, Random Forest e Multi-Layer Perceptron (MLP) utilizando métricas como acurácia, AUC, revocação, precisão, F1-Score e tempo de execução. Essas métricas são essenciais para analisar a eficácia dos modelos na previsão de falhas e manutenção de equipamentos. As tabelas detalham o desempenho de cada modelo, considerando tanto todas as características quanto as 10 mais importantes (Tabela 2.), destacando a superioridade dos classificadores Random Forest e Árvore de Decisão.

**Tabela 2 – MÉTRICAS DAS 10 CARACTERÍSTICAS MAIS IMPORTANTES DE CADA MODELO**

TREINO: MÉTRICAS DE DESEMPENHO E TEMPO						
Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Tempo (s)
Decision Tree Classifier	98,61%	0,9998	98,61%	98,72%	98,60%	5,19
Random Forest Classifier	98,61%	0,9998	98,61%	98,61%	98,61%	20,81
MLP Classifier	23,34%	0,5820	23,34%	37,83%	17,20%	21,84
TESTE: MÉTRICAS DE DESEMPENHO E TEMPO						
Modelo	Acurácia	AUC	Revocação	Precisão	F1-Score	Tempo (s)
Decision Tree Classifier	92,30%	0,9563	92,30%	92,31%	92,24%	0,18
Random Forest Classifier	94,06%	0,9952	94,06%	94,09%	94,02%	1,33
MLP Classifier	23,06%	0,5793	23,06%	31,40%	16,62%	1,37

Fonte: Elaborado pelo autor (2024)

## Conclusão

Os resultados deste estudo indicam que os modelos Árvore de Decisão e Random Forest superaram o MLP em todas as métricas de avaliação. Ambos alcançaram 96,83% de acurácia no treino e 90,88% e 92,55%, respectivamente, no teste, enquanto o MLP teve desempenho inferior, com 34,00% no treino e 33,16% no teste. A análise de 200 características foi essencial para identificar padrões e anomalias. Esses resultados destacam a importância de escolher o modelo adequado, equilibrando precisão e eficiência computacional para otimizar a performance dos equipamentos e manter a competitividade.

## Bibliografia

- Di Maggio, L. G. "Intelligent Fault Diagnosis of Industrial Bearings Using Transfer Learning and CNNs Pre-Trained for Audio Classification." Sensors 23(1): 211. 2022. <https://doi.org/10.3390/s23010211>.
- Li, X., W. Zhang, e Q. Ding. "Multi-Stage Convolutional Neural Network for Automatic Bearing Fault Diagnosis." IEEE Transactions on Industrial Informatics 18(6): 3767-3776. 2022. <https://doi.org/10.1109/TII.2021.3091658>.
- Zhao, R., R. Yan, J. Wang, K. Mao, F. Shen, e X. Wang. "Machine Learning-Based Autonomous Diagnosis of Rotating Machinery Faults: A Comprehensive Review and Future Research Direction." Journal of Manufacturing Systems 52: 227-243. 2022. <https://doi.org/10.1016/j.jmsy.2019.06.012>.
- Fernandes, M., Corchado, J.M., e Marreiros, G., "Machine learning techniques applied to mechanical fault diagnosis and fault prognosis in the context of real industrial manufacturing use-cases: a systematic literature review". Applied Intelligence. Vol. 52, Issue 12, pp.14246-14280, 2022. <https://doi.org/10.1007/s10489-022-03344-3>.
- Sharma, J., Mittal, M. L., e Soni, G., "Condition-based maintenance using machine learning and role of interpretability: a review". International Journal of System Assurance Engineering and Management. Vol.15, Issue 4, pp.1345-1360, 2022. <https://doi.org/10.1007/s13198-022-01843-7>.
- Assim, M., Obeidat, Q., Hammad, M. "Software Defects Prediction using Machine Learning Algorithms", 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), Sakheer, Bahrain, pp. 1-6, 2020.
- Lee, S.M., Lee, D. e Kim, Y.S. "The quality management ecosystem for predictive maintenance in the Industry 4.0 era". Int J Qual Innov 5, Springer Nature, 2019. Disponível em: <https://doi.org/10.1186/s40887-019-0029-5>.
- Wang, J. e Wang, Q. "Intelligent explicit model predictive control based on machine learning for microbial desalination cells", SAGE Publishing, 2018. Disponível em: <https://doi.org/10.1177/0959651818816845>.
- Art-Systems. FluidSIM. 2024. Disponível em: Art-Systems FluidSIM.
- Foit, K., Banas, W. e Cwikla, G. "The pneumatic and electropneumatic systems in the context of 4th industrial revolution", IOP Conference Series: Materials Science and Engineering, ModTech, 2018.
- Barandas, M., Folgado, D., Fernandes, L., Santos, S., Abreu, M., Bota, P., Liu, H., Schultz, T. e Gamboa, H., TSFEL: Time Series Feature Extraction Library, SoftwareX, 2020. Disponível em: <https://doi.org/10.1016/j.softx.2020.100387>.
- Ali, M., PyCaret: An open source, low-code machine learning library in Python, 2020. Disponível em: <https://pycaret.gitbook.io/docs>.
- GridSearchCV. Disponível em: [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.GridSearchCV.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html). Acesso em 05 de Jul. de 2024.
- Brownlee, J., How to Calculate Feature Importance With Python, 2020. Disponível em: <https://machinelearningmastery.com/calculate-feature-importance-with-python/>. Acesso em 05 de Jul. de 2024.