



Escola de Minas
CECAU - Colegiado do Curso de
Engenharia de Controle e Automação



Arthur Pereira Vilela

**Modelo Preditivo para Otimização de Operações em Equipamentos
Pesados utilizando uma Abordagem de aprendizado de máquina com
Dados da Rede CAN**

Ouro Preto, 2025

Arthur Pereira Vilela

Modelo Preditivo para Otimização de Operações em Equipamentos Pesados utilizando uma Abordagem de aprendizado de máquina com Dados da Rede CAN

Trabalho apresentado ao Colegiado do Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro(a) de Controle e Automação.
Orientadora: Profa. Dra. Adrielle de Carvalho Santana .

Ouro Preto

2025



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E
AUTOMACAO



FOLHA DE APROVAÇÃO

Arthur Pereira Vilela

Modelo Preditivo para Otimização de Operações em Equipamentos Pesados utilizando uma Abordagem de Aprendizado de Máquina com Dados da Rede CAN

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia de Controle e Automação

Aprovada em 25 de novembro de 2025

Membros da banca

Dra. Adrielle de Carvalho Santana - Orientadora - DECAT - Universidade Federal de Ouro Preto
Dr. Agnaldo José da Rocha Reis - Convidado - DECAT - Universidade Federal de Ouro Preto
Dr. Jadson Castro Gertrudes - Convidado - DECOM - Universidade Federal de Ouro Preto

Adrielle de Carvalho Santana, orientadora do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 17/12/2025



Documento assinado eletronicamente por **Adrielle de Carvalho Santana, PROFESSOR DE MAGISTERIO SUPERIOR**, em 17/12/2025, às 16:31, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1020517** e o código CRC **9F02C219**.

Agradecimentos

Agradeço primeiramente a Deus, pela força e sabedoria concedidas em todos os momentos desta jornada.

À minha família, pelo apoio incondicional e amor que sempre me sustentaram. Em especial, ao meu pai Claver, por ser meu maior exemplo de dedicação e que esteve comigo durante toda essa jornada, e à minha mãe Lilian, por seu carinho, paciência, incentivo constante e que nunca me deixou desistir em nenhum momento. Ao meu irmão Claver, pela amizade, companheirismo e pelas palavras de encorajamento nos momentos mais desafiadores.

À minha orientadora, professora Adrielle, pela orientação competente, pela confiança no desenvolvimento deste trabalho e pela contribuição essencial em todas as etapas do projeto. Sua dedicação ao ensino e à pesquisa foram uma inspiração constante.

À Universidade Federal de Ouro Preto, pela excelência no ensino e pela formação técnica e humana que oferece. A vivência acadêmica nesta instituição foi fundamental para meu crescimento pessoal e profissional.

Agradeço também à empresa Fast2Mine, pela oportunidade de aprendizado e pela disponibilização dos dados que tornaram este estudo possível. O ambiente de inovação e a confiança depositada no meu trabalho foram decisivos para o desenvolvimento deste projeto.

Por fim, expresso minha gratidão a todos os amigos, colegas e professores que, de alguma forma, contribuíram direta ou indiretamente para a realização deste trabalho. Cada gesto de apoio e incentivo foi essencial para a conclusão desta etapa tão importante da minha trajetória.

Resumo

A manutenção preditiva visa antecipar falhas e otimizar os processos de manutenção, reduzindo custos operacionais e aumentando a eficiência dos equipamentos, no entanto, um modelo de manutenção preditiva robusto, precisa de um conjunto de informações adicionais importantes e confiáveis sobre o equipamento. Assim, proponho desenvolver um modelo capaz de prever o estresse de um caminhão basculante, em tempo real, utilizando técnicas de aprendizado de máquina, separar os operadores em grupos de direção leve, moderada e severa, baseados em sua eficiência para a saúde do equipamento e gerar *Key Performance Indicators* (KPIs) valiosos que poderão servir como base para um modelo de manutenção preditiva. O desenvolvimento todo foi baseado em dados extraídos da rede *Controller Area Network* (CAN) de um caminhão Volvo FMX 500, ano 2023, de uma empresa do setor de mineração. Para isso, foi adotado o uso de dados como RPM, temperatura do motor, consumo de combustível e horímetro, coletados em tempo real pela rede CAN. O método utilizado envolve a coleta, pré-processamento e análise desses dados, utilizando algoritmos de aprendizado de máquina, como *MLP Regressor* e o *K-means*. A validação do modelo foi realizada com base em dados históricos e o desempenho foi avaliado por meio de métricas como precisão, *recall* e acurácia. Os resultados indicam que o modelo desenvolvido consegue, em tempo real, prever o esforço baseado em uma grandeza heurística que o equipamento está realizando com alta precisão e classificar os operadores em grupos por seus tipos de condução do equipamento, proporcionando uma solução eficiente para a gestão de manutenção e tomada de decisões para equipamentos pesados. As conclusões apontam para a viabilidade da implementação para um trabalho de manutenção preditiva, mediante a possibilidade de acesso a mais dados, com benefícios significativos para a indústria, especialmente em termos de redução de custos e aumento da vida útil dos equipamentos.

Palavras-chaves: Manutenção preditiva, KPIs, Rede CAN, Aprendizado de máquina, Eficiência operacional.

Abstract

Predictive maintenance aims to anticipate failures and optimize maintenance processes, reducing operational costs and increasing equipment efficiency. However, a robust predictive maintenance model requires a set of important and reliable additional information about the equipment. Therefore, I propose developing a model capable of predicting the stress of a dump truck in real time, using machine learning techniques, separating operators into light, moderate, and severe driving groups based on their efficiency for the health of the equipment, and generating valuable Key Performance Indicators (KPIs) that could serve as the basis for a predictive maintenance model. The entire development was based on data extracted from the Controller Area Network (CAN) of a 2023 Volvo FMX 500 truck from a mining sector company. For this, data such as RPM, engine temperature, fuel consumption, and hour meter were collected in real-time via the CAN network. The method used involves collecting, pre-processing, and analyzing these data, using machine learning algorithms such as MLP Regressor and K-means. The model's validation was carried out based on historical data, and performance was evaluated through metrics such as precision, recall, and accuracy. The results indicate that the developed model can, in real time, predict the effort based on a heuristic magnitude that the equipment is performing with high accuracy and classify operators into groups by their driving types, providing an efficient solution for maintenance management and decision-making for heavy equipment. The conclusions point to the feasibility of implementation for a predictive maintenance work, given the possibility of accessing more data, with significant benefits for the industry, especially in terms of cost reduction and increased equipment lifespan.

Key-words: predictive maintenance, CAN network, Machine learning, Operational efficiency.

Lista de Figuras

Figura 1 – Arquitetura da norma ISO 11898 em camadas	22
Figura 2 – CAN padrão: identificador de 11 bits	23
Figura 3 – CAN estendido: identificador de 29 bits	24
Figura 4 – Tráfego de informações na CAN	26
Figura 5 – dados limpos	39
Figura 6 – ESI real vs. ESI predito com médias	51
Figura 7 – Porcentagem de registros de cada cluster	54
Figura 8 – Gráfico de distribuição do registro de cluster e ESI	55

Lista de tabelas

Tabela 1	–	Descrição das features geradas para análise dos sinais	41
Tabela 2	–	Resultados comparativos das arquiteturas MLP testadas.	42
Tabela 3	–	Operadores com maiores e menores valores médios de ESI	52
Tabela 4	–	Descrição dos clusters operacionais identificados pelo algoritmo K-Means	54

Lista de abreviaturas e siglas

AI	<i>Artificial Intelligence</i>
API	<i>Application Programming Interface</i>
CAN	<i>Controller Area Network</i>
CPU	<i>Central Processing Unit</i>
CSV	<i>Comma-Separated Values</i>
CSV	<i>Comma-Separated Values</i>
DB	<i>Database</i>
ESI	<i>Engine Stress Index</i>
IoT	<i>Internet of Things</i>
K-Means	<i>K-Means Clustering Algorithm</i>
KPI	<i>key Performance indicators</i>
ML	<i>Machine Learning</i>
MLflow	<i>Machine Learning Flow</i>
MLP	<i>Multi-Layer Perceptron</i>
MLOps	<i>Machine Learning Operations</i>
MSE	<i>Mean Squared Error</i>
PKL	<i>Pickle File Format (Python Object Serialization)</i>
RMSE	<i>Root Mean Square Error</i>
RPM	Rotações por Minuto

Sumário

1	INTRODUÇÃO	11
1.1	Contextualização do Tema e Problema de Pesquisa	11
1.2	Objetivo Geral	12
1.3	Objetivos Específicos	13
1.4	Justificativa	13
1.5	Estrutura do Trabalho	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Manutenção	15
2.1.1	Manutenção corretiva	15
2.1.2	Manutenção preventiva	17
2.1.3	Manutenção por condição	19
2.1.4	Manutenção preditiva	19
2.2	Rede CAN	20
2.2.1	Arquitetura e Camadas do Padrão ISO 11898	21
2.2.2	Estrutura de Mensagens e Identificadores	23
2.2.3	Mecanismos de Comunicação e Controle de Acesso	25
2.2.4	Mecanismos de Detecção e Correção de Erros	26
2.2.5	Aplicações	27
2.3	Métodos e Técnicas	28
2.4	Telemetria	29
2.5	Modelo Multilayer Perceptron Regressor	30
2.6	Algoritmo K-means	32
2.7	Trabalhos relacionados	33
3	METODOLOGIA	36
3.1	Metodologia proposta	36
3.1.1	Coleta e pré-processamento de dados	36
3.1.2	Geração de <i>Features</i> Derivadas	37
3.1.3	Desenvolvimento e Treinamento de Modelos de IA	37
3.1.4	Cálculo de Métricas Derivadas e KPIs	37
3.1.5	Validação e Análise de Resultados	38
3.2	Dados	38
3.2.1	Fonte	38
3.2.2	Pré-processamento de dados	38
3.3	ESI heurístico	39

3.4	Geração de recursos	40
3.4.1	Recursos estatísticas derivados	40
3.5	Modelagem	41
3.5.1	<i>Multi-Layer Perceptron Regressor</i>	41
3.5.2	<i>K-Means</i>	43
3.6	Pipeline	44
3.6.1	Arquitetura hexagonal	45
3.6.2	Fluxo de dados do <i>pipeline</i>	46
3.6.3	Justificativas técnicas	47
3.7	Ferramentas e Bibliotecas Utilizadas	48
3.7.1	Gerenciamento de Ambiente – <i>Poetry</i>	48
3.7.2	Bibliotecas Utilizadas	48
4	RESULTADOS	50
4.1	Avaliação do MLP Regressor	50
4.1.1	Métricas de Desempenho	50
4.1.2	Análise de operadores	52
4.1.3	Pontuação de eficiência individual	52
4.2	Análise de clusters K-means	53
4.2.1	Definição e Critério de Escolha do Número de Clusters	53
4.2.2	Resultados dos <i>clusters</i>	54
4.2.3	Aplicação dos <i>cluster</i>	55
4.3	Modelos	55
5	CONCLUSÕES E CONSIDERAÇÕES PARA TRABALHOS FUTUROS	57
	Referências	59

1 Introdução

Este capítulo irá introduzir a formulação do problema de pesquisa, os objetivos do trabalho e a estrutura do texto.

1.1 Contextualização do Tema e Problema de Pesquisa

A manutenção de equipamentos industriais, especialmente aqueles utilizados em setores como a construção e a mineração, é um dos principais fatores que impactam diretamente na produtividade e nos custos operacionais. Equipamentos a diesel, como escavadeiras, tratores e caminhões, desempenham um papel crucial nessas indústrias, operando frequentemente em condições extremas. Esses ambientes desafiadores exigem um gerenciamento eficiente de manutenção para garantir o funcionamento contínuo dos equipamentos e minimizar falhas inesperadas.

A implementação de estratégias de manutenção preditiva, que utilizam tecnologias avançadas para monitorar e prever o estado dos equipamentos, tem se mostrado eficaz na redução de custos e aumento da disponibilidade operacional. Segundo [Aqueveque et al. \(2021\)](#), a utilização de sensores e algoritmos de aprendizado de máquina permitiu a detecção precoce de falhas em máquinas móveis de mineração, com precisão superior a 90% .

De modo geral, a mineração é uma atividade de alto risco, que exige um alto investimento financeiro. Segundo [IBRAM – Instituto Brasileiro de Mineração \(2018\)](#), o investimento direcionado para a manutenção dos ativos usados na mineração corresponde de 35% a 50% do orçamento aplicado a operação da mina. Em relação aos custos com transporte de minério, esses podem corresponder até 60% dos custos de operação ([KRISTJANPOLLER et al., 2020](#)).

As abordagens tradicionais de manutenção, como a corretiva e a preventiva, desempenham papéis fundamentais na gestão de equipamentos industriais. A manutenção corretiva é realizada após a ocorrência de falhas, o que pode levar a custos elevados, interrupções não planejadas e riscos à segurança dos trabalhadores . Por outro lado, a manutenção preventiva, embora eficaz na redução de falhas inesperadas, pode resultar em gastos desnecessários com a substituição antecipada de componentes que ainda não atingiram o fim de sua vida útil . Além disso, autores como [Slack, Chambers e Johnston \(2002\)](#) destacam que essas abordagens, apesar de amplamente utilizadas, apresentam limitações quando comparadas a estratégias mais tecnológicas baseadas em análise preditiva e dados em tempo real.

A Indústria 4.0, caracterizada pela integração de tecnologias avançadas como Internet das Coisas (IoT), Big Data e Inteligência Artificial (IA), tem transformado o monitoramento e a manutenção de equipamentos industriais. Nesse contexto, sensores e dispositivos IoT passaram a captar dados operacionais em tempo real a partir de diversas redes de comunicação já consolidadas, como a *Controller Area Network* (CAN). Essas redes são amplamente utilizadas em veículos e maquinários pesados, possibilitando a obtenção de informações críticas, como temperatura, pressão, RPM e consumo de combustível, dados essenciais para a análise do desempenho do equipamento e para a implementação de estratégias de manutenção preditiva. De acordo com [Aqueveque et al. \(2021\)](#), a utilização de redes como a CAN permite a integração de dados em tempo real para o monitoramento de parâmetros críticos, o que tem um impacto direto na eficiência da manutenção. Além do mais, [Zheng, Paiva e Gurciullo \(2020\)](#) destacam que a implementação de IoT e *Big Data* possibilita a análise preditiva do desempenho dos equipamentos, resultando na redução de custos com falhas inesperadas.

Neste contexto, a manutenção preditiva emerge como uma solução promissora, utilizando dados coletados por sensores e sistemas de monitoramento, aliados a algoritmos de aprendizado de máquina, para antecipar falhas, monitorar a saúde do equipamento em tempo real e otimizar as estratégias de manutenção. A utilização da rede CAN, com sua capacidade de fornecer dados em tempo real, é particularmente vantajosa para o desenvolvimento de modelos preditivos de falhas. O trabalho de [Ledmaoui et al. \(2025\)](#) relata que a análise de dados em tempo real, por meio de sensores e algoritmos de aprendizado de máquina, tem se mostrado eficaz na redução de custos e no aumento da disponibilidade operacional.

Dado o exposto, torna-se interessante desenvolver um modelo capaz de monitorar continuamente o estresse do motor e padrões de operação dos condutores, permitindo identificar condições de uso mais severas, evitar sobrecargas desnecessárias e otimizar o desempenho global do equipamento.

1.2 Objetivo Geral

Propõe-se neste trabalho o desenvolvimento de um modelo preditivo do *Engine Stress Index* (ESI) baseado em aprendizado de máquina, utilizando dados extraídos da rede CAN de um modelo de caminhão de uma empresa de mineração, com o objetivo de avaliar o estresse do motor e padrões de operação dos condutores, para fornecer indicadores contínuos de desempenho e contribuir para a otimização das operações em termos de segurança, produtividade e manutenção preditiva.

1.3 Objetivos Específicos

Desenvolver um *pipeline* modular e escalável de predição do índice de estresse do motor (ESI), capaz de:

- Processar dados de equipamentos a diesel em lotes, simulando uma aplicação em tempo real, capturados via rede CAN;
- Prever o ESI a partir de variáveis operacionais críticas (RPM, velocidade, consumo instantâneo do combustível, temperatura do motor);
- Classificar padrões operacionais de cada operador do equipamento por meio de um algoritmo de agrupamento não supervisionado;
- Gerar métricas derivadas, como o *Efficiency Score*, que quantifica a eficiência relativa do operador com base no estresse imposto ao motor;
- Fornecer informações acionáveis para análise de desempenho, que poderão ser usados para o planejamento de manutenção e avaliação de eficiência operacional.

1.4 Justificativa

A importância deste trabalho reside no impacto significativo que a manutenção preditiva pode ter sobre os custos operacionais e a eficiência dos processos industriais. Em indústrias que dependem de equipamentos pesados, como construção civil, mineração e transporte, a falha inesperada de um equipamento pode resultar em prejuízos financeiros elevados, perda de produtividade e até mesmo riscos à segurança. Tradicionalmente, a manutenção tem sido realizada de forma corretiva ou preventiva, ambas com limitações substanciais. A manutenção corretiva gera custos com reparos inesperados, enquanto a preventiva pode resultar em trocas prematuras de componentes, gerando custos desnecessários.

A proposta de utilizar dados extraídos da rede CAN para prever falhas em equipamentos a diesel oferece uma alternativa mais eficiente e econômica. A rede CAN, sendo um protocolo amplamente utilizado para comunicação entre sistemas de controle de veículos e maquinários, permite uma coleta rica de dados operacionais, como temperatura, rotação do motor (RPM), pressão, e consumo de combustível. Esses dados podem ser usados para a detecção precoce de falhas, permitindo que a manutenção seja realizada apenas quando necessário, evitando paradas inesperadas e o desperdício de recursos.

A aplicação de algoritmos de Inteligência Artificial (IA) para análise desses dados representa um avanço importante, uma vez que essas técnicas podem identificar padrões complexos e prever falhas com maior precisão do que os métodos tradicionais. Com a

implementação de um modelo de manutenção preditiva, as empresas podem não só reduzir seus custos de manutenção, mas também melhorar a confiabilidade e a disponibilidade dos seus equipamentos, otimizando a produtividade das operações. Portanto, este trabalho pode ser usado como uma ferramenta que irá fornecer dados para ajudar no desenvolvimento de um modelo, para a modernização e otimização da gestão de manutenção na indústria.

1.5 Estrutura do Trabalho

O presente trabalho está estruturado da seguinte forma:

- **Capítulo 2 – Fundamentação Teórica:** Apresenta os conceitos fundamentais sobre manutenção preditiva, redes CAN e inteligência artificial aplicados em sistemas industriais e as técnicas de aprendizado de máquina utilizado no trabalho.
- **Capítulo 3 – Metodologia:** Descreve a abordagem adotada para a coleta e pré-processamento dos dados, além da construção e avaliação dos modelos de inteligência artificial.
- **Capítulo 4 – Resultados e Discussões:** Apresenta os resultados obtidos com o modelo desenvolvido, discutindo sua eficácia e possíveis melhorias.
- **Capítulo 5 – Considerações Finais:** Conclui o trabalho com uma análise dos resultados, as contribuições do estudo e sugestões para futuras pesquisas e aplicações.

2 Fundamentação Teórica

Este capítulo apresenta a base científica usada para o desenvolvimento desse trabalho.

2.1 Manutenção

A gestão de manutenção industrial é composta por um conjunto de técnicas fundamentais para garantir a continuidade operacional, a segurança e a eficiência dos processos produtivos. Entre as abordagens mais relevantes estão a manutenção corretiva, aplicada após a falha; a manutenção preventiva, realizada em intervalos programados para evitar falhas e a manutenção preditiva, baseada no monitoramento contínuo da condição dos ativos. Cada uma dessas técnicas possui características, vantagens e limitações que as tornam adequadas a diferentes contextos operacionais. No setor de mineração, onde os equipamentos são submetidos a condições severas de operação e longos ciclos de trabalho, a escolha e combinação estratégica dessas técnicas tornam-se decisivas para otimizar a disponibilidade, reduzir custos e evitar perdas produtivas

2.1.1 Manutenção corretiva

A manutenção corretiva é definida como a ação tomada para restaurar um equipamento ou sistema ao seu estado funcional após a ocorrência de uma falha ou avaria. Em muitos casos, a manutenção corretiva é realizada quando o equipamento já apresenta defeitos, o que leva a paradas imprevistas, perda de produção e aumento dos custos operacionais. Embora em algumas situações esse tipo de manutenção seja inevitável, ele pode se tornar problemático se for a única forma de manutenção utilizada em uma mina, segundo a norma NBR 5462, essa prática é considerada uma das mais custosas, devido a paradas prolongadas e prejuízos na produção ([FARIA; RAPOSO, 2023](#)).

Esta estratégia de manutenção possui Natureza reativa. Ela ocorre após a falha ou defeito, ou seja, não há planejamento prévio para impedir o problema. Isso pode resultar em falhas graves que interrompem a produção de maneira não planejada, com impactos financeiros substanciais ([PENHA; BATISTA; VIEGAS, 2021](#)).

Como a manutenção corretiva é realizada de maneira emergencial, ela frequentemente envolve custos mais elevados. Além dos custos com reparos, podem ser necessários serviços adicionais, como o transporte de peças sobressalentes e mão de obra especializada. Além disso, as falhas inesperadas causam paralisações imprevistas da produção, afetando

a continuidade dos processos e o cumprimento dos cronogramas de operação (MAGALHÃES, 2021).

O uso contínuo sem intervenções preventivas ou preditivas pode diminuir a vida útil dos equipamentos, uma vez que falhas repetitivas, mesmo que pequenas, podem causar danos cumulativos. Isso acelera o desgaste dos componentes e requer manutenções mais dispendiosas a longo prazo (FARIA; RAPOSO, 2023).

Embora a manutenção corretiva tenha o custo inicial mais baixo em comparação com a manutenção preventiva ou preditiva, devido à ausência de necessidade de inspeções ou monitoramento constante dos equipamentos, ela possui algumas vantagens e desvantagens. Entre as vantagens, destaca-se o custo inicial baixo, já que não são necessárias inspeções ou monitoramento contínuo. Além disso, a simplicidade operacional é um ponto positivo, pois a manutenção corretiva é mais simples em termos de execução, já que os técnicos ou operadores realizam o reparo necessário após a falha, sem a necessidade de planejamentos complexos.

Por outro lado, a manutenção corretiva também apresenta desvantagens consideráveis. A principal delas é o alto custo com paradas, pois, embora o custo inicial seja baixo, o impacto das paradas imprevistas pode gerar custos elevados com a interrupção das operações. Além disso, falhas frequentes e não planejadas podem danificar irreparavelmente componentes, levando a reparos mais caros ou até mesmo à necessidade de substituição de peças. Por fim, a manutenção corretiva contribui para uma baixa eficiência operacional, já que equipamentos quebrados afetam diretamente a produção e a eficiência, resultando em menor produtividade.

Na mineração, a manutenção corretiva é frequentemente aplicada a equipamentos móveis como escavadeiras, pás-carregadeiras e caminhões de transporte. Estes equipamentos operam em condições extremas, como alta vibração, poeira e umidade, o que os torna suscetíveis a falhas inesperadas. Segundo Faria e Raposo (2023) em casos de falha de motor, quebra de sistema hidráulico ou desgaste excessivo de pneus, a manutenção corretiva se torna essencial.

Um exemplo de aplicação da manutenção corretiva pode ser encontrado na manutenção de pás-carregadeiras utilizadas em minas de ferro. Quando uma dessas máquinas sofre uma falha em seu sistema hidráulico, ela precisa ser reparada imediatamente para evitar que a produção seja interrompida. Nesse caso, a manutenção corretiva é acionada para consertar a falha, mas não impede que uma nova falha aconteça, o que pode levar a mais custos operacionais no futuro (PENHA, 2021).

A manutenção corretiva é muitas vezes vista como uma abordagem reativa em sistemas de gerenciamento de manutenção. Quando não há planejamento, os gestores de manutenção se veem forçados a operar em um ciclo constante de emergências. Portanto,

muitas empresas buscam alternativas mais eficientes, como a manutenção preventiva e a preditiva, para reduzir a necessidade de manutenção corretiva e aumentar a disponibilidade dos equipamentos (MAGALHÃES, 2021).

O estudo de Faria e Raposo (2023) destaca que a manutenção corretiva não pode ser completamente eliminada, mas deve ser minimizada para evitar os impactos negativos que ela acarreta. A gestão de confiabilidade é uma abordagem que integra diferentes tipos de manutenção, com foco na maximização da disponibilidade dos equipamentos enquanto minimiza os custos totais de manutenção.

2.1.2 Manutenção preventiva

A manutenção preventiva consiste em um conjunto de ações planejadas com o objetivo de evitar falhas e manter os equipamentos operando dentro dos padrões esperados de desempenho. Ela é baseada em intervalos regulares de tempo ou uso, estabelecidos com base em recomendações do fabricante, histórico de falhas ou critérios estatísticos. No setor de mineração, onde a continuidade da operação é fundamental para a produtividade, a manutenção preventiva assume papel estratégico.

Segundo Penha, Batista e Viegas (2021), a manutenção preventiva é essencial para garantir a confiabilidade operacional dos equipamentos em plantas de mineração. Ela permite identificar possíveis problemas antes que resultem em falhas, reduzindo o número de paradas não planejadas e aumentando a vida útil dos ativos. Isso é alcançado por meio de inspeções regulares, ajustes, limpezas, lubrificações e substituições programadas de componentes.

Além disso, a autora destaca que o planejamento eficiente da manutenção preventiva, quando alinhado ao gerenciamento de projetos e às práticas do PMBOK (*Project Management Body of Knowledge*), contribui diretamente para o controle de recursos, o cumprimento de prazos e a segurança dos processos industriais.

As principais vantagens da utilização da manutenção preventiva incluem a redução de paradas inesperadas, uma vez que ao intervir nos equipamentos antes que falhas ocorram, a manutenção preventiva diminui a frequência de interrupções inesperadas na produção (PENHA; BATISTA; VIEGAS, 2021). Além disso, equipamentos submetidos a manutenções periódicas apresentam maiores taxas de disponibilidade, o que é fundamental no contexto de alta demanda de produção das operações de mina. A manutenção preventiva também contribui para o controle de custos, pois, embora envolva custos regulares com inspeções e trocas de componentes, ela evita os altos custos associados às manutenções corretivas emergenciais. Outra vantagem é a segurança, já que equipamentos bem mantidos tendem a operar com menor risco de acidentes ou falhas graves que possam comprometer a integridade dos operadores ou da planta.

No entanto, de acordo com [Marins \(2024\)](#), as principais desvantagens da abordagem preventiva incluem a substituição prematura de peças. Em muitos casos, componentes podem ser trocados antes de atingirem o fim de sua vida útil, baseando-se apenas em prazos fixos. Isso pode resultar em desperdício de materiais e aumento de custos operacionais desnecessários. Outra desvantagem é o custo operacional contínuo, já que a manutenção preventiva exige recursos financeiros recorrentes, incluindo peças de reposição, mão de obra especializada e tempo de parada programada, sendo que esses custos são mantidos mesmo quando os equipamentos estão funcionando perfeitamente. Além disso, a necessidade de paradas programadas, mesmo quando os equipamentos estão operacionais, pode interromper o fluxo produtivo, exigindo reprogramações logísticas para não impactar o volume de produção.

Em ambientes industriais de mineração, como na moagem de bauxita no processo Bayer descrito por [Penha, Batista e Viegas \(2021\)](#), a manutenção preventiva é aplicada em sistemas críticos como moinhos, transportadores de correia, silos e bombas. As inspeções periódicas e a programação de revisões evitam que os equipamentos atinjam estados de falha irreversível. Durante o *Major Overhaul* de um moinho de bauxita, por exemplo, a manutenção preventiva envolve desmontagens programadas para substituição de peças desgastadas e revisão geral do sistema. O objetivo é garantir que o equipamento retorne à operação com confiabilidade e desempenho esperado, minimizando riscos operacionais e otimizando a performance do processo produtivo.

A etapa de planejamento e execução da manutenção preventiva requer um processo rigoroso e sistematizado, voltado à maximização da disponibilidade dos equipamentos e à redução de paradas não programadas. Esse planejamento envolve a definição da periodicidade de cada atividade de manutenção, de acordo com as recomendações do fabricante e o histórico operacional do equipamento. Inclui também a estimativa do tempo de máquina parada, permitindo o agendamento das intervenções com o mínimo impacto produtivo, além da alocação adequada de recursos humanos, peças e ferramentas necessárias para a execução das tarefas. Por fim, o controle documental e o registro histórico das intervenções garantem rastreabilidade, conformidade com normas técnicas e fornecem insumos valiosos para análises futuras de confiabilidade e otimização do plano de manutenção ([MARQUES DA COSTA; PAULO SACHETTO, 2023](#)).

O trabalho de [Penha, Batista e Viegas \(2021\)](#) também enfatiza que o uso de ferramentas de gerenciamento de projetos, como cronogramas estruturados e análises de risco, é fundamental para o sucesso de grandes manutenções preventivas em plantas industriais.

Em resumo, apesar de suas vantagens, a manutenção preventiva pode levar à substituição prematura de peças ainda em bom estado, resultando em custos desnecessários. Além disso, se mal planejada, pode interferir no cronograma de produção. Assim, seu

sucesso depende do equilíbrio entre frequência, custo e impacto operacional.

2.1.3 Manutenção por condição

A manutenção por condição é uma abordagem estratégica que visa realizar intervenções em equipamentos somente quando suas condições operacionais indicam que uma falha iminente ou anomalia pode ocorrer. Diferente das abordagens de manutenção preditiva ou preventiva, a manutenção por condição baseia-se em monitoramento contínuo dos componentes para ajustar as atividades de manutenção de forma dinâmica, otimizando os custos e evitando tanto falhas inesperadas quanto manutenções desnecessárias.

Segundo [Teixeira, Lopes e Braga \(2020\)](#), a implementação da manutenção por condição tem ganhado destaque devido ao seu potencial para aumentar a eficiência operacional e reduzir custos com manutenções desnecessárias. A utilização de tecnologias como sensores, sistemas de monitoramento em tempo real e análise de dados permite que as intervenções sejam realizadas de forma mais assertiva, baseando-se no estado real dos ativos.

A principal vantagem dessa abordagem está em sua capacidade de realizar manutenções apenas quando necessário, evitando o desgaste excessivo de componentes ou substituições prematuras. Estudos mostram que a aplicação da manutenção por condição pode aumentar a vida útil dos equipamentos, reduzir o tempo de inatividade não programado e melhorar a confiabilidade dos processos industriais ([TEIXEIRA; LOPES; BRAGA, 2020](#)).

A manutenção por condição é especialmente eficaz em ambientes industriais complexos, onde os custos de falhas são elevados, como no setor de petróleo e gás, mineração e energia elétrica. A implementação de técnicas de monitoramento remoto e análise preditiva proporciona uma melhor alocação dos recursos de manutenção, além de contribuir para a segurança operacional e a redução de custos operacionais. De acordo com [Teixeira, Lopes e Braga \(2020\)](#), a literatura tem demonstrado resultados positivos na implementação dessa estratégia, principalmente com o auxílio de ferramentas tecnológicas avançadas, como a Internet das Coisas (IoT) e inteligência artificial.

2.1.4 Manutenção preditiva

A manutenção preditiva é uma abordagem estratégica voltada para o acompanhamento contínuo das condições reais de funcionamento dos equipamentos, com o objetivo de prever falhas antes que elas ocorram. Diferentemente das manutenções corretiva e preventiva, essa modalidade utiliza tecnologias de monitoramento e análise de dados para estimar com precisão o momento ideal para a intervenção, otimizando a vida útil dos componentes e reduzindo os custos operacionais.

Segundo [Marins \(2024\)](#), a manutenção preditiva representa um avanço significativo na gestão de ativos industriais, especialmente no setor de mineração, onde a alta disponibilidade dos equipamentos é crítica para o sucesso operacional. Ela permite intervenções baseadas em condições reais e não em intervalos fixos de tempo, o que melhora a eficiência e confiabilidade da planta ou da central de manutenção.

Essa abordagem é baseada em condições, fundamentada na coleta e análise de dados sobre variáveis operacionais (vibração, temperatura, ruído, pressão, entre outros), que indicam o estado real dos componentes monitorados. De acordo com [Xenos \(1998\)](#) ao detectar desgastes e anomalias em estágio inicial, a manutenção preditiva evita falhas inesperadas, aumentando a disponibilidade dos ativos.

O uso de ferramentas analíticas, sensores e tecnologias como Internet das Coisas (IoT) e aprendizado de máquina (*Machine Learning*) permite decisões mais assertivas sobre o momento de realizar manutenção.

A aplicação da manutenção preditiva em equipamentos de mineração, como escavadeiras hidráulicas, carregadeiras, correias transportadoras e moinhos — tem gerado resultados expressivos, tais como:

- Aumento da disponibilidade física dos equipamentos;
- Prolongamento dos intervalos entre paradas programadas, com redução de downtime;
- Aproveitamento total do ciclo de vida dos componentes, evitando trocas prematuras;
- Melhora na confiabilidade e segurança operacional, por evitar falhas catastróficas;

Estudos de caso analisados por [Marins \(2024\)](#) demonstram que o monitoramento preditivo de componentes como mancais, motores a diesel e tubulações permitiu detectar microfalhas em estágio inicial, permitindo ações corretivas planejadas, com mínima interferência na produção.

2.2 Rede CAN

A rede CAN é um sistema de comunicação serial padronizado internacionalmente pela ISO 11898-1:2015, desenvolvido originalmente pela empresa Robert Bosch GmbH em 1983, com o objetivo de otimizar a comunicação entre unidades de controle eletrônico (ECUs) em veículos automotivos. Antes do desenvolvimento da rede CAN, cada módulo de controle de um veículo, como o motor, o sistema de freios e os sensores necessitavam de uma conexão elétrica direta e dedicadas entre si, o que resultava em chicotes complexos, pesados e caros ([LAWRENZ, 2013](#)). A introdução do barramento CAN reduziu significativamente essa complexidade, permitindo que múltiplos dispositivos compartilhassem um

mesmo meio físico de comunicação, trocando mensagens curtas e altamente confiáveis sem a necessidade de um controlador central.

A rede CAN foi projetado com foco em alta confiabilidade, tolerância a falhas e robustez eletromagnética, características essenciais para ambientes automotivos e industriais. De acordo com [Corrigan \(2016\)](#), o CAN é um sistema multi-master e orientado a mensagens, em que qualquer nó pode transmitir dados sempre que o barramento estiver livre, e todas as mensagens são recebidas simultaneamente por todos os nós conectados. Essa arquitetura baseada em broadcast garante a consistência das informações em toda a rede e permite uma coordenação eficiente entre múltiplos módulos.

O protocolo CAN rapidamente ultrapassou os limites da indústria automotiva e passou a ser adotado em diversos setores industriais, como automação de processos, sistemas médicos, máquinas agrícolas, equipamentos de mineração e embarcações navais, devido à sua robustez, baixo custo e simplicidade de implementação ([ETSCHBERGER, 2001](#)). A padronização internacional pela ISO 11898-1:2015 consolidou sua aplicação em sistemas industriais que exigem comunicação determinística e confiável, o que tornou o CAN uma das tecnologias de rede mais utilizadas na Indústria 4.0 para telemetria, controle distribuído e manutenção preditiva ([BOLAND et al., 2021](#)).

2.2.1 Arquitetura e Camadas do Padrão ISO 11898

O protocolo CAN foi normalizado pela Organização Internacional de Normalização (ISO) sob o padrão ISO 11898-1:2015, como visto na Figura 1, o qual define as especificações elétricas e de comunicação de um barramento serial voltado para aplicações em tempo real e ambientes industriais. Segundo [Corrigan \(2016\)](#), o protocolo CAN segue a estrutura conceitual do modelo OSI (*Open Systems Interconnection*), utilizando principalmente as duas camadas inferiores: a camada física (Physical Layer) e a camada de enlace de dados (*Data Link Layer*). Essa arquitetura modular é o que permite a interoperabilidade entre diferentes dispositivos e fabricantes, garantindo confiabilidade e consistência na comunicação entre nós da rede.

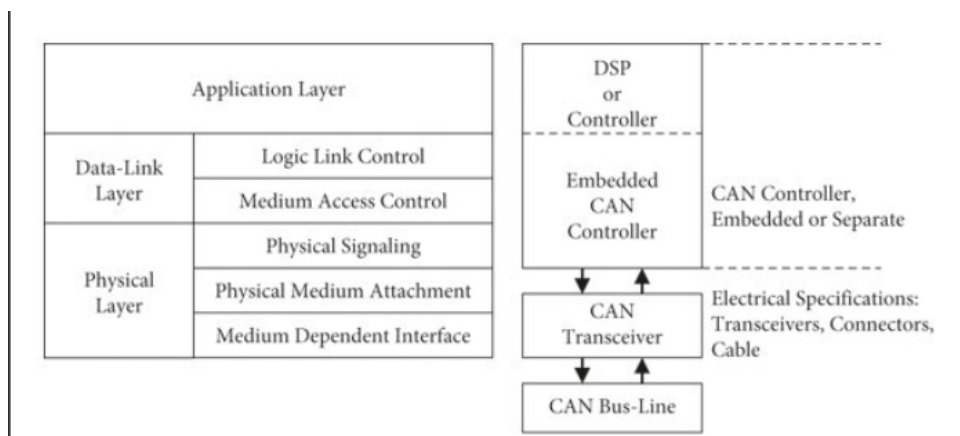


Figura 1 – Arquitetura da norma ISO 11898 em camadas

Fonte: [Corrigan \(2016\)](#)

A camada física é responsável por especificar os aspectos elétricos e de transmissão do sinal, incluindo o tipo de cabos, conectores, níveis de tensão, topologia de rede e taxas de transmissão. O CAN utiliza uma topologia de barramento linear composta por um par trançado de fios, identificado como CAN High (CANH) e CAN Low (CANL). Essa configuração adota um sinal diferencial balanceado, que reduz a interferência eletromagnética e assegura maior imunidade a ruídos externos, característica essencial em ambientes industriais e automotivos. O padrão ISO 11898 especifica que o sistema opere com uma impedância característica de $120\ \Omega$ (ohms) e que as extremidades do barramento sejam terminadas com resistores do mesmo valor para evitar reflexões de sinal ([CORRIGAN, 2016](#)).

A variação de tensão entre as linhas define dois estados lógicos: o estado dominante, quando há diferença de aproximadamente 2 V entre CANH e CANL, e o estado recessivo, quando ambas permanecem próximas de 2,5 V. Essa forma de codificação binária permite a detecção de erros e o controle de acesso ao meio sem destruição de mensagens concorrentes.

Já a camada de enlace de dados é responsável pelo gerenciamento do tráfego de mensagens na rede, pelo controle de acesso ao meio e pela detecção e correção de erros. Essa camada é subdividida em duas subcamadas: *Medium Access Control* (MAC) e *Logical Link Control* (LLC). A subcamada MAC coordena o acesso ao barramento por meio do método *Carrier Sense Multiple Access with Collision Detection and Arbitration on Message Priority* (CSMA/CD+AMP). Nessa abordagem, cada nó monitora o barramento antes de transmitir, e, caso duas transmissões ocorram simultaneamente, o sistema realiza uma arbitragem bit a bit com base na prioridade das mensagens, garantindo que apenas a de maior prioridade continue sendo transmitida. A subcamada LLC, por sua vez, é responsável pela formatação dos quadros de mensagem, controle de fluxo e reconhecimento de dados recebidos, garantindo que todas as unidades da rede recebam informações consistentes e sincronizadas ([CORRIGAN, 2016](#)).

2.2.2 Estrutura de Mensagens e Identificadores

O protocolo do CAN organiza a comunicação entre dispositivos por meio de quadros de mensagem (*frames*), que contêm todos os elementos necessários para garantir a integridade, a priorização e a confiabilidade da transmissão. Diferente de protocolos tradicionais de rede que utilizam comunicação ponto a ponto com endereços específicos, é um barramento orientado a mensagens. Isso significa que os dispositivos não são identificados por endereços únicos, mas sim por identificadores de mensagem, que indicam o tipo e a prioridade da informação transmitida. Existem dois tipos de mensagens, o primeiro padrão com identificador de 11 bits, observada na Figura 2 e a mensagem estendida, geralmente usada em equipamentos pesados com identificador de 29 bits, de acordo com a Figura 3 (CORRIGAN, 2016).

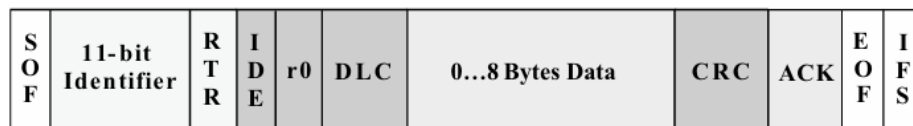


Figura 2 – CAN padrão: identificador de 11 bits

Fonte: Corrigan (2016)

Segundo o padrão ISO 11898-1:2015, o quadro de dados (data frame) é o tipo mais comum de mensagem CAN e é composto por diversos campos que desempenham funções específicas:

- **SOF** (*Start of Frame*) - bit dominante que indica o início da transmissão e sincroniza os nós do barramento.
- **Identificador** - O identificador CAN padrão de 11 bits estabelece a prioridade da mensagem. Quanto menor o valor binário, maior a sua prioridade.
- **RTR** (*Remote Transmission Request*) – O bit de solicitação de transmissão remota é dominante quando informações são necessárias de outro nó. Todos os nós recebem a solicitação, mas o identificador determina o nó específico. Os dados de resposta também são recebidos por todos os nós e utilizados por qualquer nó interessado. Dessa forma, todos os dados utilizados em um sistema são uniformes.
- **IDE** (*Identifier Extension*) – Um bit de identificador único dominante significa que um identificador CAN padrão, sem extensão, está sendo transmitido.
- **r0** - Bit reservado (para possível utilização em futuras alterações da norma).
- **DLC** (*Data Length Code*) – O código de comprimento de dados de 4 bits contém o número de bytes de dados que estão sendo transmitidos.

- *Data Field* - contém os dados da aplicação (até 64 bits).
- *CRC (Cyclic Redundancy Check)* – A verificação de redundância cíclica de 16 bits (15 bits mais o delimitador) contém a soma de verificação (número de bits transmitidos) dos dados da aplicação anteriores para detecção de erros.
- *ACK (Acknowledge)* – Cada nó que recebe uma mensagem correta sobrescreve este bit recessivo na mensagem original com um bit dominante, indicando que uma mensagem sem erros foi enviada. Caso um nó receptor detecte um erro e deixe este bit recessivo, ele descarta a mensagem e o nó transmissor repete a mensagem após a retransmissão. Dessa forma, cada nó confirma (ACK) a integridade de seus dados. O ACK tem 2 bits, um é o bit de confirmação e o segundo é um delimitador.
- *EOF (End of Frame)* – Este campo de fim de quadro de 7 bits marca o final de um quadro (mensagem) CAN e desabilita o preenchimento de bits, indicando um erro de preenchimento quando dominante.
- *IFS (Interframe Space)* – espaço de 7 bits contém o tempo necessário para que o controlador mova um quadro recebido corretamente para sua posição na área do *buffer* de mensagens.

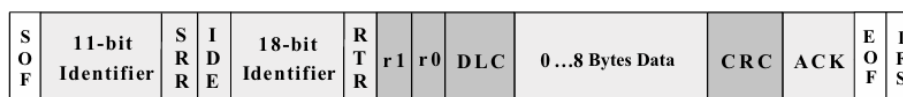


Figura 3 – CAN estendido: identificador de 29 bits

Fonte: [Corrigan \(2016\)](#)

Conforme mostrado na Figura 3 a mensagem CAN estendida é igual à mensagem padrão, com a adição de:

- *SRR (Substitute Remote Request)* – O bit de solicitação remota substituta substitui o bit RTR na posição padrão da mensagem, funcionando como um espaço reservado no formato estendido.
- *IDE (Identifier Extension)* – Um bit recessivo na extensão do identificador indica que mais bits de identificador se seguem. A extensão de 18 bits segue o IDE.
- *r1* – Após os bits RTR e r0, um bit de reserva adicional foi incluído antes do bit DLC.

A priorização de mensagens é uma das características que tornam o protocolo CAN adequado para sistemas críticos em tempo real. O processo de arbitragem garante que,

quando dois ou mais nós tentam transmitir simultaneamente, apenas o quadro com o identificador de maior prioridade continue no barramento, enquanto os demais aguardam o próximo ciclo de transmissão. Essa arbitragem é não destrutiva, ou seja, nenhuma mensagem é perdida durante o processo, preservando a integridade dos dados transmitidos.

2.2.3 Mecanismos de Comunicação e Controle de Acesso

A comunicação baseia-se em um modelo multimestre, em que qualquer nó pode iniciar uma transmissão sempre que o barramento estiver livre. Para coordenar o acesso de múltiplos dispositivos sem comprometer a integridade dos dados, este controle é atingido por meio do CSMA/CD +AMP.

De acordo com [Corrigan \(2016\)](#), durante a arbitragem, cada transmissor envia bits de seu identificador e simultaneamente lê o valor real presente no barramento. O bit dominante (nível lógico 0) sempre sobrepõe o bit recessivo (nível lógico 1). Assim, se um nó transmite um bit recessivo e detecta um bit dominante, ele reconhece que existe outra mensagem com prioridade superior e automaticamente cessa sua transmissão. O nó vencedor, aquele cuja sequência de bits se mantém dominante por toda a fase de identificação, continua enviando seu quadro até o final. Esse mecanismo assegura que mensagens críticas, como as relacionadas à segurança, sejam sempre processadas primeiro.

O sistema de comunicação do CAN, portanto, é determinístico e tempo-real, já que a prioridade das mensagens é conhecida e constante. Além disso, o barramento permite que todos os nós recebam todas as mensagens transmitidas, reforçando o caráter orientado a broadcast do protocolo. Essa característica simplifica a arquitetura de sistemas distribuídos e aumenta a confiabilidade, pois qualquer unidade pode atuar como transmissor ou receptor conforme a necessidade da aplicação ([LAWRENZ, 2013](#)).

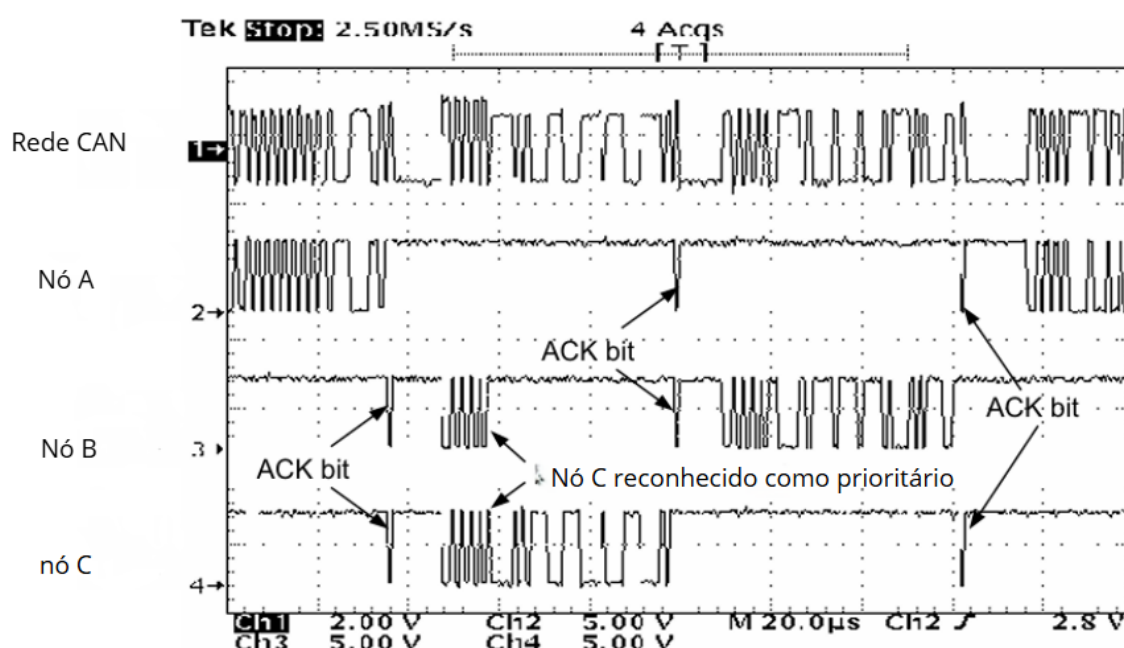


Figura 4 – Tráfego de informações na CAN

Fonte: Corrigan (2016)

Em uma aplicação prática, por exemplo, um sensor de temperatura pode enviar uma atualização usada para ajustar a velocidade de um ventilador. Se um sensor de pressão tentar transmitir ao mesmo tempo, o processo de arbitragem garante que apenas uma mensagem seja enviada. No caso da Figura 4, o nó A conclui o envio de sua mensagem enquanto os nós B e C confirmam o recebimento correto; em seguida, B e C iniciam a arbitragem, C vence por prioridade e transmite sua mensagem, que é reconhecida por A e B, permitindo que B continue com a sua. Também é importante observar a polaridade oposta entre a entrada e a saída no barramento.

2.2.4 Mecanismos de Detecção e Correção de Erros

São mecanismos que garantem, ainda mais confiabilidade e robustez na comunicação mesmo em ambientes com ruído elétrico, vibração ou interferência eletromagnética, condições comuns em veículos e máquinas industriais. Essa robustez é obtida por meio de um conjunto de mecanismos de detecção e confinamento de erros, definidos pelo padrão ISO 11898-1:2015 e detalhados por Corrigan (2016).

Esses mecanismos incluem verificações contínuas dos bits enviados e recebidos, controle de sequências de bits para manter a sincronização, validação por meio de códigos de redundância cíclica (CRC) e checagem da estrutura e do reconhecimento das mensagens transmitidas. Em conjunto, essas técnicas permitem identificar e interromper imediatamente qualquer transmissão corrompida, assegurando que apenas quadros válidos circulem pelo

barramento [Corrigan \(2016\)](#).

Esses métodos são complementados por um sistema de confinamento de falhas (*fault confinement*), que diferencia nós operacionais de nós com falhas recorrentes por meio de contadores de erro de transmissão e recepção. Quando os limites definidos pela norma ISO 11898-1:2015 são excedidos, o nó é colocado automaticamente em modo de erro passivo ou desconectado da rede (*bus-off*), evitando que comprometa o funcionamento geral. Esse modelo de autodiagnóstico mantém a operação da rede mesmo diante de falhas localizadas, característica essencial em aplicações industriais e automotivas, nas quais a confiabilidade e a segurança da comunicação são fundamentais ([CORRIGAN, 2016](#)).

2.2.5 Aplicações

Desde sua introdução pela Robert Bosch GmbH na década de 1980, a CAN tornou-se um dos sistemas de comunicação serial mais utilizados em aplicações automotivas, industriais e embarcadas. Inicialmente concebido para substituir complexos chicotes elétricos entre as unidades de controle de um veículo, o CAN expandiu-se rapidamente para outros setores devido à sua robustez, confiabilidade e baixo custo de implementação ([CORRIGAN, 2016](#)).

No setor automotivo, a rede CAN é amplamente empregada para interconectar ECUs (*Electronic Control Units*) responsáveis por funções críticas, como o gerenciamento do motor, freios ABS, airbags, controle de estabilidade, sistemas de injeção eletrônica e painéis de instrumentação. A comunicação em tempo real entre essas unidades é essencial para garantir a segurança, o desempenho e a eficiência energética do veículo. Em veículos comerciais e máquinas fora-de-estrada, é utilizado para integrar sistemas de diagnóstico, monitoramento de motores a diesel e controle de transmissão, assegurando que falhas sejam detectadas e registradas de forma imediata ([AVATEFIPOUR; SARGOLZAEI; ABDELGHANI, 2018](#)).

Na automação industrial, o CAN é empregado em sistemas distribuídos que demandam comunicação determinística e tolerante a falhas. Protocolos derivados como CANopen e DeviceNet estendem o padrão ISO 11898, permitindo o controle de motores elétricos, atuadores, sensores de pressão e temperatura, além de controladores lógicos programáveis (CLPs). Essa padronização facilita a comunicação entre equipamentos de diferentes fabricantes e viabiliza arquiteturas de controle descentralizadas, cada vez mais utilizadas em linhas de produção automatizadas ([ETSCHBERGER, 2001](#)).

O protocolo também se destaca em máquinas agrícolas, equipamentos de mineração e veículos de construção, onde as condições ambientais são severas e o ruído eletromagnético elevado. Nesses casos, a sinalização diferencial e os mecanismos de detecção de erro garantem a transmissão confiável de dados de telemetria, fundamentais para o monitoramento de

parâmetros como rotação do motor (RPM), temperatura, pressão hidráulica e consumo de combustível. Esses dados alimentam plataformas de manutenção preditiva, que permitem identificar anomalias operacionais e otimizar os ciclos de manutenção dos equipamentos (BOLAND *et al.*, 2021).

Além disso, o uso também se estende a setores como médico-hospitalar (sistemas de ventilação e instrumentação cirúrgica), ferroviário, marítimo e aeronáutico, em aplicações que exigem alta confiabilidade, latência reduzida e sincronização precisa entre dispositivos embarcados. Nessas áreas, o protocolo se consolidou como uma solução de comunicação robusta para sistemas embarcados críticos (LAWRENZ, 2013).

2.3 Métodos e Técnicas

Considerando a natureza temporal e multivariada das informações obtidas dos equipamentos, é necessário empregar variáveis que representem de forma consistente o comportamento operacional e o estresse mecânico ao longo do tempo.

As variáveis selecionadas foram escolhidas por serem as disponíveis no sistema da empresa e por empiricamente possuírem relevância direta com o desempenho e a integridade do motor, permitindo a construção de indicadores de esforço e eficiência. Entre as principais variáveis monitoradas destacam-se:

- **Hodômetro:** mede a distância total percorrida pelo equipamento, servindo como indicador de desgaste relacionado ao uso contínuo e apoio para o planejamento de manutenção preventiva.
- **RPM (Rotações por Minuto):** parâmetro essencial para a avaliação do regime de operação do motor. Variações anormais de rotação podem indicar sobrecarga, falhas mecânicas ou condução inadequada.
- **Consumo Instantâneo de Combustível:** reflete diretamente a eficiência energética e o desempenho do motor. Aumentos inesperados podem sinalizar anomalias em sistemas de injeção ou combustão, se comparado com as outras grandezas e estiver fora das suas faixas normais de atuação.
- **Volume Total de Combustível Acumulado:** permite observar o comportamento de consumo ao longo do tempo, contribuindo para análises de eficiência e desgaste progressivo do equipamento.
- **Horímetro:** registra o tempo total de operação do motor, sendo utilizado para o controle de ciclos de manutenção e monitoramento do tempo de vida útil dos componentes críticos.

- **Temperatura do Motor:** variável para a avaliação do estado térmico do sistema. Picos de temperatura podem indicar falhas no sistema de arrefecimento ou sobrecarga operacional.

Essas variáveis são coletadas em tempo real a partir do barramento CAN do equipamento e consolidadas em *micro-lotes* (processamento incremental de dados em lotes de 100 amostras) para análise temporal, com um *delay* de meio segundo para cada agrupamento ser inserido no modelo. A integração dessas informações em um modelo de aprendizado de máquina, como o MLP, permite capturar relações não lineares entre as variáveis e estimar o índice de estresse do motor (ESI).

A combinação entre o monitoramento contínuo, o pré-processamento dos sinais e o uso de algoritmos de inteligência artificial possibilita a identificação de padrões complexos de operação e a predição antecipada de falhas, reforçando a eficiência das estratégias de manutenção preditiva.

2.4 Telemetria

A telemetria é uma tecnologia fundamental na Indústria 4.0, usada para monitoramento e controle remoto de sistemas, baseada na medição, coleta, transmissão e análise de dados provenientes de equipamentos e sensores distribuídos. Sua aplicação tornou-se indispensável em contextos industriais, automotivos e de mineração, pois permite o acompanhamento contínuo das condições de operação de máquinas e processos, viabilizando ações proativas e a otimização da manutenção.

Conforme destacado por [Rakholia et al. \(2025\)](#), a telemetria é um dos pilares da Indústria 4.0, uma vez que integra sensores inteligentes, redes de comunicação e algoritmos de inteligência artificial para formar sistemas de manutenção preditiva e tomada de decisão autônoma. Em tais sistemas, os dados são coletados por sensores embarcados e transmitidos para plataformas de análise que executam algoritmos de aprendizado de máquina ou inteligência artificial, capazes de identificar padrões e anomalias operacionais. Essa abordagem permite prever falhas antes que elas ocorram, evitando paradas não planejadas e reduzindo significativamente os custos de manutenção.

Na prática, a telemetria é implementada por meio de uma infraestrutura composta por sensores físicos, unidades de aquisição de dados e protocolos de comunicação que realizam o transporte das informações em tempo real.

Conforme [Gadiraju \(2024\)](#), os sistemas de telemetria modernos utilizam arquiteturas IoT (Internet das Coisas) combinadas a algoritmos de análise de séries temporais, que realizam o processamento de dados em tempo real e a geração de alertas automáticos quando parâmetros operacionais excedem limites predefinidos. Esse tipo de abordagem,

chamada de telemetria inteligente (*AI-powered telemetry*), está se tornando cada vez mais comum em ambientes industriais, pois permite decisões automatizadas baseadas em dados e integração direta com plataformas de algoritmos de aprendizado de máquina. Além disso, o uso de computação em nuvem tem ampliado as capacidades de processamento desses sistemas, reduzindo a latência na comunicação e aumentando a escalabilidade das soluções de monitoramento remoto.

Segundo [Avatefipour, Sargolzaei e Abdelghani \(2018\)](#), a rede CAN é amplamente utilizada em sistemas de telemetria, especialmente em veículos e maquinários pesados, devido à sua alta capacidade de comunicação em tempo real e robustez em ambientes industriais adversos. Essa rede é ideal para a transmissão de dados de sensores, pois oferece baixa latência e alta confiabilidade, características essenciais para a manutenção.

A análise preditiva baseada em telemetria segue um fluxo de dados bem definido. O processo começa com a coleta de dados por meio de sensores embarcados, os quais transmitem essas informações em tempo real para servidores locais ou na nuvem. Em seguida, os dados são processados por algoritmos de aprendizado de máquina, que identificam padrões de degradação. Como resultado desse processamento, ações automatizadas são geradas, como a emissão de alertas ou recomendações de manutenção.

Além do uso em veículos e maquinários industriais, a telemetria também é amplamente aplicada em setores como energia, agricultura de precisão, saneamento e monitoramento ambiental, onde possibilita o controle remoto de equipamentos distribuídos geograficamente. No contexto deste trabalho, a telemetria é empregada como ferramenta de aquisição e análise de dados operacionais de equipamentos a diesel, com o propósito de antecipar falhas, otimizar a disponibilidade dos ativos e apoiar decisões estratégicas de manutenção.

2.5 Modelo *Multilayer Perceptron* Regressor

O *Multilayer Perceptron Regressor* (MLP) é um modelo de aprendizado supervisionado baseado em redes neurais artificiais do tipo *feedforward*, amplamente reconhecido por sua capacidade de capturar padrões complexos e não lineares em conjuntos de dados. Diferentemente de métodos estatísticos tradicionais, que geralmente assumem relações lineares entre variáveis, o MLP possibilita a modelagem de fenômenos mais sofisticados ao combinar múltiplas transformações não lineares. Essa característica o torna particularmente adequado para aplicações envolvendo telemetria, como no caso deste trabalho, no qual as variáveis extraídas da rede CAN apresentam interações dinâmicas e não triviais decorrentes do funcionamento real de equipamentos a diesel.

A arquitetura de um MLP é composta por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Cada neurônio recebe valores da camada anterior,

realiza uma combinação linear desses valores e aplica uma função de ativação responsável por introduzir a não linearidade no modelo. Formalmente, dada a entrada

$$\mathbf{x} = [x_1, x_2, \dots, x_n], \quad (2.1)$$

o valor intermediário calculado por um neurônio j é expresso por

$$z_j = \sum_i w_{ij} x_i + b_j, \quad (2.2)$$

em que w_{ij} representa os pesos aprendidos durante o treinamento e b_j corresponde ao termo de viés. Em seguida, aplica-se uma função de ativação $\phi(z_j)$, produzindo a saída do neurônio. Neste estudo, utilizou-se a função ReLU (*Rectified Linear Unit*), definida como

$$\phi(z) = \max(0, z), \quad (2.3)$$

a qual foi escolhida devido à sua eficiência computacional, estabilidade durante o processo de otimização e capacidade de mitigar o problema do gradiente dissipativo, frequentemente observado em funções tradicionais como sigmoide e tangente hiperbólica (GLOROT; BORDES; BENGIO, 2011).

Na camada de saída, utilizada para resolver o problema de regressão, aplica-se uma transformação linear dos valores provenientes da última camada oculta, gerando uma estimativa contínua para o alvo. Dessa forma, o MLP implementa uma composição sucessiva de operações lineares e não lineares que permite aproximar funções altamente complexas. Esse comportamento está associado ao Teorema da Aproximação Universal, segundo o qual redes neurais com ao menos uma camada oculta são capazes de aproximar qualquer função contínua sob determinadas condições. Assim, a escolha do MLP fundamenta-se não apenas em sua popularidade, mas sobretudo em sua capacidade comprovada de representar relações não lineares presentes nas variáveis operacionais analisadas.

O treinamento do modelo é realizado por meio do algoritmo de retropropagação de erro (*backpropagation*), combinado a um otimizador baseado em gradiente descendente. O erro entre as previsões do modelo e os valores reais é quantificado pela função de perda *Mean Squared Error* (MSE), definida por

$$L = \frac{1}{N} \sum_{k=1}^N (y_k - \hat{y}_k)^2. \quad (2.4)$$

O objetivo é ajustar pesos e vieses de forma a minimizar essa função. Para garantir estabilidade e reduzir o risco de sobreajuste, foram adotados mecanismos como *early stopping* e validação cruzada, que interrompem o treinamento quando o erro de validação deixa de apresentar redução significativa.

Com base nos testes exploratórios realizados (descritos posteriormente), definiu-se como arquitetura ideal uma rede composta por duas camadas ocultas com 64 e 32

neurônios, respectivamente. Essa configuração apresentou bom equilíbrio entre capacidade de aprendizado e custo computacional, além de desempenho satisfatório em termos de erro médio quadrático e capacidade de generalização. A adoção da função ReLU contribuiu para uma convergência mais rápida e consistente, reforçando a eficiência do modelo no contexto da regressão aplicada aos sinais provenientes da rede CAN. A seleção dessa arquitetura também encontra respaldo na literatura, que recomenda o uso de camadas ocultas contendo dezenas de neurônios para problemas de regressão não linear de complexidade moderada.

Dessa forma, o MLP Regressor utilizado neste trabalho constitui um componente central da metodologia proposta, permitindo não apenas a estimativa do indicador operacional ESI, mas também a identificação de padrões subjacentes nos dados de telemetria, essenciais para a construção de um sistema robusto de manutenção preditiva.

2.6 Algoritmo K-means

O algoritmo K-means é uma técnica de aprendizado não supervisionado amplamente utilizada para agrupamento de dados (*clustering*), cujo objetivo é particionar um conjunto de observações em grupos que apresentam características semelhantes. Diferentemente de métodos supervisionados, como o MLP utilizado para regressão neste trabalho, o K-means não depende de variáveis-alvo. Ele analisa exclusivamente os próprios padrões dos dados de entrada para identificar estruturas, similaridades e tendências operacionais. Essa característica o torna especialmente útil em contextos nos quais se deseja compreender o comportamento natural do equipamento, estabelecer regimes de operação e diferenciar padrões normais e anômalos sem depender de diagnósticos prévios.

O K-means opera por meio da minimização da soma das distâncias entre os pontos e seus centróides, buscando formar grupos internamente coesos e externamente separados. Dado um número pré-definido de grupos k , o algoritmo inicia selecionando centróides iniciais e alterna entre duas etapas principais: (i) atribuição de cada ponto ao centróide mais próximo e (ii) recomputação dos centróides como a média dos pontos atribuídos. Esse processo é repetido de forma iterativa até que haja convergência. Matematicamente, deseja-se minimizar a seguinte função de custo:

$$J = \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2, \quad (2.5)$$

em que S_i representa o conjunto de pontos pertencentes ao cluster i e μ_i é o centróide correspondente, calculado como a média dos pontos do cluster. O uso da distância Euclidiana ao quadrado como métrica reforça a sensibilidade do algoritmo a variações abruptas nos dados, o que exige cuidados importantes no pré-processamento das variáveis, como normalização e suavização.

No presente estudo, o K-means foi aplicado para identificar padrões operacionais a partir das variáveis de telemetria suavizadas. A utilização de técnicas de suavização, como a média móvel de três registros, torna-se essencial porque os valores de telemetria coletados via rede CAN apresentam flutuações rápidas e ruído intrínseco. Embora reflitam pequenas irregularidades naturais do sistema, essas flutuações não caracterizam mudanças reais no regime operacional. Como o critério central do K-means se baseia na distância entre pontos, oscilações de alta frequência podem induzir agrupamentos artificiais e instáveis. A suavização reduz essas oscilações e permite ao algoritmo captar a tendência real dos dados, favorecendo a formação de clusters mais consistentes e interpretáveis.

Além disso, a escolha do K-means se justifica pelo fato de que as variáveis analisadas apresentam relações não supervisionadas que refletem diferentes modos de operação, como regime leve, moderado e pesado. O K-means é adequado para identificar esses regimes porque sua formulação favorece a separação de grupos com baixa variância interna. Ao final do processo, os centróides podem ser interpretados como “perfis médios de uso”, fornecendo uma visão clara de como o equipamento opera ao longo do tempo. Esses perfis são particularmente úteis para apoiar estratégias de manutenção preditiva, pois permitem correlacionar regimes operacionais com o esforço estimado e com o indicador ESI produzido pelo modelo de aprendizado de máquina.

A escolha do número de clusters foi orientada pela análise rápida em uma sala de controle, para ser algo intuitivo em qual faixa de operação o equipamento está em cada momento. Observou-se que utilizar $k = 3$ forneceu um bom equilíbrio entre simplicidade e expressividade, permitindo distinguir claramente três modos de operação relevantes para a análise: condição leve, condição nominal e condição de alto esforço. Esse particionamento é coerente com a dinâmica natural de sistemas a diesel, nos quais a variação entre estados operacionais tende a se organizar em faixas relativamente estáveis, e não de forma completamente contínua.

Assim, o K-means desempenha um papel fundamental na metodologia deste trabalho. Ele não apenas complementa o modelo MLP ao fornecer uma interpretação qualitativa dos dados operacionais, como também auxilia no entendimento dos padrões de uso que influenciam o desgaste e o esforço aplicado ao equipamento. Ao segmentar o comportamento da máquina em regimes distintos, torna-se possível analisar como diferentes perfis de operação impactam o ESI e, conseqüentemente, desenvolver estratégias de manutenção preditiva mais inteligentes e contextualizadas.

2.7 Trabalhos relacionados

Estudos recentes têm explorado a integração entre Internet das Coisas, inteligência artificial e telemetria veicular para aprimorar estratégias de manutenção preditiva em

frotas automotivas. Esses trabalhos fornecem contribuições significativas que servem de referência para o desenvolvimento deste projeto, o qual se propõe a aplicar tais tecnologias no monitoramento de equipamentos a diesel por meio da rede CAN e da análise preditiva de dados operacionais.

O estudo de [Massaro, Selicato e Galiano \(2019\)](#) propõe uma arquitetura embarcada de baixo custo para monitoramento de frotas de ônibus, composta por uma placa eletrônica inteligente conectada à rede CAN (SAE J1939) e equipada com módulos de comunicação IoT. A solução coleta dados de sensores embarcados e os envia para uma plataforma em nuvem, onde algoritmos de aprendizado de máquina, como redes neurais artificiais (MLP) e agrupamento K-means, realizam previsões de falhas mecânicas e classificações de comportamento de condução. O sistema apresentou resultados promissores, reduzindo o erro médio quadrático das previsões e aumentando a confiabilidade na detecção de anomalias de motor. Os autores destacam a relevância da integração entre dados telemétricos em tempo real e modelos de IA embarcados, demonstrando o potencial da manutenção preditiva em frotas urbanas.

Já o trabalho de [Syed \(2024\)](#) amplia a discussão ao propor um sistema de análise preditiva e diagnóstico em tempo real voltado especificamente para frotas escolares movidas a diesel. A pesquisa aborda a utilização de dados de telemetria de alta frequência (sub-minuto) provenientes de sensores de motor, emissões e posicionamento GPS, processados em um modelo de aprendizado de máquina de código aberto. A solução busca reduzir o tempo ocioso dos veículos, otimizar o consumo de combustível e prever falhas com antecedência suficiente para intervenção programada. O autor demonstra que a aplicação de algoritmos de mineração de dados e regressão em conjuntos extensos de registros de telemetria permite o desenvolvimento de sistemas sustentáveis de manutenção, que aumentam a vida útil do motor e diminuem as emissões poluentes. O estudo reforça também a importância do uso de análise de dados em larga escala (*big data*) e ferramentas de visualização para apoiar decisões operacionais em frotas públicas.

Embora ambos os trabalhos apresentem resultados relevantes, eles concentram-se em contextos de transporte coletivo com foco em frotas de ônibus urbanos e escolares, enquanto o presente projeto amplia o escopo para equipamentos a diesel em ambientes industriais e de mineração, nos quais as condições de operação são significativamente mais severas e os sensores ligados a rede CAN captam parâmetros distintos. Diferentemente das abordagens embarcadas em microcontroladores e ECUs integradas, este trabalho propõe uma arquitetura modular e escalável, capaz de funcionar inicialmente por meio de arquivos CSV simulando o fluxo em tempo real de uma aplicação, facilitando assim a possibilidade de migração futura para streaming MQTT ou integração via API em tempo real.

Assim, a presente pesquisa se diferencia dos estudos anteriores ao buscar uma abordagem generalista e extensível, voltada para a previsão de desgaste do equipamento e

identificar padrões na forma de condução dos operadores de forma contínua, servindo como base para sistemas de gestão de ativos industriais baseados em dados.

3 Metodologia

Este capítulo tem como objetivo detalhar a metodologia usada para viabilizar o uso de técnicas de aprendizado de máquina voltado para manutenção, preditiva com integração a IoT.

3.1 Metodologia proposta

Este estudo tem como objetivo investigar a viabilidade e a eficiência da manutenção preditiva em equipamentos a diesel, com base na coleta de dados operacionais através de sensores de telemetria e a utilização de redes CAN. A escolha dessa abordagem se justifica pela crescente importância das tecnologias da Indústria 4.0, que, através da Internet das Coisas (IoT) e da análise de dados em tempo real, possibilitam a otimização dos processos de manutenção e podendo gerar KPIs para percepções de como é o uso dos operadores e desgaste do equipamento.

A pesquisa foi conduzida utilizando uma abordagem quantitativa e experimental, onde foram coletados dados reais de equipamentos operando em condições típicas de trabalho, como escavadeiras e tratores. A análise preditiva é realizada por meio de algoritmos de aprendizado de máquina, com o objetivo de desenvolver modelos que possam prever falhas e otimizar as intervenções de manutenção. Esses modelos foram validados por meio da comparação de resultados preditivos com dados históricos de falhas e manutenções realizadas em equipamentos similares.

A escolha de uma metodologia experimental permite uma avaliação prática da eficiência da manutenção preditiva, destacando os benefícios e as limitações da aplicação dessa tecnologia em ambientes industriais. O desenvolvimento e a implementação de uma plataforma para coleta, processamento e análise de dados, assim como a aplicação dos modelos preditivos.

3.1.1 Coleta e pré-processamento de dados

A etapa de coleta e pré-processamento de dados consiste em captar os sinais provenientes do barramento CAN do equipamento, que representam variáveis operacionais como rotação, temperatura, tensão e consumo. Após a aquisição, os dados são submetidos a um processo de limpeza e padronização, no qual colunas são uniformizadas, valores ausentes tratados, timestamps consolidados para garantir a coerência temporal das amostras. Por fim, os registros são organizados em micro-lotes, simulando um fluxo contínuo de ingestão

de dados, o que permite preservar a integridade temporal e preparar a base para análises e modelos de aprendizado de máquina em tempo quase real.

3.1.2 Geração de *Features* Derivadas

A etapa de geração de *features* derivadas envolve o enriquecimento do conjunto de dados original por meio do cálculo de médias móveis, desvios padrão, percentis, defasagens (*lags*) e derivadas temporais para cada variável operacional. Esses indicadores estatísticos permitem capturar tendências, flutuações e padrões de comportamento do equipamento ao longo do tempo. Em seguida, todas as variáveis transformadas são consolidadas em um *dataset* unificado (*all_feature.csv*), devidamente estruturado e pronto para as etapas subsequentes de modelagem preditiva e análise de KPIs de desempenho.

3.1.3 Desenvolvimento e Treinamento de Modelos de IA

A fase de desenvolvimento e treinamento de modelos de IA contempla, primeiro, o ajuste de um MLP Regressor para prever o ESI (*Engine stress index*) a partir das *features* operacionais derivadas, capturando relações não lineares e efeitos temporais relevantes. Em paralelo, um algoritmo classificador é treinado para identificar *clusters* de comportamento do operador (*driver clusters*) com base nos descritores estatísticos (médias móveis, desvios, percentis, *lags*), permitindo segmentar padrões de uso que influenciam desgaste e eficiência.

Todos os modelos e respectivos *scalers* (normalizações/padronizações) são registrados no *MLflow*, garantindo rastreamento de experimentos, versionamento de artefatos, reprodutibilidade dos resultados e facilitação da implantação em produção (via *model registry* e *inference pipelines*). Essa governança viabiliza comparações entre execuções, auditoria de métricas e rollbacks controlados durante o ciclo de vida do modelo.

3.1.4 Cálculo de Métricas Derivadas e KPIs

A etapa de cálculo de métricas derivadas e KPIs consiste em gerar, para cada registro de operação, o valor da predição do ESI e o *Efficiency Score*, permitindo quantificar o desempenho operacional em tempo contínuo. Em seguida, todos os KPIs são consolidados no arquivo *kpis.csv*, reunindo o ESI observado e predito, os *clusters* de comportamento do operador, os índices de eficiência operacional e as métricas estatísticas temporais calculadas nas fases anteriores. Essa consolidação possibilita a análise temporal da performance dos equipamentos, a comparação entre operadores e a identificação de padrões recorrentes de operação, que podem ser utilizados para suporte à decisão e otimização das estratégias de manutenção preditiva.

3.1.5 Validação e Análise de Resultados

A etapa de validação e análise de resultados tem como objetivo verificar o desempenho dos modelos desenvolvidos e interpretar seus resultados com foco em correlação em manutenção preditiva. Nessa fase, avalia-se a capacidade do MLP Regressor em reproduzir o ESI heurístico, utilizando métricas quantitativas como *Mean Squared Error* (MSE) e *Root Mean Squared Error* (RMSE) para mensurar a precisão das previsões. Em paralelo, realiza-se a análise da distribuição dos clusters obtidos pelo modelo K-Means, correlacionando-os com indicadores de desempenho operacional dos equipamentos. Essa análise integrada fornece insights sobre a eficiência dos operadores, identifica padrões de estresse do motor e contribui para o aperfeiçoamento das estratégias de manutenção preditiva e otimização de processos, reforçando a confiabilidade e a disponibilidade operacional da frota.

3.2 Dados

Esta seção apresenta os detalhes de como os dados utilizados nesse trabalho foram adquiridos e tratados.

3.2.1 Fonte

Os dados utilizados neste trabalho foram obtidos a partir da telemetria de um equipamento Volvo FMX 500, ano 2023, durante o período de 01/07/2025 até 30/09/2025, pertencente à frota operacional de uma mina cuja produção principal é de ferro. Inicialmente, os registros foram exportados em formato Excel (.xlsx) e, após o pré-processamento, consolidados em arquivos CSV para posterior análise. A coleta das informações foi realizada diretamente pelo barramento CAN-bus dos veículos, garantindo a captura contínua e precisa de variáveis críticas de operação. Cada registro corresponde a uma leitura instantânea do equipamento, incluindo timestamp gerado a cada 30 segundos enquanto o equipamento está com o motor ligado, e a identificação do operador responsável pela operação, permitindo análises temporais detalhadas e a avaliação do desempenho individual.

3.2.2 Pré-processamento de dados

Antes de iniciar a modelagem preditiva, foi necessário realizar um pré-processamento dos dados coletados. Esse procedimento garante que o modelo MLP aprenda padrões válidos e evita que informações inconsistentes comprometam a análise. Na Figura 5, é possível visualizar o conjunto de dados após a etapa de limpeza, evidenciando a padronização e a remoção de registros inadequados.

- Normalização e padronização de colunas.

- Todos os nomes de colunas foram padronizados, removendo espaços, caracteres especiais e quebras de linha.
- O *timestamp* de cada registro foi consolidado na coluna DataHora, obtida a partir da combinação das colunas Data e Hora quando necessário, garantindo que cada registro possua referência temporal única.
- normalizado as grandezas usadas entre 0 e 1 utilizando a função *MinMaxScaler* da biblioteca *scikit-learn* para garantir que a amplitude dos valores não influencie de forma desigual a aprendizagem do modelo.
- Exclusão de colunas desnecessárias.
 - Algumas colunas do arquivo bruto de registro de telemetria, como "Grupo Equipamento" e "Tipo Equipamento", foram consideradas irrelevantes, pois estou fazer a modelagem toda em cima de apenas um modelo de equipamento e para a análise de eficiência do operador.
- Preenchimento de valores ausentes.
 - Os dados coletados apresentavam eventuais valores ausentes ou inconsistentes, principalmente em sinais de sensores devido a leituras perdidas ou falhas temporárias.
 - Foi adotado um método de preenchimento *forward-fill*, onde valores ausentes ou anômalos (por exemplo, leituras muito discrepantes que indicam falhas no sensor) são substituídos pelo último valor válido registrado.

Data	Hora	Operador	Bateria	Equipamento	velocidade	rpm_norm	rpm	odometro	horimetro	temp_motor_norm	temp_motor	consumo_no	consumo	nivel_combustivel	DataHora	esi
01/07/2025	00:00:02	1	28.87	4	0.40	862	47739	550.441	0.25	80	0.30	5.8	0.76	0.76	07/01/2025 00:00	0.3431725493913601
01/07/2025	00:01:11	1	29.09	30	0.45	987	47739	5504.45	0.35	83	0.15	1.1	0.75	0.75	07/01/2025 00:01	0.3610826237310691
01/07/2025	00:02:11	1	28.94	13	0.65	1217	47739	550.446	0.35	82	0.60	23.1	0.75	0.75	07/01/2025 00:02	0.47853157507192573
01/07/2025	00:04:11	1	28.9	4	0.30	602	47739	5504.5	0.35	82	0.25	1.95	0.75	0.75	07/01/2025 00:04	0.28240935944054935

Figura 5 – dados limpos

Fonte: Elaborado pelo autor.

3.3 ESI heurístico

O ESI é um índice heurístico desenvolvido para refletir o esforço do motor e o estresse do equipamento durante a operação. Não é uma medição direta de grandezas físicas, mas sim uma combinação calculada das variáveis operacionais (RPM, consumo, temperatura do motor). O objetivo do ESI é fornecer uma referência quantitativa para:

- Avaliar o esforço do motor em cada ponto do tempo
- Derivar métricas de eficiência do operador

- Servir de *target* para o MLP, permitindo que a IA aprenda a reproduzir esse índice e generalize para novos dados

O cálculo feito para obter o ESI é

$$ESI = 0.5 \times RPM_{\text{norm}} + 0.3 \times Consumo_{\text{norm}} + 0.2 \times TempMotor_{\text{norm}} \quad (3.1)$$

A equação apresentada combina três variáveis operacionais normalizadas, rotação do motor (RPM_{norm}), consumo instantâneo de combustível ($Consumo_{\text{norm}}$) e temperatura do motor ($TempMotor_{\text{norm}}$) para o cálculo do ESI. A utilização das grandezas em formato normalizado garante que todas as variáveis contribuam proporcionalmente para o índice, evitando distorções causadas por diferenças de escala entre as medições.

Este será o alvo em que será treinado o algoritmo de MLP Regressor para fazer sua previsão em tempo real, em função das outras grandezas de entradas

3.4 Geração de recursos

Para garantir que o modelo preditivo fosse treinado em condições similares a um ambiente de operação real, os dados foram processados em micro-lotes de 100 registros. Essa abordagem simula a inserção contínua de dados, característica de um sistema em tempo real, permitindo que o pipeline seja escalável e preparado para streaming futuro via API. Cada lote representa uma sequência temporal de leituras do equipamento, preservando a ordem cronológica e permitindo o cálculo de métricas dependentes de histórico, como médias móveis e derivadas.

3.4.1 Recursos estatísticas derivados

A partir dos sinais coletados e do próprio ESI heurístico, foram geradas diversas features estatísticas e temporais com o objetivo de capturar tendências, variações e picos operacionais. No geral, essas estatísticas foram calculadas a partir de janelas de três registros, uma vez que um único registro é insuficiente para representar o comportamento real do equipamento e pode refletir apenas uma variação momentânea. A janela de três registros corresponde a aproximadamente um minuto e meio a dois minutos de operação, permitindo capturar de forma mais robusta a tendência daquele período e evitando interpretações baseadas em flutuações pontuais. Essas features foram calculadas tanto para os sinais principais quanto para o ESI, permitindo análises de tendência, volatilidade e comportamento operacional. Elas também servem como entrada para o K-Means, que agrupa os registros em clusters representativos do padrão de operação de cada operador.

Tabela 1 – Descrição das features geradas para análise dos sinais

Tipo de Feature	Descrição	Justificativa científica
*_mean_3	Média móvel sobre 3 registros	Captura tendência recente do sinal; reduz ruído momentâneo
*_std_3	Desvio padrão móvel sobre 3 registros	Mede a variabilidade do sinal, permitindo identificar operações instáveis
*_pct90_3	Percentil 90 móvel sobre 3 registros	Detecta picos e condições extremas do equipamento, importantes para avaliar estresse do motor
*_lag_1	Valor defasado de 1 registro	Permite analisar mudanças instantâneas e calcular derivadas
*_derivative	Diferença entre registros consecutivos	Representa a taxa de variação do sinal, capturando acelerações, frenagens e mudanças bruscas de operação

Fonte: Elaborado pelo autor.

3.5 Modelagem

O processo de modelagem preditiva deste trabalho foi estruturado para estimar o ESI a partir das variáveis operacionais do equipamento, bem como identificar padrões de operação dos operadores. Para isso, foram utilizados dois algoritmos de inteligência artificial: o *MLPRegressor*, para predição do ESI, e o *K-Means*, para classificação dos operadores, do caminhão de estudo, com base em *features* estatísticas derivadas.

3.5.1 *Multi-Layer Perceptron Regressor*

O MLP Regressor é uma rede neural *feedforward* que foi utilizada para aprender a relação não linear entre sinais operacionais e o ESI normalizado. Este algoritmo foi selecionado para este trabalho, pois de acordo com [Qin et al. \(2022\)](#), possui capacidade de aprender relações não lineares complexas entre múltiplas variáveis contínuas, característica essencial para modelar a previsão do ESI a partir de sinais operacionais do equipamento. Diferentemente de modelos lineares, o MLP é capaz de capturar interações entre RPM, velocidade, consumo de combustível e temperatura do motor, que apresentam efeitos combinados sobre o estresse do motor. As características do *MLPRegressor* utilizado foram:

- *Target*: ESI heurístico, normalizado entre 0 e 1.
- *Features* de entrada: apenas sinais que refletem o esforço do motor e a operação do equipamento, excluindo hodômetro e horímetro, que são acumulativos.

- Treinamento: o *dataset* foi dividido em conjuntos de treino e teste, mantendo a integridade temporal dos registros.
- Persistência: o modelo treinado e sua escala foram registrados no MLflow, permitindo que previsões futuras possam ser realizadas em tempo real ou em novos datasets.
- Divisão do *dataset*: a base de dados foi separada com 80% usada para treinamento e 20% para teste

A literatura sugere que, para tarefas de regressão com redes MLP, arquiteturas com uma ou duas camadas ocultas contendo dezenas a poucas centenas de neurônios podem apresentar bom equilíbrio entre capacidade de aprendizagem e risco de sobreajuste (RAMCHOUN *et al.*, 2016)

Antes do início do desenvolvimento completo do modelo, foram realizados testes preliminares com diferentes configurações de camadas e neurônios na rede MLP Regressor, com o objetivo de identificar uma arquitetura que equilibrasse precisão e generalização. Os resultados comparativos, apresentados na Tabela 2, indicaram que arquiteturas com duas camadas ocultas apresentaram desempenho superior às de camada única, especialmente a configuração com 64 e 32 neurônios, que obteve um erro quadrático médio (MSE) de aproximadamente 0,00067.

Essa escolha mostrou-se coerente com estudos prévios, como os de Ramchoun *et al.* (2016), que relatam melhor desempenho em redes compostas por duas camadas ocultas com dezenas de neurônios, oferecendo capacidade de aprendizado suficiente sem incorrer em complexidade excessiva. Nos experimentos conduzidos, a arquitetura (64, 32) apresentou redução consistente no erro médio quadrático (MSE) ao longo das validações, sem divergência significativa entre o erro de treino e o de validação, indicando boa capacidade de generalização. Além disso, arquiteturas mais profundas, como a de (128, 64, 32), embora tenham alcançado MSE inferior, não apresentaram um ganho expressivo o bastante para justificar o aumento do custo computacional e o maior risco de sobreajuste associado a modelos com maior número de parâmetros. Assim, a arquitetura (64, 32) foi selecionada por oferecer um equilíbrio adequado entre desempenho, simplicidade e robustez para um conjunto de dados de média complexidade como o derivado da telemetria CAN.

Tabela 2 – Resultados comparativos das arquiteturas MLP testadas.

Arquitetura (camadas ocultas)	Função de Ativação	MSE Médio
(32,)	ReLU	0.00243
(64, 32)	ReLU	0.00067
(128, 64, 32)	ReLU	0.00015
(64,)	ReLU	0.00986
(64, 64)	ReLU	0.00130

Fonte: Elaborado pelo autor.

Tendo em vista os resultados obtidos nos testes preliminares, o modelo final foi configurado com duas camadas ocultas de 64 e 32 neurônios, estrutura escolhida por equilibrar desempenho e generalização, permitindo capturar relações não lineares complexas entre os sinais operacionais e o ESI (Equipment Stress Index) sem elevar excessivamente a complexidade computacional. A função de ativação ReLU (Rectified Linear Unit) foi adotada por apresentar maior eficiência computacional e convergência mais estável durante o treinamento, conforme destacado por [Glorot, Bordes e Bengio \(2011\)](#), que demonstraram o bom desempenho da ReLU em redes profundas ao mitigar o problema do gradiente dissipativo característico de funções sigmóides e hiperbólicas. O número máximo de épocas foi definido em 500, de modo a garantir a convergência do processo de aprendizado. O *target* foi convertido em um *array* unidimensional (*ravel*) para compatibilidade com a função de regressão do MLP. Essas escolhas equilibram capacidade de aprendizado e risco de overfitting, considerando o tamanho do dataset.

O desempenho do MLP foi avaliado utilizando as métricas de erro médio quadrático (MSE) e raiz do erro médio quadrático (RMSE), amplamente utilizadas em regressão contínua.

As expressões matemáticas utilizadas são:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.2)$$

onde:

- n — número total de amostras do conjunto de dados;
- y_i — valor real do ESI da primeira até a i -ésima amostra;
- \hat{y}_i — valor previsto pelo modelo para ESI até a i -ésima amostra;
- $(y_i - \hat{y}_i)^2$ — erro quadrático associado a cada previsão.

3.5.2 *K-Means*

O K-Means é um algoritmo de aprendizado não supervisionado, utilizado para agrupar registros em clusters baseados na similaridade de features. Ele foi usado para identificar padrões operacionais distintos entre operadores, a partir de features estatísticas derivadas do comportamento do equipamento.

O K-Means busca minimizar a soma das distâncias quadráticas entre cada ponto e o centroide do cluster ao qual ele pertence. A equação matemática do algoritmo é representada a seguir

$$(3.3) \quad \arg \min_C \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2$$

onde:

- k — número de clusters (neste trabalho, definido empiricamente como 3);
 - C_i — conjunto de pontos pertencentes ao cluster i ;
 - μ_i — centróide do cluster i ;
 - x — vetor de features (características) de um ponto.
-
- Entrada (X): vetor de *features* estatísticas derivadas de cada *lotes*, incluindo, por exemplo, `RPM_mean_3`, `velocidade_std_3` e `consumo_mean_3`. Essas variáveis representam medidas agregadas de tendência e dispersão dos sinais originais e podem incluir, adicionalmente, derivadas, defasagens (*lags*) e percentis das variáveis principais.
 - Saída (y): índice do *cluster* (0, 1, 2) ao qual cada registro é atribuído, conforme resultado do agrupamento obtido pelo modelo (`driver_cluster`).

O algoritmo foi escolhido por sua simplicidade e interpretabilidade, sendo de fácil implementação, eficiente computacionalmente e capaz de fornecer resultados intuitivos na análise dos clusters. Além disso, apresenta excelente compatibilidade com features estatísticas, lidando naturalmente com médias, desvios e derivadas que refletem o comportamento dos operadores ao longo do tempo. Por fim, oferece suporte direto à análise do operador, uma vez que cada cluster representa um padrão operacional predominante, permitindo avaliar a eficiência e a consistência de cada condutor com base no ESI e nas métricas derivadas.

3.6 Pipeline

O *pipeline* desenvolvido para este trabalho foi estruturado de forma modular e escalável, seguindo o padrão da arquitetura hexagonal (ports and adapters). Essa escolha

garante que a lógica central do sistema seja isolada de dependências externas, como bancos de dados, APIs, arquivos CSV ou sistemas de monitoramento em tempo real, facilitando manutenção, testes e evolução futura.

3.6.1 Arquitetura hexagonal

A arquitetura do sistema foi organizada conforme proposto por (MARTIN, 2017) segundo o paradigma de *Clean Architecture*, dividida em três camadas principais: *Core*, *Ports* e *Adapters*. Essa estrutura visa manter baixo acoplamento e alta coesão entre os componentes, garantindo que mudanças externas não impactem a lógica de modelagem ou o cálculo dos indicadores de desempenho (KPIs).

O *Core* (domínio central) representa o núcleo da aplicação, concentrando toda a lógica de negócio e os processos de aprendizado de máquina relacionados aos indicadores. Essa camada é responsável por executar o pré-processamento dos dados brutos, garantindo sua qualidade e integridade para as etapas subsequentes. Em seguida, realiza a geração de *features* temporais e estatísticas, enriquecendo o conjunto de dados com informações derivadas que capturam padrões de comportamento operacional. O treinamento dos modelos de IA, como o MLP Regressor e o K-Means, também ocorre nesse domínio, permitindo tanto a previsão do ESI quanto a segmentação de perfis de operação. Por fim, esta camada é encarregado de efetuar a predição do ESI em tempo contínuo e o cálculo do *Efficiency Score*, consolidando indicadores fundamentais para o monitoramento da aplicação.

Os *Ports* (interfaces) constituem a camada responsável por definir os contratos de comunicação entre o Core e o ambiente externo, garantindo que a lógica de negócio permaneça independente da infraestrutura. Nessa camada são especificadas as interfaces padronizadas para leitura de dados provenientes de diferentes fontes, como arquivos CSV, APIs ou fluxos de *streaming*, bem como para a gravação de *features* e KPIs gerados pelo processamento interno. Além disso, os *ports* também contemplam as operações de registro e versionamento de modelos de aprendizado de máquina, assegurando integração transparente com ferramentas externas, como o MLflow. Essa abordagem favorece o desacoplamento, a testabilidade e a substituição modular de componentes, conforme os princípios da arquitetura hexagonal.

Os *Adapters* (implementações concretas) constituem a camada responsável por conectar os *ports* do sistema aos serviços e fontes de dados externas, atuando como ponte entre a lógica de negócio e a infraestrutura. Essa camada abrange tarefas como a leitura de micro-lotes de arquivos CSV, que simulam a ingestão contínua de dados da telemetria, a gravação das *features* derivadas em diretórios locais (*data/features*), o registro de modelos e scalers no MLflow, assegurando rastreabilidade e versionamento e a exportação dos KPIs consolidados para arquivos (*data/kpis/kpis.csv*). Essa solução facilita análise e visualização

de desempenho. Essa separação arquitetural desacopla totalmente a lógica de processamento da infraestrutura subjacente, promovendo escalabilidade, facilidade de manutenção e reuso modular dos componentes, conforme os princípios da *Clean Architecture* e da engenharia de *software* orientada a domínios. Essa separação desacopla a lógica de processamento da infraestrutura, promovendo maior escalabilidade, facilidade de manutenção e reuso dos componentes.

3.6.2 Fluxo de dados do *pipeline*

O *pipeline* desenvolvido segue um fluxo linear, porém modular, composto pelas seguintes etapas principais:

1. Ingestão de dados: os registros provenientes do equipamento são lidos em micro-*lotes* de 100 linhas, simulando um ambiente de dados em tempo real.

2. Pré-processamento: inclui a limpeza e padronização das colunas, o tratamento de valores ausentes e a criação do campo de carimbo temporal (`DataHora`).

3. Geração de *features*: cálculo de estatísticas temporais para cada sinal (`*_mean_3`, `*_std_3`, `*_pct90_3`, `*_lag_1`, `*_derivative`) e para o indicador ESI.

4. Treinamento de modelos:

- ***MLP Regressor*:** aprende a reproduzir o ESI heurístico, gerando a predição `pred_esi`;
- ***K-Means*:** identifica padrões operacionais e agrupa os registros em clusters (`driver_cluster`) para medir o estilo de sua condução.

5. *Scoring*: aplicação dos modelos treinados sobre os *lotes* processados, resultando nos valores de `pred_esi` e `efficiency_score`.

6. Persistência:

- Modelos e *scalers* registrados no MLflow;
- KPIs finais armazenados em `kpis.csv`.

3.6.3 Justificativas técnicas

O sistema foi projetado com base em princípios de modularidade e escalabilidade, incorporando conceitos de *micro-loteses*, *features* temporais e arquitetura hexagonal.

Com o objetivo de tentar simular a ingestão de dados em tempo real, utilizando dados históricos, os *micro-loteses* foram criados. Esses são estruturas de processamento que permitem se assemelhar a ingestão contínua de dados em sistemas que operam com fluxos telemétricos, possibilitando o tratamento incremental das informações sem sobrecarregar a memória. Essa abordagem divide o conjunto de dados em pequenos lotes processados sequencialmente, o que reduz o consumo de recursos computacionais e mantém a ordem temporal dos registros, aspecto fundamental para o cálculo de *features* derivadas, como médias móveis e variações instantâneas, e para a análise do comportamento operacional do operador ao longo do tempo. Dessa forma, os *micro-loteses* viabilizam um equilíbrio entre desempenho, precisão temporal e escalabilidade no pipeline.

As *features* temporais representam variáveis derivadas que capturam o comportamento dinâmico e recente dos equipamentos, refletindo tendências, variações e picos operacionais observados ao longo do tempo. Elas são calculadas a partir de janelas deslizantes e estatísticas como médias móveis, desvios padrão e gradientes, permitindo que o modelo aprenda padrões de evolução e resposta do sistema sob diferentes condições de operação. Essas *features* são empregadas de forma integrada tanto pelo MLP Regressor, responsável pela previsão contínua do ESI, quanto pelo K-Means, utilizado na clusterização dos registros com base em comportamento estatístico. Essa abordagem garante consistência e coerência entre as tarefas de predição e segmentação.

A arquitetura hexagonal (*Ports and Adapters*) viabiliza a escalabilidade do sistema ao desacoplar a lógica central da infraestrutura: a mesma aplicação que hoje lê arquivos CSV pode, amanhã, consumir fluxos em tempo real (Kafka, WebSocket ou API) sem qualquer alteração no domínio. Esse desenho também eleva a testabilidade, pois cada módulo de ingestão, geração de *features*, treinamento e scoring é isolado, por interfaces, permitindo testes unitários e mocks de dependências externas de maneira simples e confiável.

Além disso, a arquitetura favorece a reprodutibilidade do ciclo de *Machine Learning Operations* (MLOps) ao padronizar a persistência de modelos, scalars e KPIs no MLflow e em artefatos CSV, garantindo rastreabilidade de versões, comparação transparente de execuções e implantação controlada em produção. Em conjunto, esses princípios reduzem o custo de mudança, aceleram a evolução do produto e reforçam a confiabilidade operacional do *pipeline*.

3.7 Ferramentas e Bibliotecas Utilizadas

O desenvolvimento do pipeline proposto neste trabalho foi sustentado por um ecossistema bibliotecas *Python* voltadas para ciência de dados, aprendizado de máquina e engenharia de software. O gerenciamento de dependências, ambiente virtual e versionamento de pacotes foi realizado por meio da ferramenta *Poetry*, que garante reprodutibilidade e isolamento do ambiente de desenvolvimento.

3.7.1 Gerenciamento de Ambiente – *Poetry*

O Poetry é uma ferramenta moderna de gerenciamento de pacotes e ambientes virtuais para projetos em Python, projetada para oferecer maior controle, segurança e reprodutibilidade em aplicações de desenvolvimento e pesquisa. Sua utilização neste projeto possibilitou o controle de versões, por meio do registro automático de todas as dependências no arquivo `pyproject.toml`, garantindo que o ambiente de execução possa ser reproduzido integralmente em qualquer outro sistema. Além disso, o Poetry assegura o isolamento do ambiente, uma vez que cada dependência é instalada em um ambiente virtual próprio, evitando conflitos entre versões de bibliotecas.

3.7.2 Bibliotecas Utilizadas

As bibliotecas selecionadas foram escolhidas por sua maturidade, compatibilidade e relevância científica para análise de dados, modelagem e registro de experimentos. A biblioteca **Pandas (v2.3.3)** foi utilizada para manipulação e análise de dados tabulares, sendo responsável por operações de limpeza, filtragem, junção e agregação de dados provenientes dos arquivos CSV. Sua interface *DataFrame* foi amplamente empregada no pré-processamento e na geração das *features* estatísticas conforme, descrito na documentação oficial da biblioteca ([THE PANDAS DEVELOPMENT TEAM, 2025](#)). A **NumPy (v1.26)**, por sua vez, é uma biblioteca fundamental para operações matemáticas vetorizadas e manipulação de *arrays*, servindo de base para os cálculos estatísticos e operações de alto desempenho realizadas nos módulos de *features* e modelagem ([NUMPY DEVELOPERS, 2024](#)).

A **Scikit-learn (v1.5)** foi a principal biblioteca de aprendizado de máquina utilizada neste trabalho, sendo responsável pela implementação dos dois algoritmos centrais do projeto: o **MLPRegressor**, um modelo de rede neural artificial utilizado para prever o ESI, e o **K-Means**, um algoritmo de aprendizado não supervisionado para a clusterização dos padrões de operação dos motoristas. Além disso, o **Scikit-learn** foi empregado para normalização dos dados (usando *MinMaxScaler* e *StandardScaler*) e cálculo das métricas de desempenho ([SCIKIT-LEARN DEVELOPERS, 2024](#)).

O **MLflow (v2.22.2)**, uma ferramenta de gerenciamento e rastreamento de experimentos de *machine learning* (MLOps), foi utilizado para registrar e versionar os modelos

treinados (**MLP** e **K-Means**), armazenar parâmetros, métricas (como *MSE*) e artefatos associados, além de controlar as versões em produção e o histórico de execuções, garantindo rastreabilidade científica. O **MLflow** foi configurado localmente, permitindo a criação automática de experimentos e o registro dos modelos ([MLFLOW CONTRIBUTORS, 2025](#)).

Para a visualização e análise gráfica dos resultados, foi empregada a biblioteca **Matplotlib (v3.8)**, possibilitando a geração de gráficos estatísticos, bem como a análise de correlações e comparações entre operadores e *clusters*. A biblioteca foi utilizada principalmente nas etapas de análise descritiva e apresentação dos resultados, em razão de sua flexibilidade, sintaxe acessível e ampla aceitação na literatura científica ([MATPLOTLIB DEVELOPMENT TEAM, 2024](#)).

O **SciPy (v1.13)**, complementar ao **NumPy**, foi empregado para operações matemáticas avançadas, suporte à geração de estatísticas e cálculos de derivadas temporais das variáveis. Já o **PyArrow (v16.0)** foi utilizado para suporte à leitura e escrita de grandes volumes de dados e compatibilidade futura com ambientes distribuídos e *DataFrames* otimizados ([SCIPY DEVELOPERS, 2024](#)).

4 Resultados

O presente capítulo apresenta os resultados obtidos a partir do pipeline desenvolvido, incluindo a análise do ESI, a clusterização dos operadores e o *Efficiency Score*. O objetivo é demonstrar, de forma quantitativa e visual, o desempenho operacional do equipamento e dos operadores, validando o modelo de esforço do equipamento e destacando padrões relevantes de operação.

Durante o período analisado, entre 01/07/2025 e 30/09/2025, o equipamento foi operado por 37 operadores diferentes, permitindo avaliar não apenas o comportamento do equipamento, mas também a variação de eficiência e padrões de operação entre os operadores.

4.1 Avaliação do MLP Regressor

A avaliação do modelo MLP Regressor tem como objetivo verificar sua capacidade de reproduzir o comportamento do índice heurístico de estresse do motor (ESI), quantificando a precisão do aprendizado e a consistência temporal das previsões.

4.1.1 Métricas de Desempenho

Os resultados obtidos indicaram um $MSE = 0.0018$, correspondendo a um $RMSE \approx 0.042$, o que representa um erro médio de aproximadamente 4,2% em relação à escala normalizada (0–1). Em tarefas de regressão contínua, valores de erro abaixo de 5% são usualmente interpretados como indicadores de boa capacidade preditiva. Assim, o desempenho observado caracteriza um nível de erro baixo, evidenciando que o modelo conseguiu capturar de forma consistente as dinâmicas subjacentes à variação do índice de estresse do motor.

Na Figura 6 foi feito um ajuste para uma escala de 400 para 1, com objetivo de possibilitar uma visualização mais clara. Assim se pode observar pequenas variações entre o valor real e a saída do modelo.

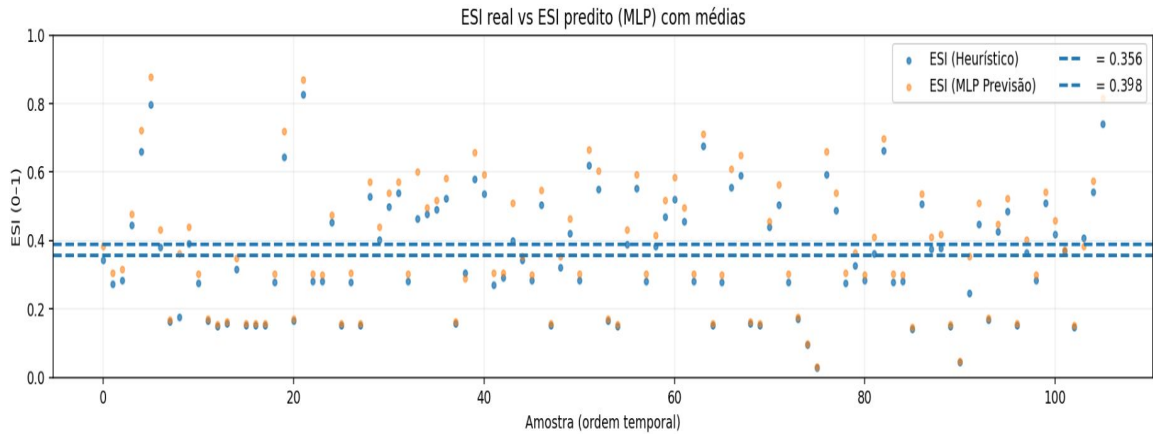


Figura 6 – ESI real vs. ESI predito com médias

Fonte: Elaborado pelo autor.

A observação do comportamento do modelo, em que o descolamento entre os valores heurísticos e previstos aumenta à medida que o ESI se eleva, é um ponto importante a ser analisado. A crescente diferença entre os valores previstos e os valores heurísticos pode ser atribuída a características não lineares dos dados à medida que o equipamento se aproxima de condições operacionais mais extremas, onde o comportamento do sistema torna-se mais complexo e difícil de modelar. Isso pode indicar que, em valores mais altos de *ESI*, o modelo precisa de mais dados representativos e variação nos parâmetros para captar adequadamente os padrões de funcionamento, o que talvez não esteja sendo totalmente representado no conjunto de treinamento.

Além disso, o fato de o modelo mostrar uma tendência para errar para mais (superestimação) quando o ESI ultrapassa 0,4 pode ser um reflexo da distribuição dos dados no conjunto de treino. Em modelos como o *MLP*, quando os dados para certas faixas de valores são limitados ou mal distribuídos, o modelo tende a generalizar de maneira tendenciosa. Isso significa que o modelo pode estar projetado para prever o valor médio de ESI de maneira mais conservadora em faixas mais altas, já que não há dados suficientes ou representativos de falhas reais nessas condições mais extremas. Uma possível solução para mitigar esse viés seria realizar um reajuste do modelo, talvez alterando a distribuição do conjunto de dados para incluir mais exemplos de condições críticas ou ajustando os pesos do modelo para essas faixas de valores mais elevadas.

4.1.2 Análise de operadores

A análise dos valores de ESI mostra que os operadores causam níveis médios de estresse do motor entre 0,26 e 0,54, com picos individuais chegando a 0,89 em alguns casos. Essa variação indica que, embora a maioria opere dentro de uma faixa intermediária de esforço do motor, há diferenças perceptíveis de desempenho entre os condutores.

De forma geral, o conjunto indica uma tendência de ESI moderado, com predominância de valores médios em torno de 0,43–0,46, o que sugere boa estabilidade operacional da frota. Os casos extremos, tanto de ESI muito alto quanto muito baixo, são pontuais e úteis para análises específicas de comportamento de condução. Assim, o ESI se mostra uma métrica eficaz para comparar a eficiência operacional entre operadores e identificar potenciais oportunidades de treinamento ou ajuste de parâmetros de operação, contribuindo para a redução do desgaste mecânico e aumento da vida útil dos equipamentos.

Na Tabela 3 são apresentados os 4 *outliers*, os dois com a maior média e os dois com a menor média, dentre os 37 operadores do período analisado.

Tabela 3 – Operadores com maiores e menores valores médios de ESI

Operador	ESI médio	ESI máximo	Categoria
Operador 1	0.543	0.889	Maior
Operador 2	0.533	0.871	Maior
Operador 3	0.262	0.617	Menor
Operador 4	0.226	0.584	Menor

Fonte: Elaborado pelo autor.

4.1.3 Pontuação de eficiência individual

Além do ESI heurístico, foi definida uma grandeza derivada denominada *Efficiency Score*, destinada a quantificar a eficiência relativa do operador em termos de estresse imposto ao motor durante a operação.

O *Efficiency Score* é calculado a partir do valor da predição do ESI (*pred_esi*) gerado pelo modelo MLP, conforme a expressão 4.1 :

$$\text{Efficiency Score} = 100 - (\text{pred_esi} \times 100) \quad (4.1)$$

O *Efficiency score* é interpretado da seguinte forma:

- Valores próximos a 100 indicam operação eficiente, com menor estresse do motor;
- Valores menores indicam maior esforço ou operação mais agressiva, refletindo aumento de estresse mecânico.

Enquanto o ESI fornece um indicador contínuo do esforço aplicado ao motor, o *Efficiency Score* converte essa informação em uma métrica mais intuitiva e comparável entre operadores. Essa transformação possibilita:

- A análise direta da eficiência de cada operador ao longo do tempo;
- A complementação dos *clusters* obtidos pelo algoritmo K-Means;
- A facilitação da visualização e interpretação científica dos resultados.

A introdução do *Efficiency Score* permite avaliar a precisão do modelo preditivo, além de fornecer *insights* operacionais concretos, relacionando o comportamento do operador ao esforço imposto ao equipamento de forma quantitativa.

o *EfficiencyScore* médio dos operadores foi 0,636 e será importante para fazer medições e análises quando essa aplicação for levada para um sistema em tempo real.

4.2 Análise de clusters *K-means*

Após a etapa de modelagem supervisionada, foi aplicada uma abordagem não supervisionada utilizando o algoritmo *K-Means* para identificar agrupamentos naturais nos dados operacionais. O objetivo principal desta etapa é caracterizar diferentes perfis de condução e padrões de uso do caminhão, permitindo uma análise comportamental dos operadores e das condições de operação do motor a partir de variáveis derivadas do sinal da rede CAN.

4.2.1 Definição e Critério de Escolha do Número de Clusters

O modelo K-Means foi configurado para gerar três clusters ($k = 3$). Esse número foi escolhido com base em uma combinação de critérios de interpretabilidade operacional.

Testes iniciais com valores de k entre 2 e 5 indicaram que a configuração com três agrupamentos representava de forma adequada a variabilidade dos comportamentos de condução, evitando a segmentação excessiva dos dados e preservando a coerência estatística entre os grupos.

Sob a ótica da gestão de frotas, a divisão em três clusters mostrou-se suficiente para representar, de maneira clara e interpretável, os principais perfis de operação observados, conforme a Tabela 4 :

Tabela 4 – Descrição dos clusters operacionais identificados pelo algoritmo K-Means

Cluster	Descrição
0 – leve	Operação em baixa rotação e velocidade, com carga reduzida.
1 – inter-mediário	Operação estável, dentro das faixas normais de carga e temperatura.
2 – severo	Operação em altos níveis de RPM, consumo e temperatura, indicando possível sobrecarga mecânica ou condução agressiva.

Assim, a definição de três agrupamentos mantém o equilíbrio entre a interpretação prática e a representatividade estatística, permitindo uma segmentação funcional do comportamento operacional dos equipamentos.

4.2.2 Resultados dos *clusters*

Na Figura 7 se observa que a maioria dos registros está na no cluster 1, seguindo da 0 e depois da 2.

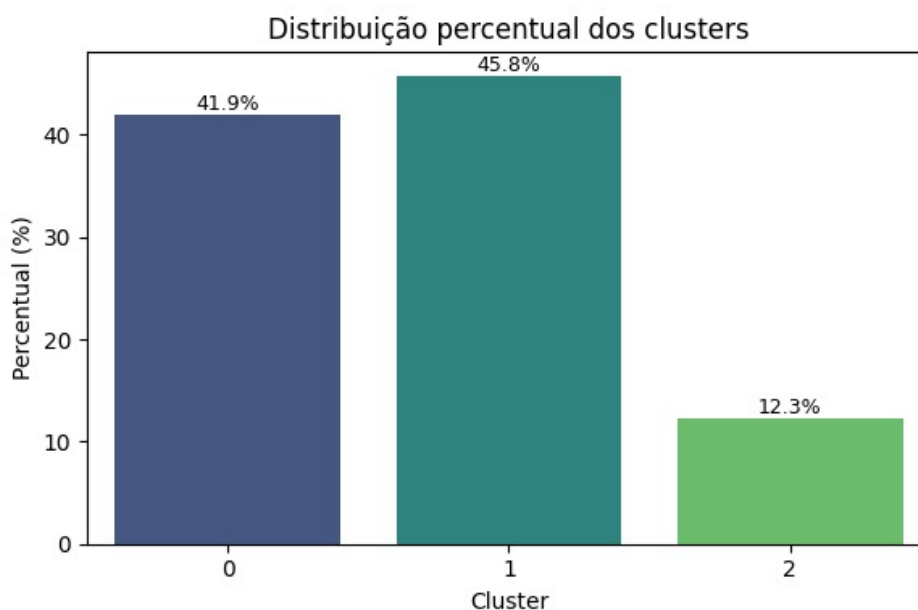


Figura 7 – Porcentagem de registros de cada cluster

Fonte: Elaborado pelo autor.

A Figura 8 apresenta a distribuição dos valores de ESI para cada cluster identificado. Percebe-se uma relação crescente entre o ESI e os clusters, indicando que, conforme o equipamento realiza maior esforço, tende a ser classificado em clusters de níveis mais elevados. Esse comportamento confirma a coerência do agrupamento com o propósito da análise, refletindo a relação esperada entre esforço operacional e ESI.

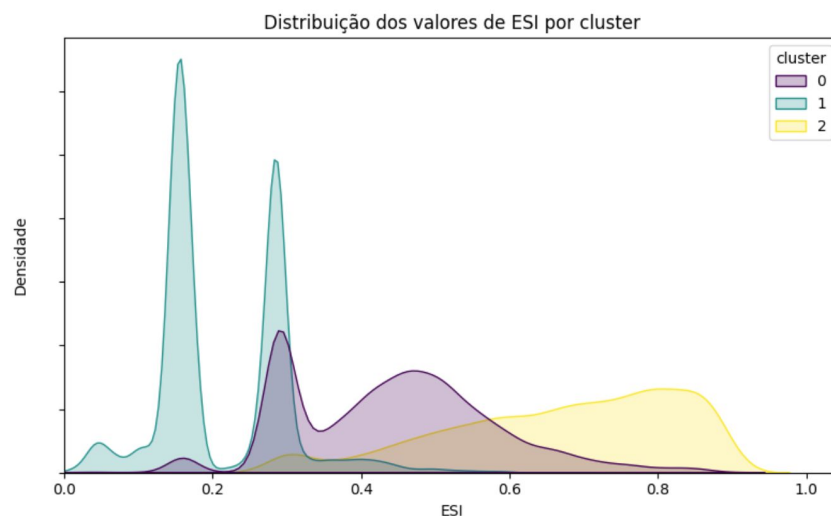


Figura 8 – Gráfico de distribuição do registro de cluster e ESI

Fonte: Elaborado pelo autor.

4.2.3 Aplicação dos *cluster*

Os resultados obtidos pelos clusters oferecem uma base analítica valiosa para o monitoramento e a gestão operacional da frota. A segmentação permite acompanhar em tempo real o comportamento de condução, associando cada operador ao cluster correspondente conforme seu padrão de operação. Além disso, possibilita a detecção precoce de anomalias, especialmente quando um operador transita com frequência para o cluster severo, indicando condições de uso potencialmente prejudiciais ao motor.

Essas informações também podem orientar programas de treinamento operacional, ao identificar estilos de condução menos eficientes e promover boas práticas que reduzam o consumo e o desgaste mecânico. Por fim, os clusters servem como suporte à priorização da manutenção preventiva, uma vez que permitem correlacionar a carga operacional de cada grupo com o risco de falhas, contribuindo para a otimização de recursos e aumento da confiabilidade dos equipamentos.

4.3 Modelos

Após o treinamento e desenvolvimento dos dois modelos utilizados nesse trabalho sobre aprendizado máquina: o MLP regressor e o *K-Means Clustering*. Ambos foram implementados utilizando a biblioteca **Scikit-learn** e configurados de modo a atender os objetivos de predição e agrupamento definidos.

O *MLPRegressor* foi responsável por estimar o ESI a partir das variáveis operacionais derivadas e normalizadas, gerando o valor predito (*pred_esi*). Já o algoritmo *K-Means* foi empregado para identificar padrões de operação e agrupar os registros em três regimes distintos de uso leve, intermediário e severo de acordo com as características estatísticas das variáveis.

Após o processo de treinamento, ambos os modelos foram salvos em formato **.pkl** (*pickle*), permitindo sua reutilização em etapas futuras sem a necessidade de reexecução do treinamento. O uso desse formato oferece diversas vantagens, tais como a **reprodutibilidade**, que garante que os mesmos pesos, parâmetros e hiperparâmetros utilizados no treinamento original sejam preservados, assegurando resultados consistentes entre diferentes execuções. Além disso, a **eficiência** é aprimorada, pois evita o retrabalho de reentrenar os modelos sempre que uma nova análise é necessária, reduzindo significativamente o tempo de processamento. O formato também facilita a **integração**, permitindo a incorporação dos modelos em sistemas externos ou pipelines de produção, como aplicações de monitoramento em tempo real, APIs ou dashboards analíticos. Por fim, a **compatibilidade com o MLflow** é garantida, pois o formato **.pkl** é totalmente compatível com o ambiente de rastreamento de experimentos utilizado, permitindo registrar os modelos, métricas e artefatos de forma padronizada e versionada.

Com os modelos devidamente salvos e versionados, torna-se possível realizar a etapa de *scoring*, aplicando o *MLPRegressor* e o *K-Means* sobre novos conjuntos de dados processados. Essa abordagem garante a continuidade do processo de análise e possibilita a geração automática de indicadores como o *pred_esi*, o *Efficiency Score* e os *KPIs* consolidados do sistema.

5 Conclusões e considerações para trabalhos futuros

O objetivo deste trabalho, conforme apresentado na introdução, foi o desenvolvimento de um modelo preditivo baseado em aprendizado de máquina, utilizando dados extraídos da rede CAN de um caminhão de carga, com o propósito de avaliar o estresse do motor e o modo de condução dos operadores, fornecendo indicadores contínuos de desempenho e contribuindo para a otimização das operações em termos de segurança, produtividade e manutenção preditiva.

Os resultados obtidos demonstraram que o objetivo proposto foi alcançado de forma satisfatória. O modelo *MLPRegressor* apresentou um erro quadrático médio (RMSE) de 4,2%, indicando boa precisão na predição do *ESI* heurístico. Além disso, o algoritmo *K-Means* foi capaz de agrupar adequadamente os operadores em três regimes distintos de operação leve, intermediário e severo refletindo diferentes padrões de condução e níveis de esforço do motor.

As estimativas de *ESI* obtidas com a MLP, da forma como foram apresentadas neste trabalho, são um primeiro passo. Com o aprimoramento da abordagem aqui implementada, objetiva-se, com a MLP, prever o *ESI* n-passos à frente, de forma a poder prever como o *ESI* deverá evoluir à medida que o tempo passa. Tal ferramenta poderá, por exemplo, alertar o operador da máquina de um possível futuro sobreaquecimento do motor caso ele siga operando a máquina da forma atual.

A abordagem proposta mostrou-se eficiente ao integrar técnicas de pré-processamento, geração de *features* estatísticas e modelagem preditiva dentro de uma arquitetura modular e escalável. O uso do *MLflow* para rastreamento de experimentos e o salvamento dos modelos em formato *.pkl* garantiram reprodutibilidade, eficiência computacional e facilidade de integração com aplicações futuras.

Com a disponibilidade de um volume maior de dados e variáveis relacionadas ao desgaste real dos componentes, seria possível desenvolver diretamente um modelo de manutenção preditiva. Entretanto, o modelo proposto neste trabalho pode servir como uma etapa preliminar a ser complementada em pesquisas futuras, auxiliando na identificação de padrões de operação e no cálculo de indicadores que alimentem e aprimorem modelos preditivos mais robustos.

Como trabalhos futuros, recomenda-se a integração do *pipeline* com fluxos de dados em tempo real, por meio de protocolos como Kafka ou WebSocket, bem como a ampliação do conjunto de variáveis monitoradas. A incorporação de novos algoritmos de aprendizado

de máquina, tais como *Random Forest*, Árvores de Decisão e classificadores baseados em *Multilayer Perceptron* (MLP), também representa uma direção promissora para aprimorar a interpretabilidade e o desempenho do sistema.

Referências

AQUEVEQUE, P. *et al.* Data-driven condition monitoring of mining mobile machinery in non-stationary operations using wireless accelerometer sensor modules. *IEEE Access*, v. 9, p. 12916–12927, 2021. DOI: [10.1109/ACCESS.2021.3052807](https://doi.org/10.1109/ACCESS.2021.3052807). Citado 2 vezes nas páginas 11, 12.

AVATEFIPOUR, O.; SARGOLZAEI, A.; ABDELGHANI, M. State-of-the-Art Survey on In-Vehicle Network Communication (CAN, LIN, FlexRay, and Ethernet) in Modern Vehicles. *arXiv preprint arXiv:1811.06292*, 2018. Disponível em: <https://arxiv.org/abs/1811.06292>. Citado 2 vezes nas páginas 27, 30.

BOLAND, Hannah M. *et al.* An Overview of CAN-BUS Development, Utilization, and Future Potential in Serial Network Messaging for Off-Road Mobile Equipment. *In: TECHNOLOGY in Agriculture*. IntechOpen, 2021. DOI: [10.5772/intechopen.98444](https://doi.org/10.5772/intechopen.98444). Disponível em: https://www.researchgate.net/publication/353499375_An_Overview_of_CAN-BUS_Development_Utilization_and_Future_Potential_in_Serial_Network_Messaging_for_Off-Road_Mobile_Equipment. Citado 2 vezes nas páginas 21, 28.

CORRIGAN, Steve. *Introduction to the Controller Area Network (CAN)*. 2016. Disponível em: <https://www.ti.com/lit/an/sloa101b/sloa101b.pdf>. Citado 10 vezes nas páginas 21–27.

ETSCHBERGER, Konrad. *Controller Area Network: Basics, Protocols, Chips and Applications*. IXXAT Press, 2001. ISBN 9783000073765. Citado 2 vezes nas páginas 21, 27.

FARIA, B.O.; RAPOSO, F.E.A. Proposição de um plano de manutenção preditiva para motores a diesel de máquinas carregadeiras que executam serviço de apoio à atividade de mineração. *UFES Repository*, 2023. Disponível em: https://repositorio.ifes.edu.br/bitstream/handle/123456789/3324/TCC_PROPOSI%C3%87%C3%83O_PLANO_MANUTEN%C3%87%C3%83O_PREDITIVA_MOTORES_DIESEL.pdf?sequence=3. Citado 4 vezes nas páginas 15–17.

GADIRAJU, R. K. AI-Powered Telemetry for Predictive Maintenance in Enterprise Devices. *International Journal of Intelligent Systems and Applications in Engineering*, v. 12, 23s, p. 3063–3070, 2024. Citado 1 vez na página 29.

GLOROT, Xavier; BORDES, Antoine; BENGIO, Yoshua. Deep Sparse Rectifier Neural Networks. *In: PROCEEDINGS of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*. 2011. v. 15. (Journal of Machine Learning Research: Workshop and Conference Proceedings), p. 315–323. Disponível em: <http://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf>. Citado 2 vezes nas páginas 31, 43.

IBRAM – INSTITUTO BRASILEIRO DE MINERAÇÃO. *5 dicas para manutenção eficaz em mineradoras*. 2018. <https://ibram.org.br/noticia/5-dicas-para-manutencao-eficaz-em-mineradoras/>. Acesso em: 18 jan. 2025. Citado 1 vez na página 11.

KRISTJANPOLLER, F. *et al.* Equivalent availability index for the performance measurement of haul truck fleets. *Eksploatacja i Niezawodność - Maintenance and Reliability*, Polônia, v. 22, n. 4, 2020. Citado 1 vez na página 11.

LAWRENZ, Wolfhard. *CAN System Engineering: From Theory to Practical Applications*. Springer, 2013. ISBN 9783642011681. DOI: [10.1007/978-3-642-01168-1](https://doi.org/10.1007/978-3-642-01168-1). Citado 3 vezes nas páginas 20, 25, 28.

LEDMAOUI, Y. *et al.* Review of recent advances in predictive maintenance and cybersecurity for solar plants. *Sensors*, v. 25, n. 1, p. 206, 2025. DOI: [10.3390/s25010206](https://doi.org/10.3390/s25010206). Citado 1 vez na página 12.

MAGALHÃES, N. Análise das manutenções preventivas e proposta de melhoria ao planejamento e controle de manutenção de equipamentos HME em uma empresa de mineração. *UFOP Repository*, 2021. Disponível em: <https://monografias.ufop.br/handle/35400000/3676>. Citado 2 vezes nas páginas 16, 17.

MARINS, Raphael Teixeira. *Estudo das contribuições da manutenção preditiva em equipamentos de mineração para consolidação de uma base teórica para área da manutenção*. 2024. Diss. (Mestrado) – Universidade Federal de Ouro Preto. Disponível em: https://monografias.ufop.br/bitstream/35400000/7031/3/MONOGRAFIA_EstudoContribui%C3%A7%C3%B5esManuten%C3%A7%C3%A3o.pdf. Citado 3 vezes nas páginas 18, 20.

MARQUES DA COSTA, Rafael; PAULO SACHETTO, João. PLANEJAMENTO DA MANUTENÇÃO PREVENTIVA EM TORNO MECÂNICO ENGRENADO. *Ciência Tecnologia*, v. 15, n. 1, e1519, dez. 2023. DOI: [10.52138/citec.v15i1.329](https://doi.org/10.52138/citec.v15i1.329). Disponível em: <https://publicacoes.fatecjaboticabal.edu.br/citec/article/view/329>. Citado 1 vez na página 18.

MARTIN, Robert C. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Pearson, 2017. Citado 1 vez na página 45.

MASSARO, Alessandro; SELICATO, Sergio; GALIANO, Angelo. Predictive Maintenance of Bus Fleet by Intelligent Smart Electronic Board Implementing Artificial Intelligence. *IoT*, v. 1, n. 2, p. 180–197, 2019. DOI: [10.3390/iot1020012](https://doi.org/10.3390/iot1020012). Disponível em: <https://www.mdpi.com/2624-831X/1/2/12>. Citado 1 vez na página 34.

MATPLOTLIB DEVELOPMENT TEAM. *Matplotlib: A 2D Graphics Environment*. 2024. Disponível em: <https://matplotlib.org/>. Citado 1 vez na página 49.

MLFLOW CONTRIBUTORS. *MLflow: An Open Platform for the Machine Learning Lifecycle*. 2025. Disponível em: <https://mlflow.org/>. Citado 1 vez na página 49.

NUMPY DEVELOPERS. *NumPy: Fundamental package for scientific computing with Python*. 2024. Disponível em: <https://numpy.org/>. Citado 1 vez na página 48.

PENHA, Daniele Costa. *Processo de planejamento e programação de manutenção em uma empresa de mineração em São Luís do Maranhão: estudo de caso*. 2021. Diss. (Mestrado) – Universidade Federal do Pará. Disponível em: https://repositorio.ufpa.br/jspui/bitstream/2011/13695/1/Dissertacao_ProcessoPlanejamentoProgramacao.pdf. Citado 1 vez na página 16.

PENHA, Daniele Costa; BATISTA, Clauderino da Silva; VIEGAS, Bruno Marques. Processo de planejamento e programação de manutenção em uma empresa de mineração em São Luis do Maranhão: estudo caso / Planning and scheduling process of a major maintenance stop in a mining company in São Luis do Maranhão: case study. *Brazilian Journal of Development*, v. 7, n. 7, p. 72326–72346, jul. 2021. DOI: [10.34117/bjdv7n7-419](https://doi.org/10.34117/bjdv7n7-419). Disponível em: <https://ojs.brazilianjournals.com.br/ojs/index.php/BRJD/article/view/33145>. Citado 5 vezes nas páginas 15, 17, 18.

QIN, Yongfei *et al.* MLP-Based Regression Prediction Model For Compound Bioactivity. *Frontiers in Bioengineering and Biotechnology*, Volume 10 - 2022, 2022. ISSN 2296-4185. DOI: [10.3389/fbioe.2022.946329](https://doi.org/10.3389/fbioe.2022.946329). Disponível em: <https://www.frontiersin.org/journals/bioengineering-and-biotechnology/articles/10.3389/fbioe.2022.946329>. Citado 1 vez na página 41.

RAKHOLIA, R. *et al.* Integrating AI and IoT for Predictive Maintenance in Industry 4.0. *Information*, v. 16, n. 9, p. 737, 2025. DOI: [10.3390/info16090737](https://doi.org/10.3390/info16090737). Citado 1 vez na página 29.

RAMCHOUN, Hassan *et al.* Multilayer Perceptron: Architecture Optimization and Training. *International Journal of Interactive Multimedia and Artificial Intelligence*, v. 4, n. 1, p. 26–30, 2016. DOI: [10.9781/ijimai.2016.415](https://doi.org/10.9781/ijimai.2016.415). Disponível em: https://www.ijimai.org/journal/sites/default/files/files/2016/02/ijimai20164_1_5_pdf_30533.pdf. Citado 2 vezes na página 42.

SCIKIT-LEARN DEVELOPERS. *Scikit-learn: Machine Learning in Python*. 2024. Disponível em: <https://scikit-learn.org/>. Citado 1 vez na página 48.

SCIPY DEVELOPERS. *SciPy: Open Source Scientific Tools for Python*. 2024. Disponível em: <https://scipy.org/>. Citado 1 vez na página 49.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. *Administração da Produção*. São Paulo: Atlas, 2002. Citado 1 vez na página 11.

SYED, Shakir. Enhancing School Bus Engine Performance: Predictive Maintenance and Analytics for Sustainable Fleet Operations. *Library Progress International*, v. 44, n. 3, p. 17765–17775, 2024. ISSN 0970-1052. Disponível em: <https://www.bpasjournals.com/LibraryProgressInternational>. Citado 1 vez na página 34.

TEIXEIRA, Humberto Nuno; LOPES, Isabel; BRAGA, Ana Cristina. Condition-based maintenance implementation: a literature review. *Procedia Manufacturing*, v. 51, p. 228–235, 2020. 30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021). ISSN 2351-9789. DOI: <https://doi.org/10.1016/j.promfg.2020.10.033>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2351978920318886>. Citado 3 vezes na página 19.

THE PANDAS DEVELOPMENT TEAM. *pandas: Powerful data structures for data analysis, time series, and statistics*. 2025. Disponível em: <https://pandas.pydata.org/>. Citado 1 vez na página 48.

XENOS, C. *Gestão da manutenção: função estratégica*. Editora Qualitymark, 1998. Citado 1 vez na página 20.

ZHENG, Haining; PAIVA, Antonio R.; GURCIULLO, Chris S. Advancing from Predictive Maintenance to Intelligent Maintenance with AI and IIoT. *arXiv preprint arXiv:2009.00351*, set. 2020. DOI: [10.48550/arXiv.2009.00351](https://arxiv.org/abs/2009.00351). Disponível em: <https://arxiv.org/abs/2009.00351>. Citado 1 vez na página 12.