PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
Escola Politécnica
Programa de Pós-Graduação em Engenharia Mecânica-PPGEM

BERNARDO LUIS TULESKI

APRENDIZADO DE MÁQUINA APLICADO AO DIAGNÓSTICO DE FALHAS EM MOTORES DE IGNIÇÃO POR COMPRESSÃO USANDO SINAIS DE ÁUDIO

CURITIBA
Setembro – 2023

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
Escola Politécnica
Programa de Pós-Graduação em Engenharia Mecânica-PPGEM

BERNARDO LUIS TULESKI

APRENDIZADO DE MÁQUINA APLICADO AO DIAGNÓSTICO DE FALHAS EM MOTORES DE IGNIÇÃO POR COMPRESSÃO USANDO SINAIS DE ÁUDIO

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Engenharia Mecânica, Curso de Pós-Graduação em Engenharia Mecânica, Escola Politécnica, Pontifícia Universidade Católica do Paraná.

Orientador: Profa. Dra. Viviana Cocco Mariani

CURITIBA
Setembro - 2023

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central
Sônia Maria Magalhães da Silva – CRB 9/1191

T917a
2023

Tuleski, Bernardo Luis
Aprendizado de máquina aplicado ao diagnóstico de falhas em motores de ignição por compressão usando sinais de áudio / Bernardo Luis Tuleski ; orientadora: Viviana Cocco Mariani. – 2023
110 f. ; il. : 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2023
Bibliografia: f. 106-110

1. Indústria automobilística. 2. Automóveis – Ignição. 3. Máquina de vetor suporte. 4. Engenharia de produção. I. Mariani, Viviana Cocco. II. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Engenharia Mecânica. III. Título.

CDD. 20. ed. – 620.1

Pontifícia Universidade Católica do Paraná–PUCPR
Escola Politécnica
Programa de Pós-Graduação em Engenharia Mecânica-PPGEM

TERMO DE APROVAÇÃO

Bernardo Luís Tuleski

Aprendizado de Máquina Aplicado ao Diagnóstico de Falhas em Motores de Ignição por Compressão Usando Sinais de Áudio

Dissertação aprovada como requisito parcial para obtenção do grau de mestre no Curso de Mestrado em Engenharia Mecânica, Programa de Pós-Graduação em Engenharia Mecânica, da Escola Politécnica da Pontifícia Universidade Católica do Paraná, pela seguinte banca examinadora:

Prof. Dr. Stephan Hennings Och
Universidade Federal do Paraná, UFPR, Relator

Prof. Dr. Gideon Villar Leandro
Universidade Federal do Paraná, UFPR

Prof. Dr. Alceu de Souza Brito Junior
Pontifícia Universidade Católica do Paraná, PUCPR

Prof. Dr. Leandro dos Santos Coelho
Pontifícia Universidade Católica do Paraná, PUCPR

Prof.ª Drª Viviana Cocco Mariani
Pontifícia Universidade Católica do Paraná, PUCPR, orientadora

Curitiba, 18 de outubro de 2023

Rua: Imaculada Conceição, 1155 – Prado Velho – CEP: 80215-901 – Curitiba-Paraná-Brasil
Telefone: +55 41 3271-1385 **E-mail**: ppgem@pucpr.br

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

À Deus, primeiramente pelo dom da vida, pela saúde e por cuidar de mim todo dia em amor e sustento.

À minha esposa, Luiza de Oliveira Revers, pelo companheirismo, pela paciência e por todo suporte emocional dedicado ao longo da jornada da vida.

Aos meus pais, Marcos Tuleski e Elda Jane de Paula Tuleski, pela orientação familiar e por apoiar minhas decisões com carinho e dedicação.

Aos meus avôs, Aldo Paulo Tuleski e Joana Tuleski, e Alceu de Paula Souza e Ivone Maria de Paula Souza, por desbravarem o mundo e estabelecerem uma base familiar que me permitisse alcançar as minhas próprias conquistas.

À minha orientadora, Prof. Dra. Viviana Cocco Mariani, pelo acompanhamento, pelos ensinamentos, pela orientação e carinho. Ao corpo docente do programa de Pós-graduação em Engenharia Mecânica (PPGEM) da Pontifícia Universidade Católica do Paraná (PUCPR), pelos ensinamentos no decorrer do programa.

Aos colegas de trabalho na Bosch, principalmente pelo suporte prestado na coleta de dados em campo.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) pelo suporte financeiro desta pesquisa.

“A enxada respondeu de fato eu vivo no chão
Pra poder dar de comer e vestir o seu patrão
Eu vim no mundo primeiro quase no tempo de Adão
Se não fosse o meu sustento
Ninguém tinha instrução”.

(Zico e Zeca, 1996)

RESUMO

O aumento dos preços dos veículos zero-quilômetro tem impulsionado o mercado de reposição automotiva, o que torna necessário o desenvolvimento de métodos alternativos, rápidos e precisos para diagnosticar e reparar falhas nos componentes dos veículos. Neste contexto, essa dissertação propõe uma solução baseada em sinais de áudio gerados pelo funcionamento de motores e aplicação de técnicas de aprendizado de máquina para reconhecer e diagnosticar falhas em motores de ignição por compressão (MIC) de diferentes modelos de veículos. Os motores de ignição por compressão desempenham um papel fundamental em uma ampla gama de aplicações industriais e de transporte, tornando essencial a detecção precoce e precisa de falhas para garantir sua operação eficiente e segura. Neste estudo, foram coletados dados de áudio durante o funcionamento normal e em condições de falha controlada de motores de ignição por compressão. Uma análise dos sinais de áudio foi realizada para identificar características distintas que poderiam ser utilizadas para discriminar entre o funcionamento padrão e falhas específicas. Para isso, foi definida uma sistemática de coleta de dados e foram capturados áudios de três modelos de veículos com falhas simuladas. Utilizando recursos baseados em estatística *Wavelet Packet Transform* (WPT) foram extraídos dados de áudio utilizados em seis distintos modelos de ML aplicados em classificação. Assim, uma variedade de algoritmos de ML, tais como, *Random Forest* (RF), *Gradient Boosting*, *Support Vector Machine* (*SVM*), *Multilayer Perceptron* (*MLP)*, *Convolutional Neural Network* (*CNN*) que utilizou *Mel-frequency Cepstrum* (MFCC) como característica de entrada, e CNN combinada com rede neural *Long Short-Term Memory* (LSTM) foram aplicados para treinar modelos de classificação capazes de identificar automaticamente as falhas nos motores. Os resultados obtidos demonstraram a viabilidade do uso de sinais de áudio para o diagnóstico de falhas em motores de ignição por compressão. Os resultados experimentais mostraram que os modelos de ML alcançaram níveis significativos de precisão na detecção de falhas, com precisão média de classificação maior do que o modelo de comparação proposto MFCC-CNN, contribuindo para a compreensão dos principais indicadores acústicos associados a diferentes tipos de falhas. Este estudo representa um avanço importante, fornecendo uma abordagem não invasiva e eficaz para a manutenção preventiva e aprimorando a confiabilidade e a segurança desses sistemas críticos. Os resultados obtidos têm implicações significativas na indústria automotiva, de transporte e em outras áreas onde esses motores são amplamente utilizados.

ABSTRACT

The increase in prices of new vehicles has driven the automotive aftermarket business, making it necessary to develop an alternative, fast, and accurate methods for diagnosing and repairing faults in vehicle components. In this context, this study proposes a solution based on audio signals generated by the operation of engines and the application of Machine Learning (ML) techniques to recognize and diagnose faults in compression ignition engines (CIE) of different vehicle models. Compression ignition engines play a fundamental role in a wide range of industrial and transportation applications, making early and accurate fault detection essential to ensure their efficient and safe operation. In this study, audio data was collected during normal operation and under controlled fault conditions of compression ignition engines. A detailed analysis of the audio signal was performed to identify distinct faults. For this purpose, a data collection system was defined, and audio recordings of three vehicle models with simulated faults were captured. Using resources based on Wavelet Packet Transform (WPT) statistics, audio data used in six different ML models for classification were extracted. Thus, a variety of ML algorithms, including Random Forest (RF), Gradient Boosting, Support Vector Machine (SVM), Multilayer Perceptron (MLP), Convolutional Neural Network that used Mel-frequency Cepstrum (MFCC) as an input featrure, and CNN combined with Long Short-Term Memory (LSTM) neural network were applied to train classification models capable of automatically identifying engine faults. The results obtained demonstrated the feasibility of using audio signals for diagnosing faults in compression ignition engines. The experimental results showed that the ML models achieved significant levels of accuracy in fault detection, with an average classification accuracy greater than the proposed MFCC-CNN comparison model, contributing to the understanding of key acoustic indicators associated with different types of faults. This study represents an important advancement, providing a non-invasive and effective approach to preventive maintenance and enhancing the reliability and safety of these critical systems. The results obtained have significant implications in the automotive and transportation industry and in other areas where theses engines are widely used.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

SÍMBOLOGIA

Alfabeto Latino:

a	*Escala da Wavelet*
b	Posição da *Wavelet*
b	*Bias* do neurônio
b	Termo de deslocamento
$c(j,k)$	Coeficiente de *Wavelet Packet*
$DCT_i(n)$	Coeficiente do vetor
dt	Variável de integração
f	Frequência
$f(t)$	Sinal de entrada contínuo no domínio do tempo
$f(x)$	Previsão final
$f(z)$	Função de ativação não-linear
F_k	*Fast Fourier Transform*
f_n	Sinal discreto
$H_M(k)$	Resposta do Filtro *Mel*
$h_m(x)$	Previsão da árvore
i	Função complexa
k	Frequência do fim
m	Frequência de início
n,t	Tempo
N	Comprimento da janela
N	Número de amostras do sinal de entrada
N	Número total de pontos
r_i	Erro residual
T	Número de árvores do conjunto
w	Vetor normal
$W(a,b)$	Transformada *Wavelet*
$w(n)$	Janela de *Hamming*
w_i	Peso da conexão
x	Número de observações
$X(k)$	Espectro de magnitude

$x(n)$	Sinal de entrada
x_i	Valor de entrada para o neurônio
x_m	Amostra do sinal de entrada
$y(n)$	Sina pré-ênfase
y_i	Valor

Alfabeto Grego:

$\propto$	Coeficiente de pré-ênfase
μ	Média ou acurácia
σ	Desvio padrão
ψ	Função *Wavelet*

ABREVIATURAS

ADC	Analógico-Digital
AE	*Autoenconders*
ANN	*Artificial Neural Network*
ATDC	*Analogic Time Digital Conversion*
BP	*Back Propagation*
BPN	*Back Propagation Network*
CIE	*Compression Ignition Engine*
CM	*Condition Monitoring*
CMF	*Cumulative Mode Function*
CNN	*Convolutional Neural Network*
CWT	*Continuous Wavelet Transform*
DAC	Digital-Analógica
DBN	*Deep Boltzmann Machine*
DFT	*Discrete Fourier Transform*
DL	*Deep Learning*
DNN	*Deep Neural Network*
DWT	*Discrete Wavelet Transform*
ECG	Eletrocardiografia
ECU	*Electronic Control Unit*
EEG	Eletroencefalografia
EKF	*Extended Kalman Filter*
EMD	*Empirical Mode Decomposition*
ENN	*Extension Neural Network*
FBNN	*Functional Basis Neural Network*
FFT	*Fast Fourier Transform*
GA	*Genetic Algorithm*
GANN	*Generative Adversarial Neural Network*
GRNN	*General Regression Neural Network*
ICE	*Internal Combustion Engine*
IMF	*Intrinsic Mode Function*
KNN	*K-nearst Neighbours*
LM	*Levenberg-Marquardt*

LOLIMOT	*Local Linear Mode Tree*
LSTM	*Long Short-Time Memory*
MFCC	*Mel-Frequency Cepstrum*
ML	*Machine Learning*
MLP	*Multilayer Perceptron*
OBD	*On-board Diagnose*
PCA	*Principal Component Analysis*
QN	*Quase-Newton*
RBM	*Restricted Boltzmann Machines*
RF	*Random Forest*
RNN	*Recurrent Neural Network*
RSL	Revisão Sistemática da Literatura
SAE	*Sparse Auto-Enconder*
SSE	*Stacked Sparse Encoders*
STFT	*Short-Time Fourier Transform*
SVM	*Support Vector Machine*
SVSF	*Smooth Variable Structure Filter*
VC	Validação Cruzada
WPT	*Wavelet Packet Transform*

CAPÍTULO 1
INTRODUÇÃO

1.1 O MERCADO DE RESPOSIÇÃO AUTOMOTIVA

De acordo com Kusumawardhani e Parikesit (2019), o mercado de reposição automotiva é uma indústria global que envolve a fabricação e distribuição de peças de reposição para veículos automotivos. Essas peças são utilizadas para reparar, substituir ou melhorar componentes de automóveis após o período de garantia do fabricante ou em caso de falhas ou acidentes. O mercado de reposição automotiva é uma parte essencial da indústria automotiva, pois fornece aos proprietários de veículos a capacidade de manter seus carros em estado de funcionamento e prolongar sua vida útil.

O mercado de reposição automotiva é um setor global dinâmico e em constante crescimento. Com milhões de veículos nas estradas em todo o mundo, a demanda por peças de reposição e serviços de manutenção automotiva permanece crescente. Fabricantes de peças de reposição competem ferozmente para atender a essa demanda, oferecendo uma ampla gama de produtos, desde peças de motor até acessórios e componentes eletrônicos. Além disso, a crescente conscientização ambiental e regulamentações de emissões rigorosas estão impulsionando a demanda por peças de reposição mais eficientes e ecologicamente corretas, como componentes elétricos e híbridos.

Conforme ilustrado na Figura 1 o mercado de reposição abrange uma ampla gama de produtos, incluindo peças de motor, sistemas de exaustão, suspensão, freios, transmissões, eletrônicos, carrocerias, vidros, pneus e muitos outros componentes.

Figura 1 - Exemplo de peças encontradas no mercado de reposição automotivo.

Fonte: Feldman (2019).

A demanda por peças de reposição automotivas é impulsionada por vários fatores. O primeiro fator encontrado na literatura mencionado por Lima (2015) é o tamanho do parque automotivo em circulação, ou seja, o número total de veículos em uso. À medida que o número de veículos nas estradas aumenta, há uma necessidade correspondente de peças de reposição para manutenção e reparo. Além disso, a idade média dos veículos em circulação também desempenha um papel importante na demanda por peças de reposição. À medida que os carros envelhecem, os componentes começam a se desgastar e requerem substituição. Em alguns países, programas de incentivo financeiro para a compra de carros novos também podem influenciar a demanda por peças de reposição, pois levam à substituição de veículos antigos.

A indústria de peças automotivas é altamente competitivo e envolve uma gama de participantes, incluindo fabricantes de peças de reposição, distribuidores, revendedores, oficinas de reparo automotivo e varejistas. Os fabricantes de peças de reposição geralmente produzem peças equivalentes ou compatíveis com as peças originais dos fabricantes de veículos. Eles precisam garantir a qualidade de seus produtos e manter uma rede de distribuição para atender às demandas dos clientes.

A tecnologia desempenha um papel cada vez mais importante no mercado de reposição automotiva. Os avanços na eletrônica embarcada, como sistemas de diagnóstico de veículos e sensores de desgaste, estão impactando a maneira como as peças de reposição são projetadas e fabricadas. Além disso, o comércio eletrônico e as vendas *online* estão ganhando destaque, permitindo que os consumidores encontrem e comprem peças de reposição com mais facilidade. Embora o mercado de reposição automotiva seja global, as condições podem variar de país para país devido a fatores como regulamentações governamentais, preferências dos consumidores e características do parque automotivo. Lima (2015), trouxe o conceito de que em alguns países, a indústria de reposição automotiva é desenvolvida e estruturada, enquanto em outros pode enfrentar desafios, como a presença de peças falsificadas ou falta de padrões de qualidade. No geral, o mercado de reposição automotiva desempenha um papel crucial na manutenção e sustentação da indústria automotiva.

O setor de reposição para veículos também está passando por uma transformação digital significativa. O comércio eletrônico desempenha um papel cada vez mais importante, permitindo que os consumidores comprem peças de reposição online com facilidade e comodidade. Além disso, a tecnologia está sendo incorporada às próprias peças de reposição, com sensores e sistemas de conectividade que permitem a manutenção preditiva e aprimorada. À medida que os veículos se tornam mais complexos, o mercado de reposição continuará a evoluir para atender às crescentes demandas dos consumidores por conveniência, qualidade e eficiência.

O mercado de peças de reposição para automóveis no Brasil é uma indústria robusta e em constante crescimento. Com uma frota de veículos que continua a aumentar, a demanda por peças de reposição é alta, tornando o setor um dos mais importantes na economia automotiva do país. Fabricantes de peças e acessórios automotivos nacionais e internacionais competem para atender a essa crescente demanda, oferecendo uma ampla gama de produtos que variam de peças de desgaste, tais como pastilhas de freio e filtros de óleo, a acessórios personalizados que permitem aos proprietários de veículos customizarem seus carros. Além disso, a crescente conscientização sobre a manutenção preventiva e a importância da segurança veicular impulsionaram ainda mais o mercado de reposição, à medida que os motoristas buscam manter seus veículos em condições ideais de funcionamento.

No entanto, a indústria de componentes automotivos no Brasil também enfrenta desafios significativos. A qualidade e a autenticidade das peças são questões cruciais, com a proliferação de produtos falsificados e de baixa qualidade representando um risco para a segurança dos motoristas e a durabilidade dos veículos. Além disso, a complexa carga tributária e as regulamentações governamentais adicionam um fator complicador ao setor, afetando os preços das peças e a capacidade das empresas de competir. Para se destacar nesse mercado altamente competitivo, as empresas de reposição automotiva precisam focar na qualidade, na inovação e na garantia de produtos autênticos para construir e manter a confiança dos consumidores, enquanto enfrentam os desafios operacionais e regulatórios em constante evolução.

De acordo com Tourme *et al.* (2022), o mercado de acessórios automotivos no Brasil é um setor bastante promissor que movimenta cerca de R$100 bilhões por ano, e a previsão é que ele cresça entre 6% e 7% até o final de 2023. Esse aumento no mercado acompanha diretamente a indústria de manutenção mecânica, que é essencial para garantir o bom funcionamento dos veículos. A manutenção mecânica é realizada em parte por oficinas, que têm a responsabilidade de identificar corretamente os componentes dos veículos que precisam de reparo para garantir a segurança dos motoristas e passageiros.

1.2 DIAGNÓSTICO DE FALHAS EM VEÍCULOS

O diagnóstico de falhas em veículos desempenha um papel fundamental na manutenção e no funcionamento seguro de automóveis modernos. Com o avanço da tecnologia automotiva, os veículos estão cada vez mais equipados com sistemas eletrônicos complexos, o que torna o diagnóstico de falhas uma tarefa desafiadora e crucial. Utilizando ferramentas de diagnóstico avançadas, tais como scanners de diagnóstico e software especializado, os mecânicos e técnicos automotivos podem

identificar problemas mecânicos, elétricos e eletrônicos em um veículo de maneira eficiente. Isso não apenas ajuda a evitar que os problemas se agravem, mas também contribui para a eficiência do combustível e a redução das emissões, além de garantir a segurança dos ocupantes e a confiabilidade do veículo.

O processo de diagnóstico de falhas em veículos envolve a leitura de códigos de erro gerados pelos computadores de bordo do veículo, a realização de testes em sistemas específicos, como motor, transmissão, freios e suspensão, e a inspeção visual de componentes importantes. Além disso, a interpretação precisa dos dados coletados é fundamental para determinar a causa geradora dos problemas e realizar os reparos adequados. Com a evolução da inteligência artificial (IA) e da conectividade veicular, também surgem diagnósticos remotos, onde os fabricantes e concessionárias podem monitorar os veículos em tempo real e até mesmo realizar atualizações de software para corrigir problemas sem a necessidade de uma visita física às oficinas. Em resumo, o diagnóstico de falhas em veículos é uma prática essencial para manter a segurança, o desempenho e a confiabilidade dos automóveis modernos, e continuará a evoluir à medida que a tecnologia automotiva avança.

No entanto, o diagnóstico de falhas em veículos é uma parte crítica da manutenção automotiva e da solução de problemas em veículos automotivos. Existem várias alternativas e técnicas utilizadas para identificar e diagnosticar falhas em veículos, tais como por exemplo leitura de códigos de erro manualmente, teste de condução, medição de parâmetros, inspeção visual, teste de componentes individuais, análise de dados e *software* de diagnóstico, teste de emissões e inspeção veicular, análise de ruídos e vibrações e *scanner On-Board Diagnostics* (OBD).

Para diagnosticar quais componentes precisam ser reparados, os mecânicos utilizam equipamentos eletrônicos conhecidos como *scanners*, conforme ilustrado na Figura 2. Esses aparelhos são responsáveis por receber e analisar, em tempo real, todos os parâmetros operacionais do sistema de injeção eletrônica do veículo, com o objetivo de detectar eventuais falhas no sistema. No entanto, o custo desses equipamentos – utilizados em âmbito profissional – é extremamente alto, o que acaba por exigir um alto investimento por parte das oficinas. Além disso, é necessário ter um *hardware* específico e conhecimento técnicos qualificados dos mecânicos que os operam para garantir a correta interpretação dos resultados e evitar falhas na manutenção.

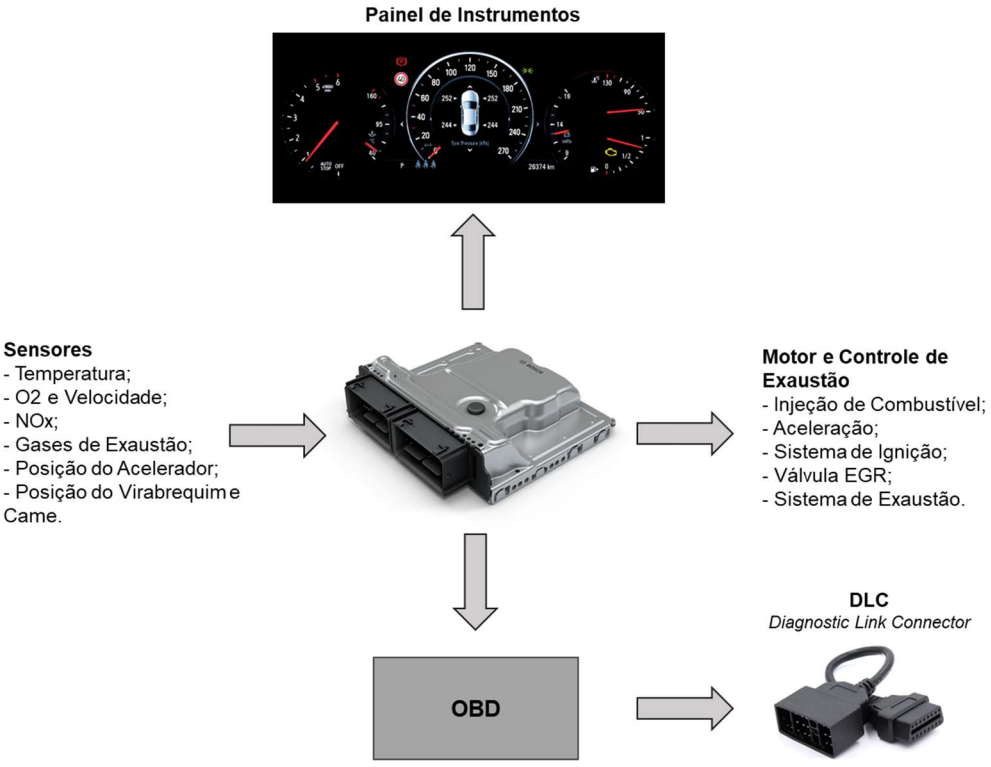
Existem vários tipos de OBD, sendo os mais comuns o OBD-I e o OBD-II. O OBD-I foi o primeiro a ser implementado e, embora varie em termos de padrões de comunicação e protocolos entre fabricantes, foi um marco importante no diagnóstico veicular. Posteriormente, o OBD-II, que é mais padronizado e amplamente utilizado, tornou-se o padrão predominante como destacado por Smith (2015). Na prática, o OBD é composto por uma rede complexa de sensores e dispositivos eletrônicos integrados no veículo, que coletam dados operacionais em tempo real, como a temperatura

do motor, pressão dos pneus, nível de combustível, velocidade do veículo e emissão de poluentes. Com base na Figura 3, esses dados são processados por unidades de controle eletrônico (ECUs) distribuídas em todo o veículo, que utilizam algoritmos avançados para monitorar o desempenho do motor e dos sistemas de emissões.

Figura 2 - Exemplo de um *scanner* automotivo de diagnóstico Bosch.

Fonte: Santana (2020).

Figura 3 - Esquema ilustrativo de um sistema de diagnóstico a bordo (OBD) em um veículo.



Fonte: Tuleski (2023).

A comunicação entre os componentes do OBD é realizada por meio de barramentos de dados internos, como o *Controller Area Network* (CAN), que permite a troca de informações de forma eficiente entre os diversos sensores e ECUs. A capacidade de processamento de dados e a velocidade de comunicação desses sistemas são cruciais para a detecção imediata de falhas ou anomalias, garantindo a segurança do veículo e a conformidade com regulamentações de emissões Gupta (2019). Além disso, o OBD permite a comunicação externa com dispositivos de diagnóstico, como *scanners* OBD-II, que utilizam protocolos de comunicação padronizados para acessar informações do veículo, Smith (2015).

Como apontado por Jones *et al.* (2018), o OBD desempenha um papel fundamental na redução das emissões de poluentes atmosféricos, ajudando a cumprir regulamentações rigorosas de controle de emissões em todo o mundo. Por outro lado, no estudo realizado por Gupta (2019) ressaltou a importância da interoperabilidade e padronização dos sistemas OBD para garantir a acessibilidade e a utilidade desses sistemas em todos os veículos, independentemente de sua marca ou modelo.

Portanto, é essencial que as oficinas estejam atentas às novas tecnologias e invistam em equipamentos modernos, bem como em treinamentos para os mecânicos, a fim de garantir um diagnóstico preciso e seguro para os clientes. Com o avanço da tecnologia, novos equipamentos e ferramentas estão surgindo para facilitar o trabalho dos mecânicos, o que pode resultar em diagnósticos mais acurados e, consequentemente, em uma eficiência na manutenção dos veículos.

1.3 MOTORES DE IGNIÇÃO POR COMPRESSÃO (MIC)

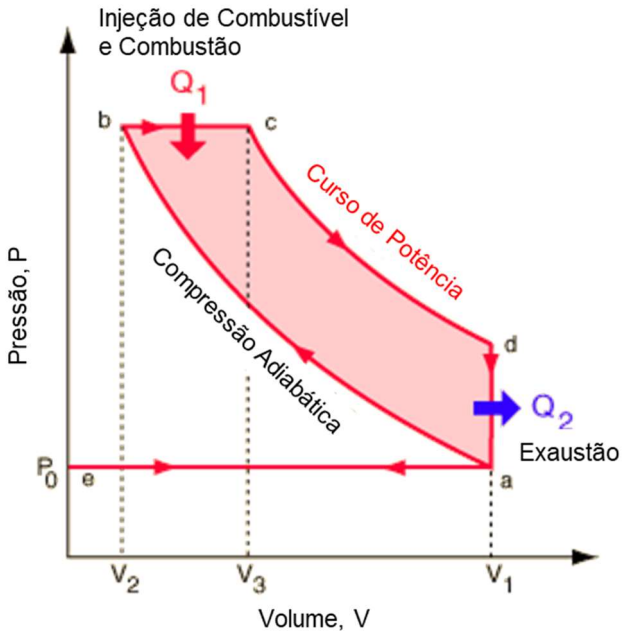
Sujesh e Ramesh (2018) abordaram em seus estudos, que os motores de ignição por compressão, popularmente conhecidos como motores Diesel, têm uma história que remonta ao final do século XIX. Em 1892, Rudolf Diesel, um engenheiro alemão, desenvolveu o primeiro protótipo funcional de um motor de ignição por compressão, patenteado em 1893. O motor diesel operava com base no princípio de compressão adiabática, no qual o ar altamente comprimido é usado para gerar a combustão do combustível diesel injetado no cilindro, conforme ciclo ilustrado na Figura 4.

Os motores de ignição por compressão apresentam características distintas que os diferenciam dos motores de ignição por centelha. Eles representam uma classe fundamental de motores de combustão interna. A principal característica desses motores é a forma como a ignição do combustível ocorre. Ao contrário dos motores a gasolina, nos quais a ignição é provocada por uma faísca gerada por uma vela de ignição, nos motores de ignição por compressão, a combustão é iniciada pela alta pressão gerada pela compressão do ar no cilindro. Esse aumento de pressão aquece o ar a ponto de inflamar o combustível diesel quando é injetado no cilindro, sem a necessidade de faísca. Isso torna

esses motores mais eficientes em termos de combustível e adequados para uma variedade de aplicações, desde veículos pesados, como caminhões e ônibus, até equipamentos industriais e marítimos.

Os motores de ignição por compressão são apreciados por sua durabilidade e eficiência térmica. Eles tendem a ser mais robustos do que os motores a gasolina em termos de durabilidade e podem suportar cargas pesadas e condições adversas, tornando-os ideais para uso em veículos comerciais e de carga. Além disso, a eficiência térmica superior desses motores significa que eles consomem menos combustível em comparação com os motores a gasolina, o que resulta em economia de combustível e menor emissão de gases poluentes. No entanto, a tecnologia de ignição por compressão também apresenta desafios, como a emissão de óxidos de nitrogênio (NO_x) (Nogueira *et al*., 2023) e partículas finas, que exigem sistemas de controle de emissões mais avançados para cumprir as regulamentações ambientais rigorosas.

Figura 4 - Ciclo de um motor de ignição por compressão.



Fonte: Nave (2018).

Conforme Ambarish e Bijan (2016), os MIC têm uma eficiência térmica superior à dos motores de ignição por centelha, devido à maior relação de compressão e ao processo de combustão mais eficiente. Isso os torna ideais para aplicações que exigem economia de combustível e alta eficiência energética.

John (2009) em seus estudos ressaltou as aplicações dos motores de ignição por compressão, dentre elas estão transporte de carga, maquinaria pesada, grupos de geradores e até mesmo aplicações marítimas. Por fim, Thangaraja *et al.* (2022) não deixaram de mencionar os grandes desafios dos MIC relacionado às emissões de poluentes, especialmente em veículos de estrado. Portanto, avanços significativos têm sido feitos na tecnologia de controle de emissões para atender às regulamentações ambientais.

1.4 JUSTIFICATIVA

Uma das grandes dificuldades na manutenção de veículos MIC é a identificação precisa de falhas nos seus sistemas. Como mencionado anteriormente, os *scanners* são recursos utilizados para diagnosticar essas falhas, no entanto, podem ser equipamentos extremamente caros e exigem conhecimentos técnicos qualificados dos mecânicos que os operam. Além disso, a interpretação dos dados obtidos pode ser subjetiva e imprecisa por parte da pessoa que avalia.

Com o intuito de solucionar a problemática mencionada, foi primeiramente inspirado o estudo de Laguarta *et al*. (2020), que abordou a aplicação de técnicas de inteligência artificial para o diagnóstico de pacientes com COVID-19 a partir da análise da gravação da tosse. A partir dessa premissa, surgiu a ideia de avaliar métodos semelhantes para aplicação no contexto automotivo. Assim sendo, tem-se o objetivo de promover a utilização da inteligência artificial para o diagnóstico de falhas em veículos MIC, por meio da análise da assinatura de áudio emitido por seus componentes.

1.5 OBJETIVOS

Este estudo tem como principal objetivo a análise do som emitido por veículos de motor de ignição por compressão, como caminhonetes, ônibus e caminhões, propondo uma solução eficiente e relativamente simples para diagnosticar e classificar falhas em tais veículos através da aplicação e avaliação de diferentes modelos de ML.

Além do objetivo principal, este estudo contempla objetivos secundários, igualmente relevantes, que são citados conforme segue:

- Definir uma metodologia adequada para coleta de dados dos veículos visando à obtenção de informações precisas e confiáveis.
- Classificar as falhas obtidas, bem como a análise adequada e o tratamento dos áudios coletados de forma experimental.

- Definir os modelos de ML e seus hiperparâmetros mais adequados para a classificação de falhas através dos áudios obtidos nos diferentes veículos.
- Desenvolver um modelo de Inteligência Artificial aplicado na classificação das falhas de veículos de forma adequada.
- Melhorar o desempenho do modelo de Inteligência Artificial proposto, aumentando a precisão, a eficiência, e a capacidade de generalização.
- Otimizar os hiperparâmetros dos modelos de ML para obter um desempenho de acurária e *F1-Score* superior para a tarefa específica deste estudo.
- Reduzir o viés e promover a equidade nas decisões tomadas pelo modelo de Inteligência Artificial proposto neste estudo.
- Comparar os resultados obtidos com outros modelos ou abordagens existentes na literatura de aprendizado de máquina.

É importante ressaltar que a aplicação da inteligência artificial na detecção de falhas em veículos MIC trará benefícios não apenas para a indústria automobilística, mas também para a segurança e qualidade de vida dos usuários desses veículos, além de contribuir para a redução de custos e emissão de poluentes. Vale destacar, que o intuito deste estudo é verificar qual modelo de ML fornece a melhor acurácia na detecção de falhas através do som em veículos comerciais leves, médios e pesados. conforme ilustrado na Figura 5. A frota de veículos apresentada na Figura 5 representou cerca de 15% dos veículos registrados no Brasil em 2020, o que demonstra a importância, relevância e necessidade de investimentos em novas tecnologias para diagnóstico e detecção de falhas em tais veículos automotivos.

Figura 5 - Exemplo de frota veicular comercial leve, média e pesada.

Fonte: Tuleski (2023).

1.6 ORGANIZAÇÃO DO DOCUMENTO

Este estudo segue a estrutura de um documento científico, composto por vários capítulos conforme descritos a seguir. Inicialmente no Capítulo 1 tem-se a Introdução onde o tema é apresentado, a problemática do estudo e seus objetivos. No Capítulo 2 tem-se a revisão bibliográfica, que por sua vez, justifica a importância da pesquisa ao apresentar trabalhos já publicados sobre o tema. No Capítulo 3 tem-se a fundamentação teórica, onde são apresentados os conceitos e teorias fundamentais que embasam a pesquisa a respeito de sinais. Para a exposição dos modelos de classificação utilizados, baseados em ML, é proposto o Capítulo 4. No Capítulo 5 está a descrição dos materiais utilizados. Os resultados obtidos são detalhados no Capítulo 6 e a Conclusão é apresentada no Capítulo 7 bem como sugestões para trabalhos futuros. Por fim, as referências bibliográficas são descritas na última seção do documento permitindo a consulta dos trabalhos citados e usados neste estudo.

CAPÍTULO 2
REVISÃO SISTEMÁTICA DA LITERATURA

A revisão sistemática da literatura (RSL) desempenha um papel fundamental neste projeto, pois oferece uma base sólida para a pesquisa, permitindo compreender o estado atual do conhecimento na área, sendo possível identificar lacunas no campo, descobrir abordagens e técnicas já testadas, bem como as que ainda não foram exploradas.

Isso ajuda a evitar a redundância na pesquisa e a direcionar os esforços para áreas onde a inovação é mais necessária. Além disso, a RSL auxilia na identificação de melhores práticas, na seleção apropriada de algoritmos de aprendizado de máquina e na avaliação crítica das metodologias empregadas em estudos anteriores, contribuindo para a construção de um arcabouço sólido e confiável para o diagnóstico de falhas em motores de ignição por compressão por meio de sinais de áudio. A RSL se dará no período de 2009 a 2023.

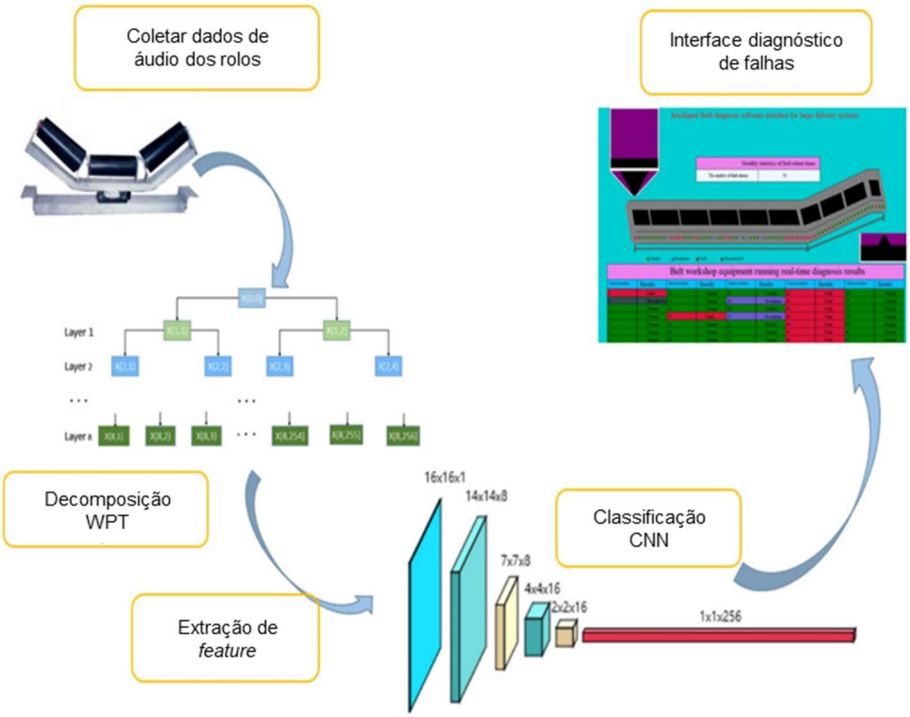
De acordo com Henriquez *et al.* (2013), o monitoramento de condições (do inglês, *Condition Monitoring*, CM) exerce um papel fundamental no diagnóstico de falhas em componentes mecânicos, pois influencia diretamente na continuidade operacional de um determinado processo. Uma das principais abordagens do CM é a utilização de dados medidos, como vibração e sinais acústicos, em conjunto com técnicas de aprendizado de máquina para diagnosticar anomalias em máquinas em funcionamento.

Zhao *et al.* (2019) apresentaram uma revisão de estudos em que foram utilizados modelos de *Deep Learning* (DL) com diferentes métodos de processamento para identificar falhas em máquinas. Esses estudos indicaram um excelente potencial no uso dessa abordagem para monitorar a vitalidade de máquinas e motores. Em uma recente e extensiva revisão, Shooter *et al.* (2021) enfatizaram a importância da extração de características e processamento adequado para otimizar a performance dos classificadores utilizados no CM aplicado a motores. Foram analisadas diferentes arquiteturas de DL, como a rede neural MLP, *Autoencoders* (AE), *Deep Boltzmann Machine* (DBN), *Recurrent Neural Networks* (RNN) e *Generative Adversial Neural Networks* (GAN), com entrada de características como vibração, acústica e sinais de som, tempo, frequência e tempo-frequência, utilizando transformadas conhecidas como *Fast Fourier Transform* (FFT), *Short-Time Fourier Transform* (STFT) e *Wavelets*. Concluiu-se que a escolha correta das características para a metodologia escolhida de ML tem impacto na generalização do classificador, além de levar em consideração o volume de dados disponíveis.

Embora as técnicas de ML aplicadas ao diagnóstico de falhas em máquinas, motores e sistemas similares sejam um tópico recente, existem pesquisas relevantes publicadas na literatura. Em

um estudo recente, Sun *et al.* (2016) utilizaram técnicas de DL não supervisionadas baseadas no *Sparse Auto-Encoder* (SAE) para classificar falhas em um motor de indução, utilizando sinais de vibração de ruídos capturados por um acelerômetro. O estudo atingiu boa acurácia de predição ao utilizar o algoritmo de *Denoising Auto-Enconder* e *Dropout Layer* para tornar a predição mais robusta. Um estudo realizado por Yang *et al.* (2020) compararam classificadores de *Support Vector Machine* (SVM), *Deep Neural Network* (DNN) e CNN utilizando características baseadas em estatísticas e MFCC extraídos do segmento de sinais de áudio para classificar a condição dos rolos em uma esteira transportadora, como pode ser observado na Figura 6.

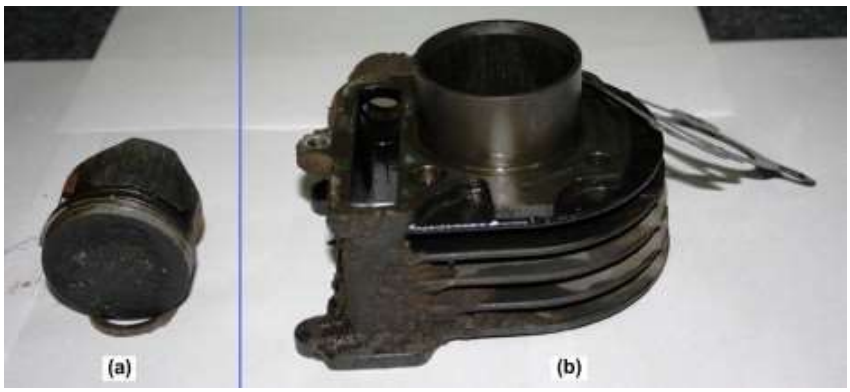
Figura 6 - Estudo sobre monitoramento do estado dos rolos de uma esteira transportadora.



Fonte: Yang *et al.* (2019).

Jena e Panigrani (2014) empregaram a extração de características baseadas em estatística, em sinais acústicos, e *Continuous Wavelet Transform* (CWT) para comparar o desempenho dos modelos MLP e SVM na classificação de falhas de folga no pistão de um motor de motocicleta, conforme exemplificado na Figura 7.

Figura 7 - (a) Pistão e (b) Cilindro utilizados no estudo para a classificação de falhas de folga.



Fonte: Jena e Panigrani (2014).

Adicionalmente, tem havido um aumento na literatura relacionada a CM em motores a combustão interna utilizando modelos de ML para analisar falhas em diferentes componentes. Jafari *et al.* (2014), por exemplo, classificaram várias condições de válvulas de escapamento, conforme ilustrado na Figura 8, utilizando recursos estatísticos extraídos de técnicas de emissão acústica em conjunto com a rede neural MLP. Aquele estudo concluiu que a utilização de recursos baseados em estatística alcançou melhor desempenho do que a análise temporal e de frequência.

Figura 8 - Exemplo de falha mecânica na válvula de escapamento.

Fonte: Jafari *et al.* (2014).

Um estudo comparativo realizado por Ahmed *et al.* (2014) avaliaram diferentes estratégias para treinar uma MLP de forma eficiente, com o objetivo de detectar e classificar falhas pré-selecionadas de um motor de combustão interna como folga no pistão e no tensor de corrente. Os métodos *Back Propagation* (BP), *Levenberg-Marquardt*, *Quase-Netwon*, *Extended Kalman Filter* (EKF) e *Smooth Variable Structure Filter* (SVSF) foram comparados, sendo que o último apresentou melhor desempenho. Hou *et al.* (2019) propuseram um modelo simples para diagnóstico de falhas no cilindro de um motor Diesel marítimo, utilizando variáveis primárias como temperatura e pressão

para treinar uma rede neural MLP por meio dos métodos BP e *Levenberg-Marquadt*. O modelo proposto obteve uma taxa de acerto acima de 95% na classificação de falhas de maneira correta.

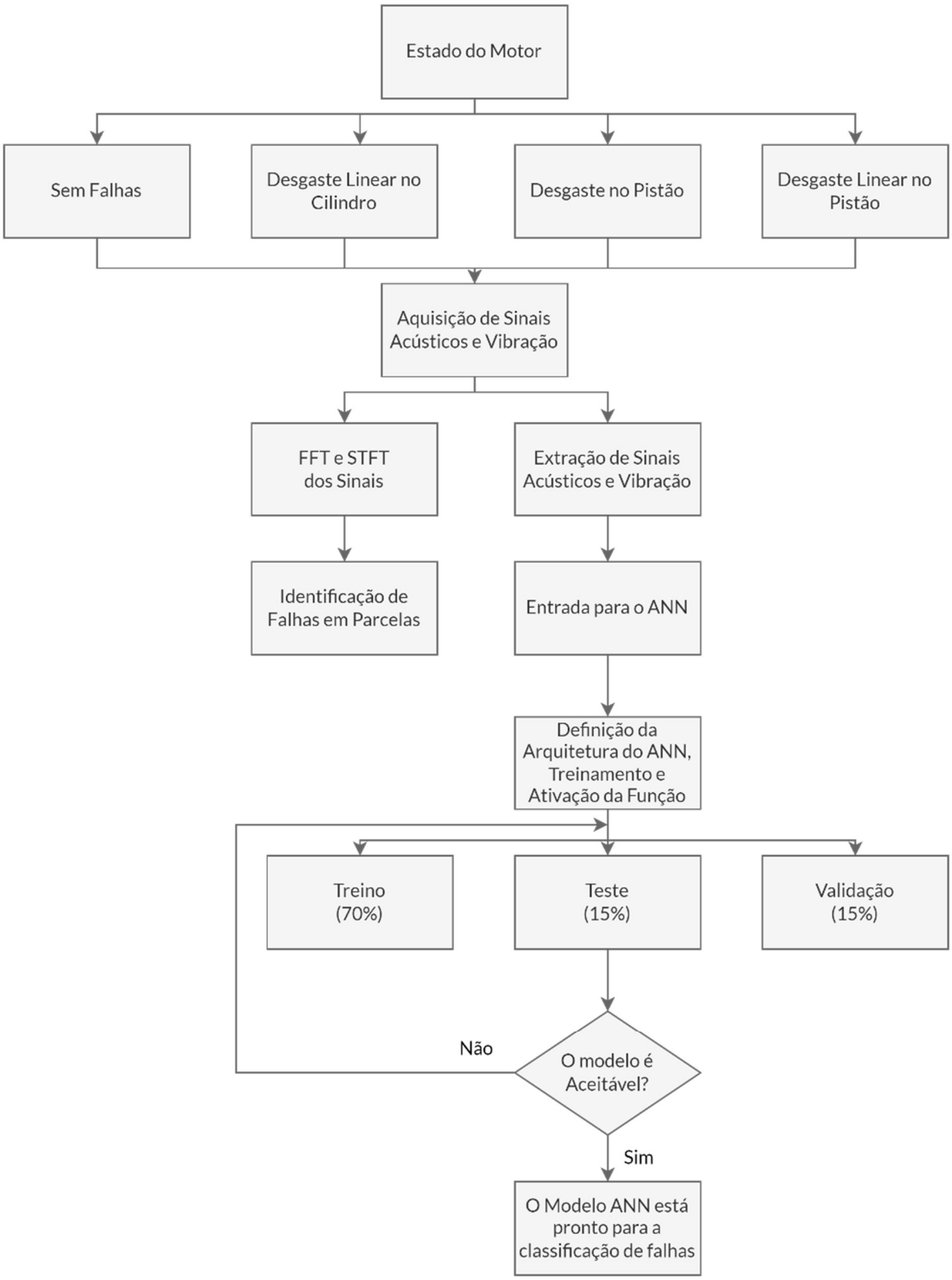
Ramteke *et al.* (2019) utilizaram a FFT em sinais de vibração e acústicos para extrair características estatísticas e detectar falhas de desgaste no revestimento do cilindro de um motor de ignição por compressão. Em um estudo mais recente, Ramteke *et al.* (2021) aprofundaram suas análises com a aplicação da STFT em conjunto com uma MLP para classificar falhas de desgaste no revestimento do cilindro, pistão ou em ambos, combinados, em motores de combustão, como pode ser observado na Figura 9.

Shiblee *et al.* (2017) propuseram uma abordagem diferenciada para o diagnóstico de falhas em cilindros de motores, que consiste na utilização do modelo empírico de decomposição para encontrar *Intrinsic Mode Function* (IMFs) de sinais de vibração medidos em quatro diferentes sensores posicionados estrategicamente. As IMFs são usadas para obter recursos estatísticos baseados em uma função cumulativa de modo, que são empregados para treinar a MLP no diagnóstico de falhas nos cilindros. Além disso, os pesquisadores destacaram o crescente número de estudos que têm aplicado a transformada *Wavelet* na extração de características nos últimos anos. A abordagem de utilizar a Transformada *Wavelet* para extrair características de um sinal de vibração ou áudio em conjunto com modelos de ML tem se mostrado uma estratégia promissora no campo de CM de *Internal Combustion Engines* (ICEs), obtendo resultados satisfatórios para diferentes tipos de falhas.

Ravikumar *et al.* (2020) compararam diferentes métodos de extração de características de sinais de vibração utilizando modelos de ML para identificar falhas em uma caixa de marcha, incluindo a abordagem CWT. Ao final do estudo, os pesquisadores concluíram que a extração de características estatísticas do CWT em conjunto com o algoritmo de *Decision Tree* apresentou um potencial adequado para o CM de motores.

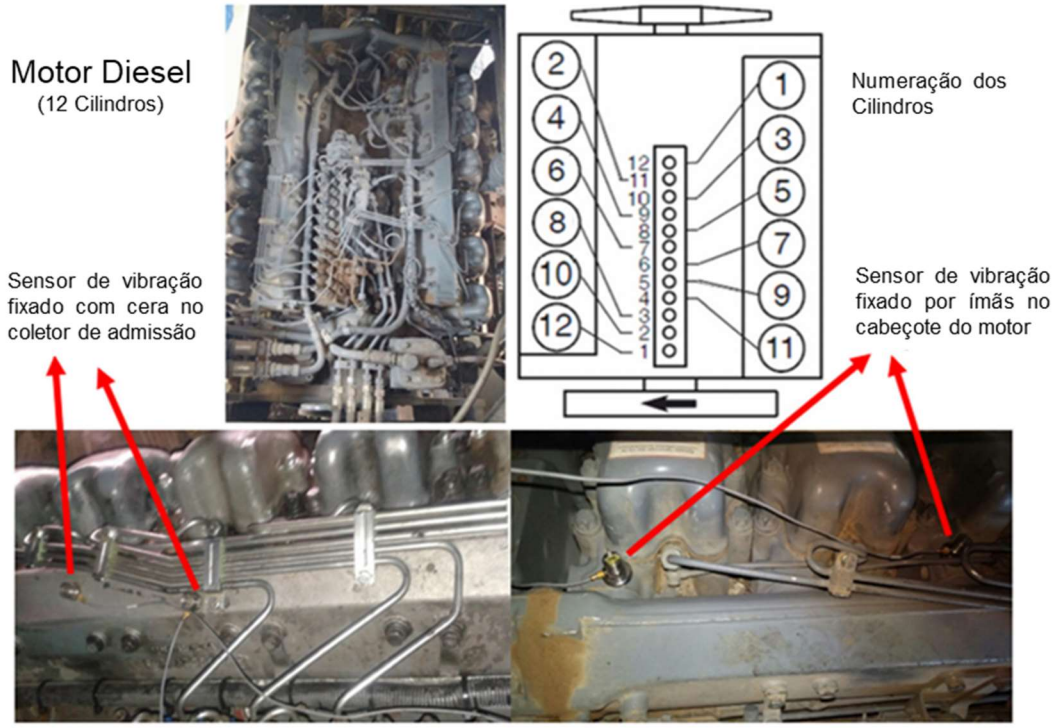
Ayati *et al.* (2020) investigaram o uso da FFT e do WPT em sinais de vibração para extrair recursos não correlacionados, que foram utilizados para treinar e comparar o desempenho dos classificadores SVM, MLP e *k-nearest neighbors* (kNN) na detecção de falhas de injeção como pode ser observado na Figura 10. Os autores concluíram que as *Wavelet* da família *Daubechies* proporcionaram recursos com alto potencial de classificação.

Figura 9 - Fluxo do delineamento experimental para a investigação de falhas em motores a combustão.



Fonte: Ramteke *et al.* (2021).

Figura 10 - Motor utilizado no experimento, instrumentado, e cilindros numerados.



Fonte: Ayati *et al.* (2020).

Shatnawi e Alkhassaweneh (2014) propuseram uma abordagem para a detecção de falhas em motores, utilizando gravações estéreo e dividindo os canais esquerdo e direito para extrair características estatísticas com o uso de pacotes de transformada *wavelet*. Posteriormente, compararam o desempenho de diferentes arquiteturas MLPs na classificação de falhas. Em um estudo semelhante, Wu e Liu (2009) utilizaram a entropia *Shannon* de cada nó de sinal de áudio do pacote de Transformada *Wavelet* como característica de entrada para treinar uma MLP para diagnosticar diversos tipos de falhas. Essa análise foi testada em condições de marcha lenta, 2000RPM e arranque. Além disso, foi comparado db4, db8 com db20 de *Daubechies* e técnica de regressão geral BP para avaliar qual abordagem produziu os melhores resultados.

No estudo, investigando a cavitação de uma bomba com base na assinatura sonora de áudio usando rede neural artificial, Anis *et al.* (2019), selecionaram nove atributos no domínio da frequência como entrada do classificador de rede neural. Este classificador foi treinado com o algoritmo de *Resilient Backpropagation*. Por fim, o desempenho do sistema em determinar a existência de cavitação apresentou uma precisão de 82,5%.

Borré *et al.* (2023) desenvolveram uma solução baseada em uma rede neural convolucional híbrida com arquitetura de memória de longo e curto prazo (CNN-LSTM) para treinar um conjunto de dados coletados através de um sensor acoplado a um motor elétrico com o principal objetivo de

prever falhas. Os autores concluíram que a aplicação desse modelo híbrido baseado em atenção CNN-LSTM, combinado com o uso de regressão quantílica para capturar incerteza, produziu resultados superiores em comparação aos modelos de referências tradicionais.

Na **Erro! Fonte de referência não encontrada.** são descritos os estudos que foram citados na revisão sistemática da literatura (RSL) listados em ordem crescente pelo ano de publicação, com os principais resultados, o modelo de classificação utilizado bem como o problema ou objetivo dele.

Em resumo, os trabalhos citados durante revisão sistemática da literatura buscam trazer o estado da arte no que toca ao tema aprendizado de máquina aplicado ao diagnóstico de falhas em motores de ignição por compresssão usando sinais de áudio. Através dessa analise, conclui-se que a literatura ainda não explora esse principal tema, o diagnóstico de falhas, como entrada sendo considerada o sinal de áudio.

Tabela 1 - Referências citadas ao longo da revisão bibliográfica em ordem cronológica.

Autores	**Ano**	**Problema Objetivo**	**Modelo**	**Resultados**
Wu e Liu.	2009	Diagnóstico de falhas em ICEs	WPT, ANN, DWT, GRNN e BPN	Precisão de classificação de 95%
Jafari *et al*	2014	Detecção de danos em válvulas de ICEs	AE e ANN	92% de desempenho na distinção de defeitos
Jena e Panigrani.	2014	Identificação de falhas em motor por meio do som	CWT, SVM e FBNN	FBNN apresentou a maior eficácia dentre as técnicas propostas
Shatnawi e Alkhassaweneh.	2014	Diagnóstico de falhas em motores de combustão interna (ICEs)	ENN	Eficácia e alta taxa de reconhecimento na classificação de falhas
Henriquez *et al.*	2014	RSL sobre diagnóstico de falhas baseado em vibração e áudio	CM	Mais de 100 referências disponíveis para serem utilizadas
Sun *et al.*	2016	Diagnóstico de falhas em motores de indução	SAE e DNN	O método proposto alcançou um desempenho superior que uma rede neural tradicional
Shiblee *et al.*	2017	Diagnóstico de falhas em ICEs	EMD, IMF, CMF e ANN	96% de precisão alcançada na classificação de falhas
Zhao *et al.*	2019	RSL de DL no monitoramento de máquinas	AE, RBM, DBN, DBM, CNN e RNN	Comparação entre o desempenho das abordagens e tendência de métodos
Ramteke *et al.*	2019	Processamento de sinais de som e vibração aplicados ao diagnóstico de falhas em ICEs	FFT	Os métodos de análise de sinal de vibração podem ser utilizados na detecção de falhas de escoriação
Anis *et al.*	2019	Investigação de cavitação em bombas com base na assinatura de áudio	ANN	82,5% de acurácia média na classificação de bombas com cavitação
Hou *et al.*	2019	Previsão de falhas em um motor Diesel marítimo	GA e MLP	A taxa média de diagnóstico correto foi maior que 95%

Ayati *et al.*	2020	Detecção de falhas na injeção baseada em vibração de um motor Diesel	FFT, WPT, MLP, SVM, KNN e LOLIMOT	O desempenho médio do método proposto foi de 99,72%
Ravikumar *et al.*	2020	Diagnóstico de falhas na caixa de engrenagem em um ICE	CWT e STFT	As técnicas utilizadas apresentam alto desempenho para a aplicação utilizada
Yang *et al.*	2020	Monitoramento dos rolos em uma correia	SSE e CNN	O sistema de diagnóstico de falhas funciona com 96,7% de precisão
Shooter *et al.*	2021	RSL sobre o processamento de atributos na implementação de DL aplicado ao CM	MLP, AE, RBM, DBM, DBN, CNN, RNN e GAN	Identificar lacunas de pesquisa na aplicação dos métodos citados
Ahmed *et al.*	2021	Detecção de falhas do motor por meio de vibração no domínio ângulo-manivela	ANN, BP, LM, QN, EKF e SVSF	A técnica apresentada é altamente indicada para aplicação de diagnóstico em campo
Ramteke *et al.*	2021	Detecção de falhas de escoriação em componentes de um motor Diesel	FFT, STFT e ANN	As falhas foram diagnosticadas com uma precisão de 100%
Borré *et al.*	2023	Previsão de falhas em máquinas elétricas	CNN-LSTM	Resultados superiores em comparação com os modelos de referência tradicional

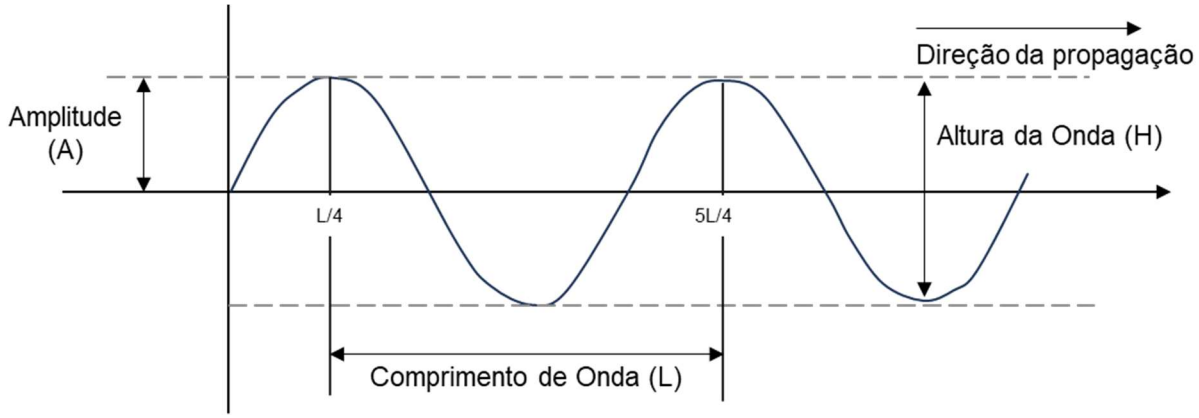
Fonte: Tuleski (2023).

CAPÍTULO 3
FUNDAMENTAÇÃO TEÓRICA

3.1 ASSINATURA DE ÁUDIO

A assinatura de áudio (Umapathy *et al,* 2010) é um campo que envolve a análise dos padrões sonoros distintos produzidos por fontes de som, sejam elas humanas, máquinas, instrumentos musicais ou ambientes naturais. A assinatura de áudio, também conhecida como uma impressão digital é a única para cada elemento. A sua singularidade inclui atributos que estão demonstrados na Figura 11 e podem variar como frequência, amplitude, tempo, padrões de ruído e outros elementos que compõem o som.

Figura 11 - Parâmetros que definem uma assinatura de áudio.



Fonte: Tuleski, 2023.

Com base em Wu *et al.* (1998), um motor, uma das fontes sonoras mais comuns em nosso dia a dia, gera uma assinatura de áudio distintiva devido à sua mecânica interna e às condições em que opera. Quando o motor está em funcionamento, diversas partes móveis, como pistões, válvulas e rolamentos, produzem ruídos característicos. Além disso, a carga e velocidade de operação do motor também influenciam a assinatura de áudio. Problemas de manutenção, como desgaste em componentes, desalinhamento ou falta de lubrificação, podem introduzir variações na assinatura de áudio que indicam problemas.

Umapathy *et al.* (2010) definiram que uma assinatura de áudio é composta por várias características essenciais que a tornam única. O espectro de frequência é uma característica fundamental, cada fonte sonora tem padrões distintos de frequência que podem ser identificados na análise espectral. A amplitude e intensidade, são características que determinam o volume e a força

do som emitido pela fonte. Os padrões temporais, como duração, ritmo e pausas entre os sons podem revelar informações importantes sobre o comportamento da fonte sonora. E por fim, o ruído de fundo, pode fornecer detalhes como contexto sobre a localização e as condições em que a fonte sonora está operando.

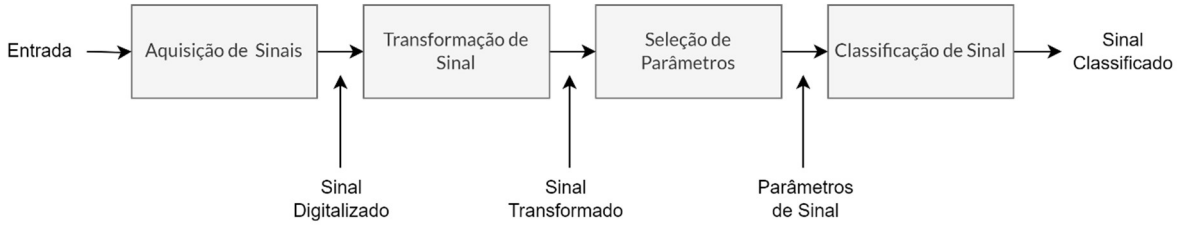
A assinatura de áudio é uma ferramenta poderosa para identificação e compreensão de fontes sonoras. Seja na indústria ou qualquer outra aplicação, a análise da assinatura de áudio proporciona informações valiosas sobre as características únicas do som emitido.

3.2 PROCESSAMENTO DE SINAIS

O processamento de sinais é uma área da engenharia e ciência da computação que trabalha com a manipulação, análise e interpretação de sinais, que são representações matemáticas de fenômenos físicos, como som, imagens, sinais biológicos, dados de sensores, entre outros. Oppenheim e Schafer (2010), elencaram que o principal objetivo do processamento de sinais é extrair informações úteis e relevantes desses sinais, seja para fins de análise, interpretação, compressão, reconhecimento de padrões ou tomada de decisões. Os sinais podem ser analógicos ou digitais, e o processamento de sinais pode ocorrer em tempo real ou em tempo *offline*, dependendo da aplicação.

Conforme Smith (2021) comentou sobre processamento de sinais, existem várias etapas envolvidas no processamento de sinais como aquisição de sinal, conversão ou transformação de sinal, filtragem ou seleção de parâmetros e análise espectral, conforme principais etapas do processamento de sinais ilustradas na Figura 12.

Figura 12 - Principais Etapas no Processamento de Sinais.



Fonte: Tuleski, 2023.

A aquisição de sinais é um dos componentes essenciais no processamento de sinais, e seu sucesso depende diretamente da eficiência técnica e da escolha adequada da instrumentação ou método de coleta de dados. A precisão e fidelidade da aquisição de sinais são cruciais para garantir a

qualidade dos dados que alimentam os subsequentes estágios de processamento. Em consonância com essa importância, Lyons (2011) destacou a necessidade de uma abordagem rigorosa na aquisição de sinais, enfatizando a importância de evitar erros sistemáticos e de calibração adequada dos sistemas de aquisição. Além disso, os trabalhos de Rassoul *et al.* (2011) ofereceram percepções valiosas sobre a seleção de equipamentos apropriados para a aquisição de diferentes tipos de sinais, considerando fatores como taxa de amostragem, resolução e faixa dinâmica.

No segundo passo, conversão de sinais, é o momento em que ocorre a transformação de informações analógicas ou digitais de um domínio para outro. Nesse contexto a escolha do método de conversão apropriado é uma decisão fundamental em projetos que envolvem aquisição e processamento de sinais. Dentre os diversos métodos de conversão de sinais existentes, alguns se destacam por suas características e aplicações específicas. Bin *et al.* (2005) propuseram que a conversão analógico-digital (ADC) é amplamente empregada para transformar sinais analógicos em formato digital, tornando-os passíveis de processamento digital subsequente. Por outro lado, Proakis e Manolakis (2006) propuseram que a conversão digital-analógica (DAC) é fundamental para recriar sinais analógicos a partir de dados digitais, sendo essencial em sistemas de reprodução de áudio e comunicação. Enquanto isso, Huang e Temes (1994), defenderam que a conversão sigma-delta, conhecida por sua alta precisão em aplicações de baixa frequência, encontra aplicação em dispositivos como sensores de pressão e temperatura. Por fim, Stephan (2010), trouxeram o conceito de que a conversão de tempo a digital (ATDC) é valiosa em sistemas de medição de alta resolução, como os utilizados em radar de comunicações. Portanto, a compressão aprofundada desses métodos e sua escolha criteriosa são essenciais para o sucesso de projetos de processamento de sinais em diversas áreas.

A terceira etapa, extração ou seleção de parâmetros é de suma importância no processamento de sinais, desempenhando um papel crítico na qualidade e na relevância das informações obtidas. A escolha cuidadosa dos parâmetros a serem extraídos ou selecionados é essencial para garantir que apenas os aspectos mais relevantes dos dados sejam considerados na análise. De acordo com Oppenheim *et al*. (2010), dentre as técnicas de extração de parâmetros, a transformada de Fourier é amplamente empregada para representar sinais no domínio da frequência, permitindo a identificação de componentes espectrais relevantes. Além disso, Bovik (2009) em seu estudo, utilizou métodos de extração baseados em estatísticas, como média, desvio padrão, e momentos estatísticos, são frequentemente utilizados para caracterizar distribuição de valores de um sinal. Por outro lado, Jolliffe (2002), propôs a seleção de parâmetros para identificar um subconjunto dos parâmetros disponíveis que são mais informativos para uma determinada tarefa de análise. Métodos de seleção de recursos,

como a análise de componentes principais (PCA) e algoritmos de seleção de recursos baseados em aprendizado de máquina, são utilizados para esse fim.

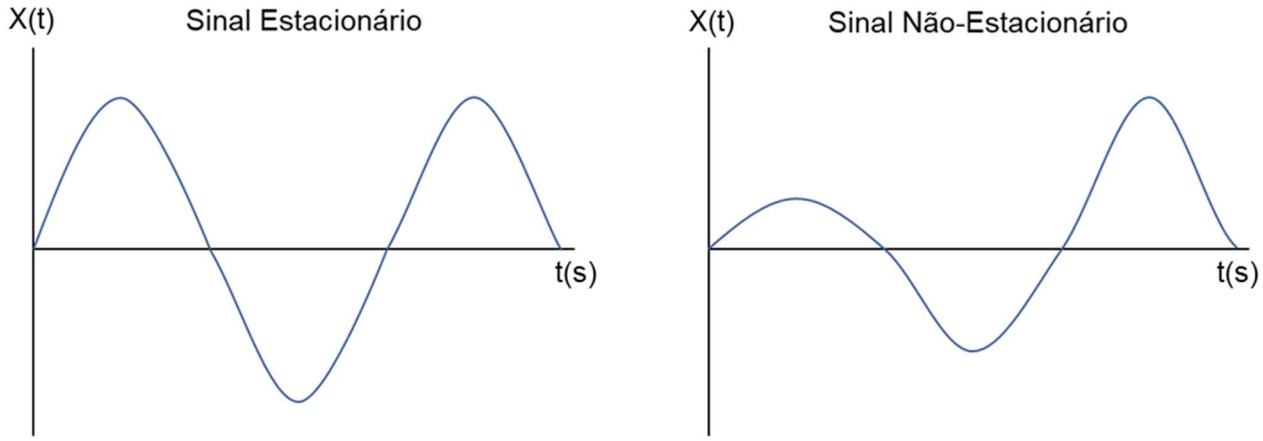
Por fim, a etapa de análise espectral envolve a decomposição de um sinal no domínio da frequência para identificar os componentes espectrais que o compõem. Conforme o estudo de Mallat (1998), técnicas avançadas como a análise de *wavelet* permitem uma análise mais detalhada e adaptável da estrutura espectral do sinal.

Huang *et al.* (2014) iniciaram a discussão que o processamento de sinais também envolve o uso de algoritmos e técnicas de processamento de dados, como algoritmos de ML e inteligência artificial, para reconhecer padrões complexos nos sinais e tomar decisões ou realizar tarefas específicas, como reconhecimento de fala, processamento de imagens, diagnóstico médico, processamento de áudio, entre outros. O processamento de sinais tem uma ampla gama de aplicações em várias áreas, incluindo comunicações, processamento de áudio e vídeo, medicina, processamento de imagem, processamento de fala, geofísica, processamento de radar e sonar, processamento de sinais biomédicos, entre outros. É uma área em constante evolução, impulsionada por avanços na tecnologia de hardware e *software*, e continua a desempenhar um papel crucial em muitos campos da ciência e engenharia.

Resumindo, o processamento de sinais envolve técnicas que permitem a análise de sinais elétricos, sonoros, visuais e outros tipos de sinais para extrair informações relevantes. Como observado na Figura 13, dois tipos comuns são os sinais estacionários e não-estacionários. Sinais estacionários são aqueles que têm suas propriedades estatísticas, como média e variância, constantes ao longo do tempo, ou seja, suas características não mudam com o tempo. Já sinais não estacionários apresentam variações nas suas propriedades estatísticas ao longo do tempo, ou seja, suas características mudam com o tempo. É importante distinguir entre os tipos de sinais, pois as técnicas de processamento e análise de dados podem variar de acordo com as características de cada tipo de sinal.

Na indústria automotiva, processamento de sinais tem sido cada vez mais utilizado no diagnóstico de falhas em veículos. Uma das formas de utilizar o processamento de sinais no diagnóstico de falhas em veículos é por meio da análise do som emitido pelo motor. O som emitido pelo motor pode indicar a presença de falhas, como por exemplo, o ruído de um rolamento desgastado ou o som de um motor desregulado. Essa análise pode ser realizada utilizando-se técnicas como a análise espectral, que permite a identificação das frequências presentes no sinal de áudio, e a filtragem adaptativa, que pode ser utilizada para remover ruídos de fundo e outras interferências.

Figura 13 - Perfis de (a) sinal estacionário e (b) sinal não estacionário.



Fonte: Tuleski, 2023.

Outra forma de utilizar o processamento de sinais no diagnóstico de falhas em veículos é por meio da análise do sinal de vibração. A vibração é uma medida importante da saúde mecânica de um veículo, e a análise de sinais de vibração pode identificar a presença de falhas em componentes como motor, o sistema de transmissão, o sistema de injeção e assim por diante. Técnicas como a análise de frequência e análise de componentes principais podem ser utilizadas para analisar sinais de vibração e identificar a presença de falhas.

Alguns pesquisas abordam o uso do processamento de sinais para diagnóstico de falhas em veículos incluem o estudo de Al-Bugharbee *et al.* (2015), que utilizaram a análise espectral para identificar a presença de falhas em rolamentos, e o estudo de Sun *et al.* (2016), que empregaram a análise de componentes principais para identificar falhas em sistemas de transmissão. Em conclusão, o processamento de sinais é poderoso para o diagnóstico de falhas em veículos, e pode ser utilizado em conjunto com outras técnicas, como a análise visual e inspeção manual, para garantir a integridade mecânica de um veículo e a segurança dos usuários.

3.3 TRANSFORMADA DE FOURIER CONTÍNUA E DISCRETA

A Transformada de Fourier permite decompor um sinal ou função em suas componentes de frequência, revelando a contribuição de cada frequência para a representação do sinal no domínio da frequência. Conforme Bracewell e Ronald (1986) definiram, a Transformada de Fourier pode ser contínua ou discreta, são duas abordagens relacionadas, mas distintas, para analisar sinais no domínio da frequência. A Transformada de Fourier de um sinal contínuo $x(t)$ pode ser definida como,

$$X(k) = \int_{-\infty}^{\infty} x(t).e^{-i2\pi ft}dt \quad (1)$$

onde $X(k)$ representa a representação em frequência do sinal, f é a frequência, t é o tempo contínuo, i é a unidade imaginária e $\int$ denota a integral. Essa equação descreve como a Transformada de Fourier calcula a contribuição de cada componente de frequência para a representação do sinal no domínio da frequência.

A DFT é utilizada para analisar sinais discretos no tempo, com uma sequência finita de amostras. Ela transforma um sinal $x(n)$, onde n é uma variável discreta, em sua representação no domínio da frequência. A Transformada Discreta de Fourier poder ser representada por,

$$X(k) = \int_{n=0}^{N-1} x(n).e^{-i2\pi ft/N}dt \quad (2)$$

onde $X(k)$ representa a representação em frequência do sinal, k é uma frequência discreta variando de 0 a $N-1$, i é a unidade imaginária, e N é o número de amostras no sinal. A DFT é amplamente utilizada em sistemas digitais e computadores para a análise de sinais amostrados, como áudio digital, processamento de imagens e comunicação digital.

3.4 TRANSFORMADA RÁPIDA DE FOURIER

A Transformada Rápida de *Fourier* (FFT, na sigla em inglês) é uma técnica matemática que permite a análise de sinais complexos, como o som e a vibração em veículos, para identificar suas frequências dominantes ou outros aspectos relevantes. Em conjunto com a inteligência artificial, a FFT pode ser aplicada no diagnóstico de falhas em veículos por meio da análise de som do motor.

A FFT, é um conjunto de algoritmos eficientes para calcular a Transformada de Fourier discreta (DFT), desempenha um papel central no processamento de sinais, oferecendo uma poderosa ferramenta para analisar e extrair informações em domínio de frequência a partir de sinais no domínio do tempo. A eficiência computacional da FFT, especialmente na sua versão Cooley-Turkey, revolucionou o processamento de sinais, tornando-o aplicável em diversas áreas. Conforme Turkey e

Cooley (1965) trouxeram em seu trabalho, a FFT reduz significativamente a complexidade computacional em comparação com o cálculo direto da DFT.

Uma das aplicações da FFT no diagnóstico de falhas em veículos é a análise do espectro de frequência do som do motor. Esse espectro pode indicar a presença de falhas em componentes como rolamentos e pistões, cujas vibrações geram frequências específicas no som do motor. Por meio da FFT, é possível extrair essas frequências e identificar a presença de falhas. A aplicação da inteligência artificial nesse processo pode aumentar a precisão da identificação de falhas, ao utilizar algoritmos de ML para classificar padrões de frequência e identificar possíveis defeitos.

Um exemplo de estudo científico que aborda a aplicação da FFT e da inteligência artificial no diagnóstico de falhas em veículos é o trabalho de Lui *et al.* (2019), que utilizaram redes neurais convolucionais para identificar falhas em componentes de um veículo por meio da análise do espectro de frequência do som do motor. Outro exemplo é o estudo de Fakharian *et al.* (2019), que utilizaram técnicas de agrupamento de dados para identificar padrões de vibração em um veículo e identificar falhas em componentes como suspensão e transmissão.

Portanto, a FFT em conjunto com técnicas de inteligência artificial é uma estratégia promissora no diagnóstico de falhas em veículos, pois permite a análise precisa de sinais complexos como o som e a vibração, com potencial para melhorar a segurança e eficácia dos automóveis. A Transformada Rápida de *Fourier* também pode ser interpretada como um algoritmo que torna viável o cálculo da Transformada Discreta de *Fourier* (DFT). Matematicamente, considera-se um conjunto de pontos $N = 2^p$, em que p é um número inteiro, embora nem todos os algoritmos de FFT utilizem essa base elevada a 2.

Da definição de DFT, tem-se,

$$F_k = \sum_{n=0}^{N-1} f_n . e^{-i\frac{2\pi}{N}.k.n} \tag{3}$$

Dividindo o somatório por 2, chega-se,

$$F_k = \sum_{n=0}^{N/2-1} f_{2n} . e^{-i\frac{2\pi}{N/2}.k.n} + e^{-i\frac{2\pi}{N}.k} . \sum_{n=0}^{N/2-1} f_{2n+1} . e^{-i\frac{2\pi}{N/2}.k.n} \tag{4}$$

A primeira e segunda parcela da equação representam respectivamente a parte par e ímpar da transformada. As duas somas têm o mesmo expoente, que é dividido $N/2$. Dessa forma, torna-se evidente a relação entre o ponto k e $k + N/2$. Com essa relação, podemos ver que F_k e $F_{k+N/2}$ tem o mesmo expoente e podem ser calculadas ao mesmo tempo. Não somente isso, a nova forma da transformada pode ser sucessivamente dividida, cada vez produzindo limites menores.

3.5 TRANSFORMADA *WAVELET*

A Transformada *Wavelet* é uma técnica matemática poderosa que permite a análise de sinais em diferentes escalas de tempo e frequência conforme a definição de Mallat *et al.* (1998). Uma de suas principais aplicações é na compressão de dados de imagens e áudio, mas também tem sido utilizada em outras áreas, como na análise de séries temporais, processamento de sinais biomédicos e detecção de falhas em equipamentos industriais.

A ideia da transformada wavelet é decompor um sinal em diferentes escalas de tempo e frequência, anal detalhes finos e grosseiros que outras técnicas de análise de sinal não podem capturar facilmente. Opostamente a Transformada de *Fourier*, que analisa um sinal em termos de suas componentes de frequência, a Transformada *Wavelet* analisa o sinal em diferentes escalas de frequência e tempo.

A Transformada de Wavelet é definida por,

$$W(a,b) = \frac{1}{\sqrt{(a)}} . \int f(t)\psi\left(\frac{(t-b)}{a}\right) dt \qquad (5)$$

onde $W(a,b)$ é a transformada *wavelet* do sinal $f(t)$, $\Psi(t)$ é a função *wavelet*, a é o fator de escala e b é o deslocamento temporal. O fator de escala a determina a largura da função *wavelet* no domínio do tempo e da frequência. Quanto maior o valor de a, mais ampla será a função *wavelet* no domínio do tempo e mais estreita no domínio da frequência. O deslocamento temporal b determina a posição da função *wavelet* no eixo do tempo.

A função *Wavelet* $\Psi(t)$, em conformidade com Mallet *et al.* (1998), é uma função matemática que descreve uma forma de onda localizada no tempo e na frequência. Existem vários tipos de funções *Wavelet*, sendo a mais comum a *Wavelet* de *Haar* sendo descrita por,

$$\Psi(t) = \begin{cases} 1 & 0 \leq t < 1/2, \\ -1 & 1/2 \leq t < 1, \\ 0 & Caso\ Contrário \end{cases} \quad (6)$$

Outros exemplos de funções *Wavelet* incluem a *Wavelet* de *Daubechies*, a *Wavelet* de *Coiflet* e a *Wavelet* de *Morlet*.

A Transformada *Wavelet* é calculada em diferentes escalas de tempo e frequência, permitindo a análise de detalhes finos e grosseiros do sinal em diferentes níveis de resolução. A decomposição do sinal em diferentes escalas é realizada através da aplicação da transformada *Wavelet* em uma série de sinais suavizados, obtidos através da convolução do sinal original com uma função de suavização, geralmente a função de *Haar*, descrita pela Equação (6).

De acordo com Fungati *et al.* , a Transformada *Wavelet* é uma ferramenta matemática poderosa que permite a análise de sinais em diferentes escalas de tempo e frequência, oferecendo uma abordagem flexível e eficaz para a análise de sinais em diversas áreas da engenharia.

3.6 TRANSFORMADA WAVELET PACKET

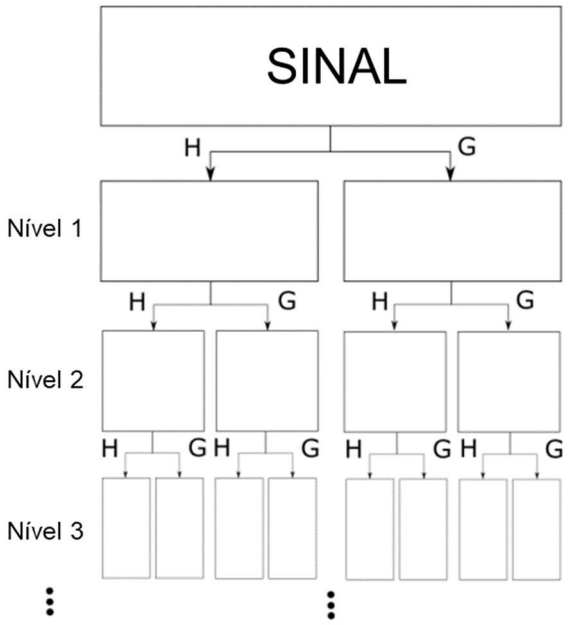
A WPT é uma extensão da DWT que permite a análise de sinais com maior flexibilidade e resolução. Enquanto a DWT divide o sinal em escalas e frequências discretas, a WPT divide o sinal em sub-bandas com diferentes resoluções de frequência e tempo (Wu e Liu, 2018). Essa capacidade de análise multi-resolução torna a WPT uma ferramenta muito utilizada para a análise de sinais complexos, como os encontrados em processamento de imagem, comunicações e análise de dados, conforme dita Daubechies *et al.* (1992).

A WPT é definida como uma transformada linear que leva um sinal $f(t)$ em um conjunto de coeficientes de *Wavelet Packet* $c(j,k)$ e um conjunto de vetores de base de *Wavelet Packet* $\Psi_j, k(t)$. A WPT é definida por,

$$f(t) = c(j,k)\Psi_j, k(t) \quad (7)$$

onde j é o índice de escala e k é o índice de deslocamento. Os coeficientes de *Wavelet Packet* $c(j,k)$ são calculados a partir da decomposição do sinal em sub-bandas usando filtros passa-baixa e passa-alta, como mostra a Figura 14.

Figura 14 - Representação da transformada Wavelet, em que H e G representam filtros de passagem de baixa e alta respectivamente.



Fonte: Tuleski, 2023.

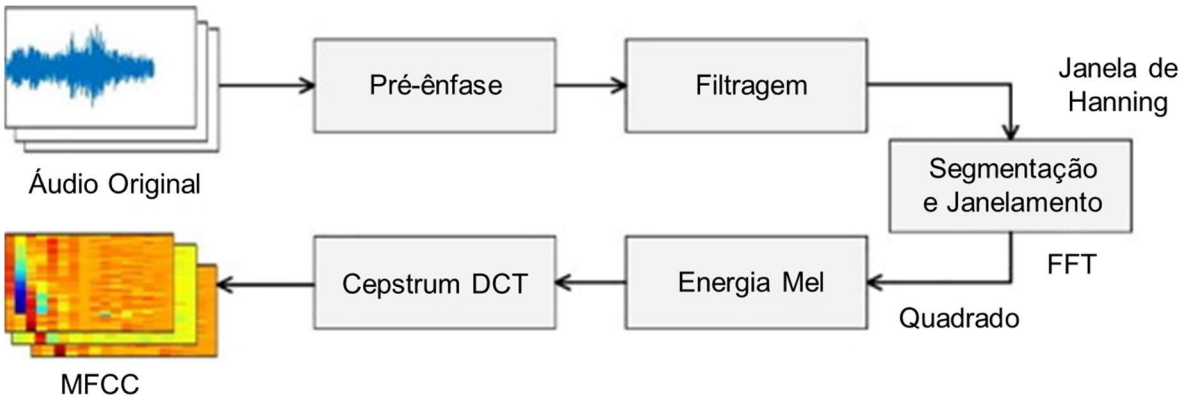
O sinal é segmentado em pedaços de curta duração e então a Transformada *Wavelet* é aplicada, resultando em 2^j sinais transformados para cada pedaço. Após isso, recursos baseados em estatística são computados para cada sinal transformado. Múltiplas estatísticas podem ser extraídas dos nós dos sinais, tais como por exemplo, curtose, assimetria, média, comprimento de onda, contagem cruzada média, quantidade de mudanças e raiz quadrada média. Então, a extração de uma caraterística é sucedida de um algoritmo de seleção de características, em que, seleciona os recursos de cada nó da Transformada *Wavelet*, analisando a correlação entre o recurso e a saída (classes) e entre os próprios recursos. Esse procedimento descarta características não necessárias e duplicadas, como as com baixa correlação com a saída ou as altamente correlacionadas entre elas mesma.

Uma das principais aplicações da WPT é na compressão de dados. A WPT pode ser utilizada para representar sinais complexos com maior eficiência em termos de armazenamento e transmissão. Além disso, a WPT é usada em análise de sinais biomédicos, como eletroencefalografia (EEG) e eletrocardiografia (ECG), para extrair informações relevantes do sinal e diagnosticar doenças.

3.7 *MEL-FREQUENCY CEPSTRAL COEFFICIENTS*

O processamento de sinais de áudio tem sido objeto de pesquisa em diversas áreas, incluindo identificação e diagnóstico de falhas em suas demasiadas aplicações. De acordo com Smith *et al.* (1999), citaram em seu livro, que uma técnica amplamente utilizada nessa área é a extração de MFCC, que tem sido bem-sucedida em capturar as características acústicas da fala. Mitra (2006), define em seu livro, que o processo de extração de MFCC pode ser descrito com base nas seguintes etapas apresentadas na Figura *15*.

Figura 15 - Passo a passo para a obtenção de uma MFCC.



Fonte: O Autor, 2023.

Em primeiro lugar, durante a pré-ênfase, o sinal de áudio é pré-processado para melhorar a relação sinal-ruído e realçar as características de alta frequência. A seguir representa-se a operação de pré-ênfase,

$$y(n) = x(n) - \propto . x(n-1) \tag{8}$$

onde $x(n)$ é o sinal de entrada, $y(n)$ é o sinal pré-ênfase e $\propto$ é o coeficiente de pré-ênfase.

Na sequência, na fase de janela, o sinal pré-ênfase é dividido em segmentos de curta duração, geralmente de 20 a 40 milissegundos, usando uma janela de *Hamming*, definida pela Equação (9), ou outra janela de análise. Isso é necessário para evitar artefatos devido à convolução com a Transformada de *Fourier*.

$$w[n] = 0{,}54 - 0{,}46.\cos(2\pi n/(n-1)) \quad (9)$$

onde $w[n]$ é a amostra da janela de *Hamming* no tempo n, e N é o tamanho da janela.

Em terceiro lugar, a Transformada de *Fourier*, mais especificamente a Transformada Discreta de *Fourier* (DFT) é aplicada a cada segmento de áudio para obter seu espectro de magnitude. A DFT é definida por,

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \quad (10)$$

onde $x(n)$ é o sinal de entrada, $X(k)$ é o espectro de magnitude e N é o comprimento da janela.

Em seguida, a próxima etapa é conhecida como banco de filtros *Mel*, realiza a passagem do espectro de magnitude por um banco de filtros *Mel* para extrair as características relevantes do sinal de áudio. O banco de filtros *Mel* consiste em um conjunto de filtros triangulares espaçados linearmente em frequências de *Mel*. A transformação de frequência *Mel* é definida por,

$$Mel(f) = 2595 * log_{10}(1 + f/7000) \quad (11)$$

onde f representa a frequência em *Hertz* (Hz). Já a equação para calcular o filtro *Mel* pode ser descrita por,

$$H_m(k) = \begin{cases} 0, & para\ k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)}, & para\ f(m-1) \le k < f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)}, & para\ f(m) \le k < f(m+1) \\ 0, & para\ k \ge f(m+1) \end{cases} \quad (12)$$

onde $H_m(k)$ é a reposta do filtro *Mel* para o filtro m e k, $f(m-1)$, $f(m+1)$ e $f(m)$ são as frequências de início, centro e final do filtro m.

O próximo passo, o logaritmo do espectro de magnitude filtrado é calculado para converter a escala linear em uma escala logarítmica. Isso é feito porque a audição humana é mais sensível às diferenças de intensidade em baixas frequências do que em altas frequências.

Na sequência, a Transformada de Cosseno Discreta (DCT) é aplicada ao algoritmo do espectro filtrado para obter os coeficientes *cepstrais*. A DCT é uma técnica de processamento de sinais que converte um sinal do domínio do tempo para o domínio da frequência, similar à Transformada de *Fourier*. A DCT, definida pela Equação (13), é usada no método MFCC pela capacidade de concentrar a maior parte da energia do sinal em um número limitado de coeficientes, o que torna a representação do sinal mais compacta.

$$DCT_i(n) = \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} x_m \cos\left(\frac{\pi}{N}\left(m + \frac{1}{2}\right) i\right) \tag{13}$$

onde N é o número de amostras do sinal de entrada, x_m é a amostra m do sinal de entrada e $DCT_i(n)$ é o coeficiente i do vetor DCT. O índice n representa o quando de tempo atual.

Por fim, os coeficientes *cepstrais* são normalizados para melhorar a robustez do método a variações na intensidade do sinal e nos efeitos do ruído. Isso é feito subtraindo-se a média dos coeficientes *cepstrais* de cada quadro e, em seguida, dividindo-se o resultado pelo desvio padrão dos coeficientes *cepstrais* do quadro.

CAPÍTULO 4
MODELOS DE *MACHINE LEARNING* - CLASSIFICAÇÃO

4.1 INTRODUÇÃO

Segundo Hastie *et al.* (2009), os modelos de *Machine Learninf* (ML) para classificação são algoritmos que permitem que um sistema aprenda a classificar dados em categorias ou classes predefinidas. Esses modelos são usados em uma ampla variedade de aplicações, tais como reconhecimento de padrões, diagnóstico médico, detecção de fraudes, análise de sentimentos, entre outros. Existem diversos tipos de modelos de ML para classificação, cada um com suas características e aplicabilidades específicas. Vale mencionar alguns dos modelos mais comuns. Cada modelo tem suas vantagens, desvantagens e suposições subjacentes, e a escolha do modelo mais adequado depende das características do problema e dos dados disponíveis. Seguem alguns destes modelos que serão utilizados nesta pesquisa.

Florestas Aleatórias (Random Forests): De acordo com Breiman (2001), uma floresta aleatória é composta por múltiplas árvores de decisão treinadas em subconjuntos aleatórios do conjunto de dados original. A classificação final é determinada pela maioria das classificações individuais das árvores. Esse modelo é robusto, evita o *overfitting* e é eficiente em grandes conjuntos de dados.

Regressão Logística: Apesar do nome, Hosmer *et al.* (2013), citaram que a regressão logística é um modelo de classificação que estima a probabilidade de uma instância pertencer a uma classe específica. A saída é uma probabilidade entre 0 e 1, e um limite de decisão é aplicado para atribuir a instância a uma classe.

Máquinas de Vetores de Suporte (SVM): As SVMs, de acordo com Cortes e Vapnik (1995), são modelos que mapeiam os dados para um espaço dimensional maior e procuram encontrar o hiperplano que melhor separa as classes. Esses modelos são eficazes em problemas com conjuntos de dados de alta dimensionalidade e podem lidar com classes não linearmente separáveis usando *kernel*.

Redes Neurais: Goodfellow *et al.* (2016) abordaram que as redes neurais são modelos inspirados no funcionamento do cérebro humano. Eles consistem em camadas de neurônios artificiais interconectados, onde cada neurônio recebe entradas, aplica uma função de ativação e passa o resultado para os neurônios da próxima camada. As redes neurais profundas, também conhecidas

como DL, são capazes de aprender representações complexas e têm tido sucesso em muitas tarefas de classificação, como reconhecimento de imagem e processamento de linguagem natural.

Os modelos de classificação de ML são algoritmos que têm como objetivo identificar e categorizar dados com base em padrões e características específicas. Eles são amplamente utilizados em diversas áreas, como reconhecimento de fala, análise de imagens, detecção de fraudes e diagnósticos médicos.

Esses modelos utilizam dados rotulados para aprender e classificar novos dados com base em suas características, e podem ser treinados utilizando técnicas de aprendizado supervisionado ou não supervisionado. Alguns desses modelos foram selecionados e serão aplicados neste estudo *Randon Forest, Gradient Boosting, Support Vector Machine, Multilayer Perceptron* e *Convolutional Neural Network*.

4.2 HIPERPARÂMETROS

Na área de aprendizado de máquina, especialmente em contextos de treinamento de modelos complexos, um elemento crucial que desempenha um papel fundamental no processo de otimização é a definição de hiperparâmetros. Hiperparâmetros são parâmetros ajustáveis que não são aprendidos diretamente durante o processo de treinamento do modelo, mas têm um impacto significativo em seu desempenho e generalização. Eles abrangem uma ampla gama de aspectos, como taxas de aprendizado, número de camadas em uma rede neural, tamanho do *batch* e valores de regularização. A otimização adequada desses hiperparâmetros pode resultar em melhorias substanciais no desempenho e na capacidade de generalização do modelo. Portanto, a seleção criteriosa e a sintonia eficaz dos hiperparâmetros são essenciais para alcançar resultados robustos e confiáveis em tarefas de aprendizado de máquina complexas.

Na busca pela otimização dos hiperparâmetros, diversas abordagens têm sido empregadas para encontrar combinações que maximizem o desempenho dos modelos de aprendizado de máquina. Cada uma dessas abordagens apresenta características distintas que se adequam a diferentes cenários e recursos computacionais.

O *GridSearch*, por exemplo, de acordo com a definição de Belete e Manjaiah (2021) envolve a definição prévia de uma grade de valores possíveis para cada hiperparâmetros. A busca sistemática por todas as combinações possíveis dentro dessa grade é realizada, avaliando o desempenho do modelo por meio de validação cruzada. Embora seja abrangente, o *Grid Search* pode se tornar

computacionalmente custoso, especialmente em espaços de busca com muitas dimensões, o que limita sua aplicabilidade em cenários com restrições de tempo.

Por outro lado, Bergstra e Bengio (2012) elencaram que o *Randomized Search* oferece uma abordagem mais eficiente em termos de tempo, selecionando aletoriamente um conjunto de combinações de hiperparâmetros para avaliação. Essa estratégia permite explorar um subconjunto relevante do espaço de busca, reduzindo consideravelmente o esforço computacional, enquanto ainda mantém a busca por boas configurações. Essa abordagem é particularmente útil quando o espaço de hiperparâmetros é extenso e a busca exaustiva se torna inviável, permitindo uma exploração mais ampla e eficaz.

A otimização *Bayesiana*, abordada no estudo de Snoek *et al.* (2012), como implementada no *Bayes Search,* introduz uma abordagem probabilística mais avançada. Ela utiliza um modelo estatístico para estimar a relação entre as configurações de hiperparâmetros e o desempenho do modelo. Ao longo das iterações, o algoritmo atualiza as estimativas do modelo com base nos resultados das avaliações anteriores, focando mais nas regiões promissoras do espaço de busca. Essa abordagem permite uma busca mais eficiente por soluções ótimas, tornando-se vantajosa em cenários onde o tempo de busca é um fator crítico.

Além das estratégias convencionais de otimização de hiperparâmetros, outras abordagens que ganhou destaque é o *Fine Tuning*, e foi principalmente exposta por Bergstra e Bengio (2012). Essa técnica envolve um ajuste mais refinado dos hiperparâmetros após a busca inicial, com base nos resultados obtidos. A ideia é concentrar os esforços em uma região estreita e específica do espaço de busca. O *Fine Tuning* pode ser particularmente mais eficaz quando já se possui um conhecimento razoável sobre o espaço de hiperparâmetros após a busca inicial ou quando a busca global não atingiu o desempenho desejado. Essa abordagem permite um refinamento preciso das configurações, levando a ganhos adicionais no desempenho do modelo.

4.3 VALIDAÇÃO CRUZADA

A validação cruzada (VC), com base na definição de Hastie *et al.* (2009), é uma técnica essencial no campo de aprendizado de máquina e estatística, amplamente utilizada para validar o desempenho e a generalização de modelos preditivos. Ela visa mitigar o risco de superajuste e fornecer estimativas confiáveis do desempenho do modelo em dados não observados. A ideia fundamental da VC é dividir o conjunto de dados em várias partições ou *folds* e realizar múltiplos treinamentos e testes, alternando os conjuntos de treinamento e teste em cada iteração. A métrica de desempenho, tais como acurácia

ou erro quadrático médio, é então calculada para cada *fold* e combinada para obter uma medida geral do desempenho do modelo.

De acordo com Kohavi (1995) a validação cruzada pode ser representada da seguinte forma. Seja D um conjunto de dado de tamanho N, onde $D = \{(X_1, y_1), (X_2, y_2), \dots, (X_N, y_N)\}$, em que X_i representa as características e y_i representa os rótulos.

O processo de validação cruzada segue a seguinte iteração, em primeiro lugar o conjunto de dados D é dividido em k partições disjuntas ou *folds*, geralmente $k = 5$ ou $k = 10$. Cada *fold* é denotado como $D_1, D_2, \dots, D_k$. O processo de validação cruzada consiste em k iterações. Em cada iteração i, um dos *folds* D_i é usado como conjunto de teste, enquanto os outros *folds* são combinados para formar o conjunto de treinamento, denotado como D_{train}. O modelo treinado em D_{train} é avaliado em D_i. Para cada iteração i, uma métrica de desempenho P_i, como a acurácia ou o erro médio quadrático, é calculada com base nos resultados do modelo. Após k iterações, calculamos a média das métricas de desempenho P_i para obter uma estimativa geral do desempenho do modelo,

$$Desempenho\ Médio = \frac{1}{k}\sum_{i=1}^{k} P_i \tag{24}$$

Essa equação representa a estrutura matemática da validação cruzada e como as métricas de desempenho são calculadas e resumidas para avaliar um modelo em uma variedade de partições dos dados originais. A VC ajuda a evitar problemas de viés na avaliação do modelo, uma vez que todos os dados são usados tanto para treinamento quanto para teste, e a média das métricas fornecerá uma estimativa mais precisa do desempenho do modelo em dados não vistos.

4.4 RANDOM FOREST

O método de inteligência artificial *Random Forest* (RF) é uma técnica popular e poderosa de ML que pode ser utilizada para classificação, regressão e outras tarefas de previsão. O algoritmo foi introduzido por Breiman *et al.* (2001) e é baseado em um conjunto de árvores de decisão, em que cada árvore é treinada em uma amostrada aleatória dos dados de treinamento e as previsões são feitas pela média das previsões individuas de cada árvore.

As árvores de decisão são construídas por meio da divisão dos dados em subconjuntos menores, baseados em regras simples. Em cada nó da árvore, uma pergunta é feita sobre uma das variáveis do conjunto de dados e as respostas são usadas para dividir o conjunto de dados em dois ou mais subconjuntos menores. Essas perguntas são baseadas em medidas de impureza, como a entropia

ou o índice de *Gini*, e o objetivo é encontrar a pergunta que reduz a impureza do conjunto de dados ao máximo possível.

O algoritmo *Random Forest* utiliza várias árvores de decisão em conjunto para reduzir o risco de *overfitting* e aumentar a precisão das previsões. Cada árvore é treinada em uma amostra aleatória dos dados de treinamento, com substituição, ou seja, alguns dados são repetidos e outros dados são deixados de fora. Esse processo é conhecido como *bootstrapping*. A cada divisão em um nó da árvore, apenas um subconjunto aleatório das variáveis é considerado, o que ajuda a evitar a criação de árvores correlacionadas.

A previsão de uma na amostra é feita pela média das previsões individuais de cada árvore. Na classificação, a previsão final é classe mais comum entre as previsões e, na regressão, é a média das previsões.

Conforme ilustra a Figura 16, para demonstrar a lógica por trás de uma *Decision Tree*, podemos imaginar que existe conjunto de dados que é composto por duas vezes o número um e cinco vezes o número zero, portanto pode-se diferenciar esse conjunto em duas classes, a primeira contendo todos os números um e a segunda contendo todos os números zeros. Neste exemplo, o objetivo é separar os dados do conjunto nas classes corretas utilizando apenas suas caraterísticas, que podem ser definidas pela cor e se o número é sublinhado ou não.

Inicialmente, a característica cor parece óbvia para classificar os dados, porém um dos zeros não é azul. Dessa forma, podemos iniciar a primeira fase da separação dos dados com a pergunta se a caraterística cor é vermelha ou não. Nesse momento, pode-se interpretar essa primeira pergunta como sendo o nó de uma árvore que separa o caminho em dois – os dados que atendem aos critérios descem pelo ramo da esquerda e os que não, descem pelo ramo da direita.

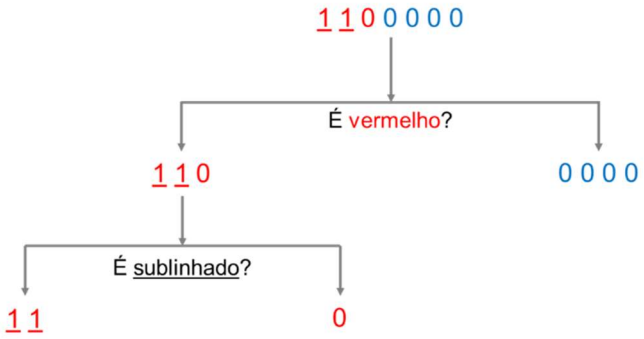
O ramo da direita contém todos os zeros com a cor azul, porém no ramo da esquerda, ainda temos dados que podem ser divididos mais uma vez. Agora, podemos utilizar uma pergunta diferente, se os dados estão sublinhados ou não. Por fim, a *Decision Tree* foi capaz de utilizar duas características para separar os dados corretamente.

A partir das previsões de cada árvore, o RF calcula a previsão final usando uma votação majoritária para classificação ou uma média para regressão. Como cada árvore é independente, a probabilidade de erro de cada árvore é reduzida, o que leva a uma previsão mais estável e geralmente mais precisa. A equação matemática para o RF pode ser descrita como,

$$f(x) = \left(\frac{1}{T}\right)\Sigma_t = 1^T(f_t(x)) \tag{35}$$

onde $f(x)$ é a previsão final para a observação x, T é o número de árvores no conjunto de árvores e $f_t(x)$ é a previsão da árvore t para a observação x.

Figura 16 - Exemplo de uma *Decision Tree*.



Fonte: Tuleski, 2023.

Para construir o modelo de RF, é necessário escolher o número de árvores a serem incluídas no conjunto de árvores e os parâmetros de cada árvore, como a profundidade máxima da árvore o número mínimo de amostras necessárias para dividir um nó. Esses parâmetros devem ser ajustados usando validação cruzada para encontrar a combinação ideal que maximize a precisão do modelo.

Os hiperparâmetros desempenham um papel fundamental na construção de modelos RF e afetam diretamente o desempenho e comportamento do algoritmo. Explorararemos alguns dos principais hiperparâmetros do RF e como cada um deles influencia o modelo, com base nos descritos na Tabela 2.

Tabela 2 - Principais hiperparâmetros de uma RF.

Hiperparâmetros	**Descrição**
n_estimators	Número de árvores
max_depth	Profundidade máxima das árvores
min_samples_leaf	Número mínimo de amostras em folhas
min_samples_split	Número mínimo de amostras por divisão
max_features	Número máximo de atributos
bootstrap	*Bootsrap sampling*
criterion	Critério de divisão

Fonte: O Tuleski, 2023.

Na Tabela 3, pode-se ver as principais vantagens e desvantagens dos hiperparâmetros do modelo RF.

Tabela 3 – Vantagens e Desvantagens dos principais hiperparâmetros da RF.

Hiperparâmetros	**Descrição**	**Vantagens**	**Desvantagens**
n_estimators	Número de árvores	Robustez do modelo e *overfitting.*	Complexidade computacional.
max_depth	Profundidade máxima das árvores	Generalização do modelo e *overfitting*.	Captura de relações complexas nos dados.
min_samples_leaf	Número mínimo de amostras em folhas	Complexidade modelo e *overfitting*.	Sensibilidade à nuances nos dados.
min_samples_split	Número mínimo de amostras por divisão	Divisão excessiva e *overfitting*.	Árvores rasas e menos representativas.

Fonte: Tuleski, 2023.

O *overfitting* - ou sobreajuste - é visto repetidamente como uma vantagem comum entre os hiperparâmetros. De acordo com Hastie (2009), isso é um fenômeno crítico no aprendizado de máquina e na modelagem estatística. Ele ocorre quando um modelo se adapta de forma excessiva aos dados de treinamento, capturando não apenas os padrões verdadeiros, mas também o ruído e variabilidade aleatória presente nesses dados. Bishop (1995), ressalta em seus estudos que isso pode resultar em um modelo que apresenta um desempenho excelente nos dados de treinamento, com erros muito baixos, mas falhas em generalizar eficazmente para novos dados não vistos, onde seu desempenho frequentemente degrada.

Outros hiperparâmetros como o número máximo de recursos (max_features) define o número máximo de recursos a serem considerados em cada divisão de nó. Isso introduz aleatoriedade e diversidade no modelo, tornando-o menos suscetível ao *overfitting*. Um valor comum é a raiz quadrada do número total de atributos. O *bootstrap sampling* (*bootstrap*) controla se o modelo utiliza ou não amostragem com reposição *bootstrap* ao criar conjuntos de dados para treinamento das

árvores. A amostragem *bootstrap* introduz mais variabilidade nos dados de treinamento e ajuda a melhorar a generalização do modelo. E por fim, o critério de divisão (*creiterion*), define a função utilizada para medir a qualidade das divisões nos nós das árvores. As opções mais comuns incluem "*gini*" para o índice Gini e "*entropy*" para a entropia. A escolha do critério pode influenciar a forma como o modelo faz divisões e, consequentemente, sua eficácia em diferentes cenários.

4.3 GRADIENT BOOSTING

O *Gradient Boosting* é um método de ML supervisionado que constrói um modelo preditivo por meio da combinação de várias árvores de decisão, de modo que cada árvore corrija os erros cometidos pelas anteriores. O método foi introduzido por Friedman *et al.* (1999) tem sido amplamente utilizado em várias aplicações incluindo previsão de demanda, detecção de fraudes e análise de riscos.

O *Gradient Boosting* é um método de *ensemble learning* que constrói um modelo preditivo por meio da combinação ponderada de várias árvores de decisão. Diferentemente do RF, o *Gradient Boosting* constrói cada nova árvore para ajustar os erros cometidos pelas árvores anteriores, em vez de escolher aleatoriamente subconjuntos de dados de treinamento e variáveis explicativas.

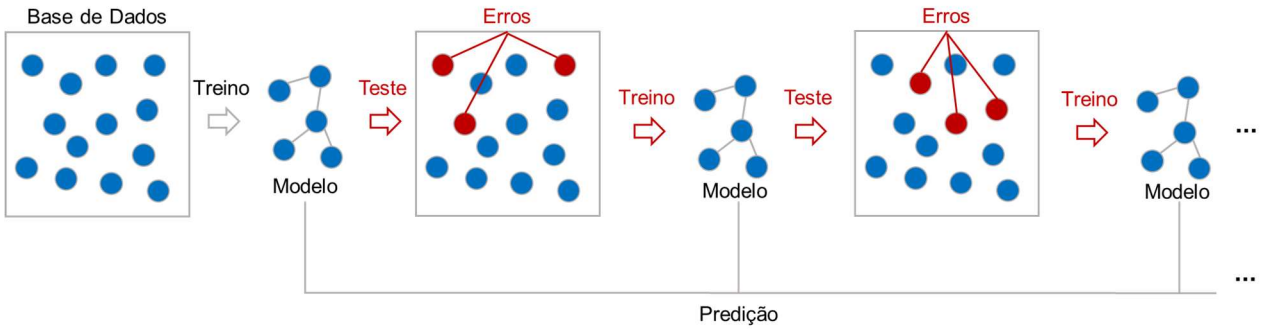
O processamento de construção do modelo começa com uma árvore de decisão simples, chamado de árvore base (*base tree*), que é treinada com os dados de treinamento e usada para fazer previsões iniciais. A partir das previsões iniciais, é calculado o erro residual, ou seja, a diferença entre as previsões iniciais e as observações reais. Em seguida, uma árvore é construída para ajustar os erros residuais, adicionando-a ao modelo. Esse processo é repetido várias vezes, adicionando novas árvores para corrigir os erros das anteriores, até que o modelo atinja um número máximo de árvores ou o erro residual não possa mais ser reduzido, conforme ilustrado na Figura 17.

A combinação ponderada das previsões de todas as árvores é feita de acordo com a soma ponderada, tal como:

$$f(x) = \Sigma_m = 1^M \lambda_m . h_m(x) \tag{46}$$

onde $f(x)$ é a previsão final para a observação x, m é o número de árvores no modelo e $h_m(x)$ é a previsão da árvore m para observação x, λ_m é o peso (*learning rate*) da árvore m, que controla a contribuição de cada árvore para o modelo final.

Figura 17 - Demonstração do Gradient Boosting.



Fonte: Tuleski, 2023.

A Equação (17), define o erro residual.

$$r_i = y_i - f_{i-1}(x_i) \tag{17}$$

onde r_i é o erro residual para a observação i, y_i é o valor real para a observação i, $f_{i-1}(x_i)$ é a previsão atual para a observação i, que é a soma ponderada das previsões de todas as árvores construídas até o momento $i-1$. O ajuste fino do desempenho do *Gradient Boosting* envolve o ajuste dos hiperparâmetros, que são configuráveis antes do treinamento. A Tabela 4 representa os principais hiperparâmetros do *Gradient Boosting*.

Tabela 4 - Principais hiperparâmetros do *Gradient Boosting*.

Hiperparâmetros	**Descrição**
n_estimators	Número de árvores
learning_rate	Taxa de Aprendizado
max_depth	Profundidade máxima das árvores
min_samples_leaf	Número mínimo de amostras em folhas
min_samples_split	Número mínimo de amostras por divisão
max_features	Número máximo de atributos
subsample	Fração de amostras a serem usadas para ajustar a árvore
random_state	Semente de aleatoriedade para reprodução

Fonte: Tuleski, 2023.

Na Tabela 5, estão descritas as vantagens e desvantagens de alguns dos hiperparâmetros do modelo *Gradient Boosting*.

Tabela 5 - Vantagens e Desvantagens dos principais hiperparâmetros do *Gradient Boosting*.

Hiperparâmetros	**Descrição**	**Vantagens**	**Desvantagens**
n_estimators	Número de estimadores	Robustez do modelo.	Complexidade computacional.
learning_rate	Taxa de Aprendizado	Generalização do modelo.	Complexidade computacional.
max_depth	Profundidade máxima das árvores	Complexidade modelo e *overfitting*.	Complexidade computacional.
min_samples_split	Número mínimo de amostras por divisão	Divisão excessiva e *overfitting*.	Árvores rasas e menos representativas.
subsample	Fração de amostras	Aleatoriedade e generalização.	Ajuste aos dados.

Fonte: Tuleski, 2023.

Outros hiperparâmetros como o número máximo de recursos (*max_features*), podem controlar o número máximo de características consideradas ao dividir um nó. Limitar esse número pode reduzir a complexidade do modelo e evitar o *overfitting*. Por fim, a semente de aleatoriedade (*Random_state*), define a aleatoriedade para garantir que os resultados sejam reproduzíveis.

A combinação correta de valores desses hiperparâmetros é geralmente realizada por meio de técnicas como *GridSearch* ou otimização Bayesiana. Essas técnicas de otimização serão exploradas nos próximos capítulos.

4.5 SUPPORT VECTOR MACHINE

O método, *Support Vector Machine* (SVM), é um algoritmo de ML supervisionado usado para classificação e regressão. O SVM foi proposto originalmente por Vapnik *et al.* (1995) e é um dos

algoritmos mais populares em *machine learning*, especialmente em problemas de classificação binária.

De acordo com a definição de Carreira *et al.* (2012), o SVM busca encontrar um hiperplano que separe as classes de dados de entrada no espaço de características. O hiperplano é escolhido de forma que a distância entre o hiperplano e os pontos de cada classe mais próximos, chamados vetores de suporte, seja máxima. Isso é conhecido como margem máxima. O SVM usa esses vetores de suporte para fazer a classificação dos novos dados de entrada.

O SVM funciona bem em problemas de alta dimensionalidade, ou seja, quando há muitas características no conjunto de dados, e é resistente a dados ruidosos. O SVM também tem uma boa capacidade de generalização, o que significa que é capaz de classificar corretamente novos dados de entrada que nunca foram vistos antes.

A matemática por trás do SVM é bastante complexa, mas a ideia básica é simples. Dado um conjunto de treinamento N pares de entrada/saída, onde cada entrada é um vetor x_i de n dimensões e cada saída é uma classe y_i $\{-1, 1\}$, o SVM busca encontrar o hiperplano que maximize a margem entre duas classes. O modelo matemático utilizado para a equação geral do hiperplano do modelo SVM pode ser descrito de acordo com,

$$w^T x + b = 0 \tag{18}$$

onde w é o vetor normal ao hiperplano e b é o termo de deslocamento. A distância do hiperplano para o ponto mais próximo de cada classe é dada por,

$$\gamma = \frac{y(w^T x + b)}{\|w\|} \tag{19}$$

onde $\|w\|$ é a norma do vetor w. O SVM busca maximizar a margem entre as duas classes, o que é equivalente a minimizar a norma do vetor w. Portanto, o problema de otimização é definido por

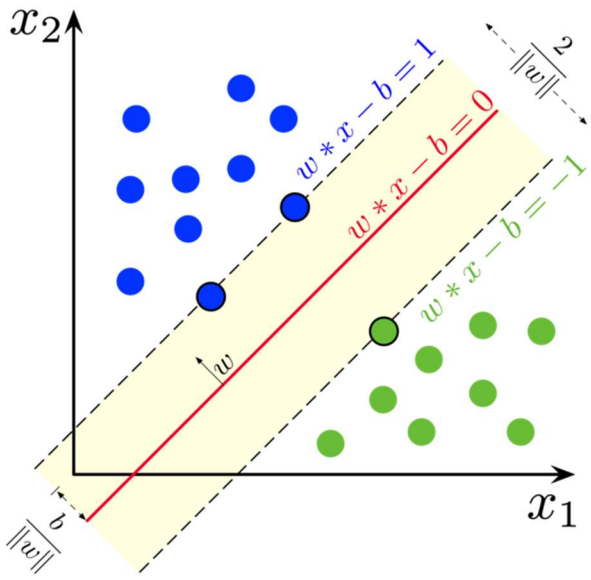
$$min \frac{1}{2} \|w\|^2 \tag{20}$$

Sujeito a,

$$y_i(w^T x_i + b) \geq 1 \text{ para todo } i = 1,2,\ldots,N. \quad (21)$$

conhecido como a formulação primal do SVM. Em problemas de classificação não-linear, o SVM usa o truque de *kernel* para transformar o espaço de entrada em um espaço de características de dimensão superior. O SVM encontra um hiperplano nesse espaço de características que separe as classes. Alguns exemplos de *kernel* comuns são o *kernel* polinomial e o *kernel* RBF (*Radial Basis Function*). Na Figura 18, podemos correlacionar como o hiperplano é utilizado para a separação de classes.

Figura 18 - Ilustração do método SVM



Fonte: Autor Desconhecido, 2022.

Os principais hiperparâmetros do modelo SVM estão demonstrados na Tabela 6.

Tabela 6 - Principais hiperparâmetros da SVM.

Hiperparâmetros	**Descrição**
C	Parâmetro de Regularização
kernel	Função de Kernel
degree	Grau do *Kernel* Polinomial
gamma	Coeficiente do *Kernel*

coef0	Termo independente no *Kernel*
shrinking	Ativar ou desativar encolhimento
probability	Ativar ou desativar cálculo de probabilidade
tol	Tolerância para parada
class_weight	Peso das classes
verbose	Ativar saída detalhada
random_state	Semente de aleatoriedade para reprodução

Fonte: Tuleski, 2023.

O parâmetro de regularização (C), controla o equilíbrio entre a maximização da margem de separação entre as classes e a minimização do erro de classificação no conjunto de treinamento. Um valor menor de C torna a margem mais larga, tornando o modelo mais tolerante a erros de treinamento, mas p possivelmente com uma capacidade de generalização reduzida. Um valor maior de C torna a margem mais estreita, ajustando o modelo mais rigorosamente aos dados de treinamento, mas pode levar ao *overfitting* se for muito alto.

A função *kernel*, determina como o SVM mapeia os dados originais para um espaço dimensional superior, em que é mais fácil encontrar um hiperplano de separação linear. Os kernel mais comuns são o *kernel* linear, RBF (Radial Basis Function), e polinomial. A escolha do kernel depende da natureza dos dados e das relações entre as classes. O kernel RBF é flexível e funciona bem em muitos cenários, mas requer ajuste cuidadoso do hiperparâmetros *gamma*.

O grau do kernel (*degree*), é relevante apenas quando o kernel é polinomial. Ele controla o grau do polinômio usado para a transformação dos dados. Um grau mais alto pode capturar relações mais complexas, mas também aumenta o risco de *overfitting*.

O coeficiente de kernel (*gamma*), é relevante para os kernels RBF e polinomial. Ele controla a influência de um único exemplo de treinamento e afeta a flexibilidade do modelo. Um valor menor de *gamma* suaviza a função de decisão, enquanto u valor maior a torna mais sensível aos pontos de treinamento, o que pode levar ao *overfitting*.

O termo independente no kernel (coef0), é relevante para kernels polinomiais e sigmoidais. Controla o termo independente na função de kernel e pode ser usado para ajustar a forma da função de decisão.

O hiperparâmetros, ativação de encolhimento (*shrinking*), é uma técnica que acelera o treinamento do SVM, desativando o processo de seleção de vetores de suporte menos relevantes durante o treinamento. Isso pode melhorar a eficiência do modelo, especialmente em conjuntos de dados grandes.

E por fim, o cálculo de probabilidade (*probability*), permite que o SVM estime as probabilidades de pertencimento de uma observação a cada classe. Isso é útil em problemas de classificação em que a probabilidade é mais informativa do que a simples previsão de classe.

Em resumo, o ajuste correto desses hiperparâmetros é essencial para obter um modelo SVM com bom desempenho de classificação ou regressão.

4.6 MULTILAYER PERCEPTRON MULTICAMADAS

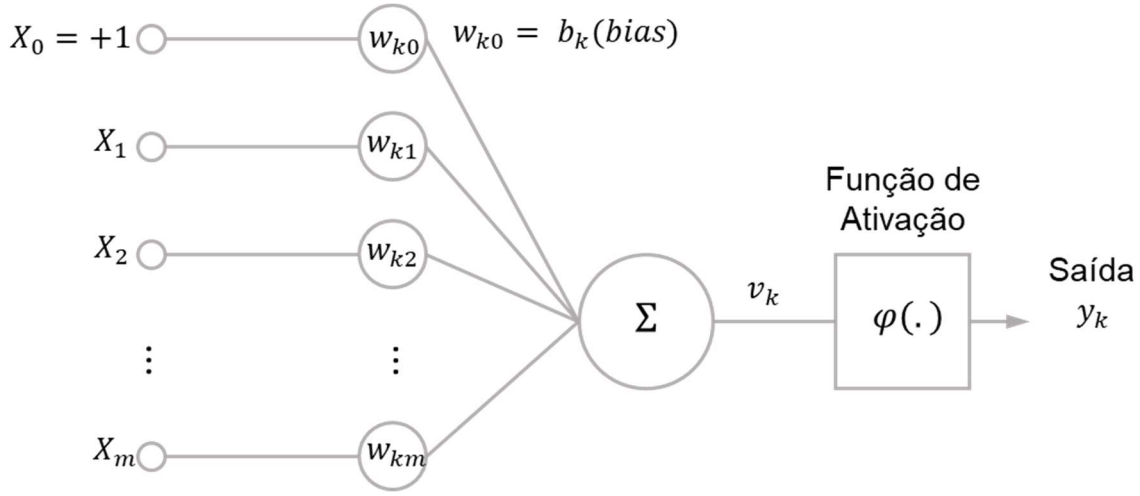
De acordo com Haykin (2009), o MLP é um modelo de rede neural artificial *feedforward*, comumente utilizado para classificação e regressão. A rede neural MLP é composta por uma ou mais camadas ocultas de neurônios que transformam os dados de entrada em um conjunto de saídas. Cada neurônio em uma camada está conectado a todos os neurônios da camada anterior e da camada seguinte.

A MLP é capaz de modelar funções não-lineares complexas, o que torna uma das técnicas de ML mais poderosas. Ele é especialmente útil para problemas de classificação com múltiplas classes e problemas de regressão com múltiplas saídas, conforme ilustrado na Figura .

Bishop (1995) demonstrou que a matemática associada a rede neural MLP é baseada no modelo de neurônio artificial, que é uma função matemática que recebe um vetor de entrada x e produz uma saída y . Cada neurônio em uma camada oculta calcula uma combinação linear dos seus inputs e adiciona um *bias*, em seguida, aplica uma função de ativação não-linear como a função *sigmoide* ou função *ReLu*. A função de ativação é responsável por introduzir a não-linearidade no modelo.

A saída de cada neurônio é então passada para os neurônios na próxima camada oculta ou na camada de saída. Na camada de saída, a MLP geralmente usa uma função de ativação diferente, dependendo do tipo de problema. Para problemas de classificação binária, a função de ativação mais comum é a função *sigmoide*, enquanto para problemas de classificação com múltiplas classes, a função *softmax* é mais apropriada.

Figura 21 - Ilustração da rede neural MLP.



Fonte: Ferreira, 2007.

A MLP é treinada usando o algoritmo de BP, que é um método de otimização baseado no gradiente descendente. O objetivo do treinamento é ajustar os pesos das conexões entre os neurônios para minimizar a função de custo, que é uma medida da diferença entre as saídas previstas pelo modelo e as saídas reais.

O cálculo da saída de um neurônio em uma camada oculta é descrito por,

$$z = \sum (x_i . w_i) + b \tag{22}$$

onde x_i é o valor de entrada para o neurônio i, w_i é o peso da conexão entre o neurônio i e o neurônio atual, e b é o *bias* do neurônio atual. A saída do neurônio é definida por,

$$y = f(z) \tag{53}$$

onde $f(z)$ é a função de ativação não-linear. A função de ativação não-linear é essencial no processo de aprendizado de redes neurais artificias, pois permite que a rede modele relações complexas entre os dados de entrada e saída. Cada função de ativação tem suas próprias vantagens e desvantagens, dependendo do problema em questão e da arquitetura da rede neural utilizada. A escolha da função de ativação correta pode fazer a diferença entre uma rede neural bem-sucedida e uma que não consegue aprender efetivamente.

Os hiperparâmetros em um modelo MLP desempenham um papel crucial na rede neural. A Tabela 7 ilustra os principais hiperparâmetros da rede neural MLP.

Tabela 7 - Principais hiperparâmetros da MLP.

Hiperparâmetros	**Descrição**
hidden_layer_size	Número de neurônios em cada camada oculta
activation	Função de ativação
solver	Otimizador
alpha	Parâmetro de Regularização L2
batch_size	Tamanho do lote
learning_rate	Taxa de aprendizado
learning_rate_init	Taxa de aprendizado inicial
tol	Tolerância para parada
max_iter	Número máximo de iterações
shuffle	Embaralhar amostras durante o treinamento
random_state	Semente de aleatoriedade para reprodução

Fonte: Tuleski, 2023.

O hiperparâmetro, número de neurônios na cada camada oculta (*Hidden_layer_sizes*) define a arquitetura da rede neural, especificando o número de neurônios em cada camada oculta. A escolha do tamanho e do número de camadas ocultas afeta a capacidade da rede de aprender representações complexas dos dados. Mais neurônios ou camadas ocultas podem permitir que o modelo capture padrões mais complexos, mas também podem aumentar o risco de *overfitting*.

A função de ativação (*activation*), é aplicada em cada neurônio para introduzir não linearidade na rede. Opções comuns incluem ReLu (*Rectified Linear Unit*), Tanh (Tagente Hiperbólica) e *Logistic* (Sigmoide). A escolha da função de ativação afeta como a rede aprende e modela relações não lineares nos dados.

O hiperparâmetros otimizador (*solver*) é o algoritmo que ajusta os pesos da rede durante o treinamento. Opções comuns incluem *adam*, *sgd* (*Stochastic Gradient Descent*) e *lbfgs* (*Limited-*

memory Broyden-Fletcher-Goldfarb-Shanno). A escolha do otimizador pode afetar a velocidade de treinamento e a convergência para uma solução ótima.

O termo de regularização L2 ou hiperparâmetro *alpha* (*Alpha*) controla a penalização de pesos grandes na rede. Um valor maior de alpha aumenta a regularização, ajudando a evitar o *overfitting*, mas também pode diminuir a capacidade de ajuste do modelo aos dados de treinamento.

O tamanho do lote (*batch_size*) determina quantos exemplos de treinamento são utilizados em cada etapa de atualização dos pesos durante o treinamento. Valores típicos incluem *auto*, que escolhe automaticamente um tamanho de loto com base no tamanho do conjunto de treinamento, ou um valor inteiro. O tamanho do lote pode afetar a estabilidade do treinamento e a velocidade de convergência.

O hiperparâmetro taxa de aprendizado (*learning_rate*) determina o tamanho dos passos que a rede dá durante a otimização dos pesos. Valores diferentes afetam a taxa de convergências do treinamento e a estabilidade. A escolha adequada da taxa de aprendizado é fundamental para garantir que o treinamento da rede seja bem-sucedido.

Por fim, o número máximo de iterações (*max_iter*) define o número máximo de iterações (épocas) permitidas durante o treinamento. É importante para evitar treinamentos excessivamente longos que podem não convergir. A quantidade apropriada de iterações depende da convergência do modelo.

4.7 CONVOLUTIONAL NEURAL NETWORK

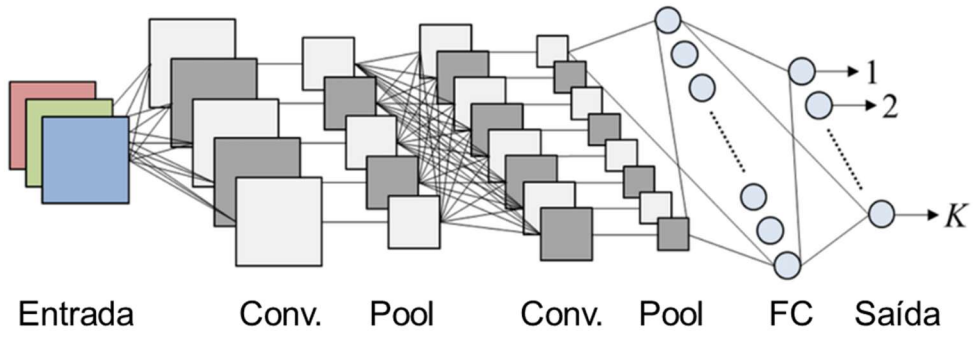
Conforme Dumoulin *et al.* (2018), as CNNs são um tipo de rede neural artificial que são especialmente projetadas para processar dados de alta dimensionalidade, como imagens e sinais de áudio. As CNNs são compostas por camadas convolucionais e de *pooling*, seguidas por camadas totalmente conectadas que realizam a tarefa de classificação ou regressão.

A camada convolucional é responsável por extrair características de baixo nível a partir das imagens de entrada. Ela consiste em um conjunto de filtros, conhecidos como *kernels*, que deslizam pela imagem de entrada para realizar a convolução e produzir um mapa de características. Cada filtro é ajustado durante o treinamento para capturar padrões específicos, como bordas, curvas ou texturas.

A camada de *pooling* é usada para reduzir a dimensão do mapa de características. Mantendo as características mais importantes. A técnica de *pooling* mais comum é a de máximo, que retorna o valor máximo de cada região do mapa de características. Isso reduz a quantidade de dados que precisa ser processada nas camadas posteriores e ajuda a evitar o *overfitting*.

As camadas totalmente conectadas são responsáveis por fazer a classificação ou regressão final com base nos recursos extraídos pelas camadas convolucionais e de *pooling*. Cada neurônio em uma camada totalmente conectada é conectado a todos os neurônios na camada anterior, e a saída final é obtida aplicando uma função de ativação na última camada, conforme a Figura 19.

Figura 19 - Ilustração do método Convolucional Neural Network



Fonte: Akinori, 2017.

Matematicamente a operação de convolução pode ser descrita como,

$$y(i,j) = f\left(\sum (x(i+m, j+n).w[m.n]) + b\right) \tag{64}$$

onde x é a imagem de entrada, w é o filtro convolucional, b é o *bias*, e f é a função de ativação.

A CNN é amplamente utilizada em tarefas de processamento de imagens, como reconhecimento de objetos, classificação de imagens e detecção de objetos. Alguns exemplos de aplicações da CNN incluem a classificação de imagens de satélites, a detecção de timores em imagens médicas e a identificação de objetos em cenas de vigilância. Na Tabela 8, levantou -se os principais hiperparâmetros da CNN.

Tabela 8 - Principais hiperparâmetros da CNN.

Hiperparâmetros	**Descrição**
input_shape	Formato da entrada
num_classes	Número de classes de saída

learning_rate	Taxa de aprendizado
batch_size	Tamanho do lote
num_epochs	Número de Épocas
dropout_rate	Taxa de dropout para regularização
filters	Número de filtros em cada camada convolucional
kernel_sizes	Tamanho dos filtros em cada camada convolucional
pool_sizes	Tamanho das camadas de *pooling*
dense_units	Número de unidas nas camadas totalmente conectas

Fonte: Tuleski ,2023.

O número de camadas e filtros convolutivos, a arquitetura da CNN é determinada pelo número de camadas convolutivas e o número de filtros em cada camada. O número de camadas convolutivas afeta a profundidade da rede, permitindo que ela aprenda características complexas em diferentes níveis de abstração. O número de filtros em cada camada controla a quantidade de características especiais que a rede pode extrair. Mais filtros podem capturar detalhes, mas também aumentam a complexidade computacional.

O tamanho do filtro convolutivo define a região espacial que a convolução analisa. Filtros menores são úteis para destacar detalhes finos, enquanto filtros maiores podem capturar características mais amplas. A escolha do tamanho do filtro depende da natureza das características que se deseja extrair.

O *stride* controla o passo com que o filtro convolutivo percorre a entrada, afetando o tamanho da saída. Um *strider* maior reduz o tamanho da saída e pode levar à perda de informações espaciais. O *padding* pode ser *valid* (sem preenchimento) ou *same* (preenchimento para manter o tamanho da saída igual ao da entrada). O *padding* afeta o tamanho da saída e a preservação das bordas da imagem.

A função de ativação, como ReLU, Sigmoide ou Tanh, é aplicada após cada operação convolutiva para introduzir não linearidade na rede. A escolha da função de ativação afeta a capacidade da rede de aprender representações complexas e a velocidade de convergência.

As camadas de *pooling* reduzem a dimensionalidade da saída, agrupando informações em regiões menores. O tamanho do *pooling* controla a quantidade de redução. Um *pooling* maior reduz mais drasticamente a dimensionalidade, mas também perde informações detalhadas.

O *dropout* é uma técnica de regularização que ajuda a prevenir o *overfitting*, deligando aleatoriamente um percentual de neurônios durante o treinamento. A taxa de *dropout* controla a fração de unidades desligadas. Um valor tipicamente utilizado é 0.5, o que significa que metade das unidades é desligada durante o treinamento.

A taxa de aprendizado controla o tamanho dos passos que o otimizado dá ao ajustar os pesos da rede durante o treinamento. Um valor muito alto pode fazer com que o treinamento seja instável, enquanto um valor muito baixo pode tornar o treinamento lento ou levar à convergência prematura. O algoritmo de otimização como *Adam*, SGD ou RMSprop, determina como os pesos da rede são atualizados durante o treinamento. Algoritmos diferentes têm comportamentos diferentes, afetando a velocidade de treinamento e a convergência.

4.8 LONG SHORT-TERM MEMORY

As redes neurais *Long Short-Term Memory* (LSTM) representam uma classe de arquiteturas de redes neurais recorrentes (RNN) que têm desempenhado um papel revolucionário no campo do aprendizado de máquina e do processamento de sequências. Hochreiter e Schmidhuber (1997) introduziram o termo *Long Short-Term Memory* como uma solução para os desafios enfrentados pelas RNNs convencionais no aprendizado de sequências longas. A principal inovação por trás das LSTMs é o uso de células de memória com um mecanismo de portões para controlar o fluxo de informações. Essas células permitem que a LSTM mantenha e manipule informações relevantes ao longo de sequências temporais.

O mecanismo de portões das LSTMs envolve operações matemáticas fundamentais que controlam o fluxo de informações dentro da célula da rede. A essência desse mecanismo é composta por portão de esquecimento (*Forget Gate*), portão de entrada (*Input Gate*) e portão de saída (*Output Gate*). O *Forget Gate* utiliza uma função sigmoide para decidir quais informações da memória anterior devem ser mantidas ou esquecidas. A equação matemática para o cálculo do valor de esquecimento f em cada elemento da célula é dada por,

$$f_t = \sigma(W_f . [h_{t-1}, x_t + b_f] \tag{25}$$

Em que, f_t é o vetor de esquecimento, σ é a função sigmoide, W_f é a matriz de pesos do portão de esquecimento, h_{t-1} representa o resultado oculto anterior e x_t é a entrada atual.

O *Input Gate* controla quais informações serão adicionadas à memória atual. Ele também utiliza uma função sigmoide para calcular um vetor de ativação i que decide quais valores serão atualizados. A equação para o cálculo do vetor de entrada i é dada por,

$$i_t = \sigma(W_i.[h_{t-1}, x_t + b_i] \tag{26}$$

O *Output Gate* controla quais informações da memória atual serão usadas para calcular a saída da célula. Ele envolve uma função sigmoide para gerar um vetor de saída o que influencia a saída final. A equação matemática para o cálculo do vetor de saída o é definido por,

$$o_t = \sigma(W_o.[h_{t-1}, x_t + b_o] \tag{27}$$

As LSTMs encontraram aplicação em diversas áreas, por exemplo, Sutskever *et al.* (2014) aplicaram rede neural para o processamento de linguagem natural (NLP). Goodfellow *et al.* (2016) destacaram-se aplicando a arquitetura para a previsão de séries temporais em áreas como finanças. Para visão computacional, Karpathy *et al.* (2015) implementaram LSTM para reconhecimento de objetos e segmentação de vídeo.

As LSTM representam um marco no campo do aprendizado de máquina e do processamento de sequências, tanto do ponto de vista teórico quanto prático. Sua capacidade de lidar com dependências de longo prazo e de manter informações relevantes ao longo do tempo as torna essenciais para uma ampla variedade de aplicações. Á medida que continuamos a explorar e aprimorar as LSTMs, seu impacto pode ser estendido para aplicações como diagnóstico de falhas em motores de ignição por compressão por meio do sinal de áudio.

CAPÍTULO 5
MATERIAIS

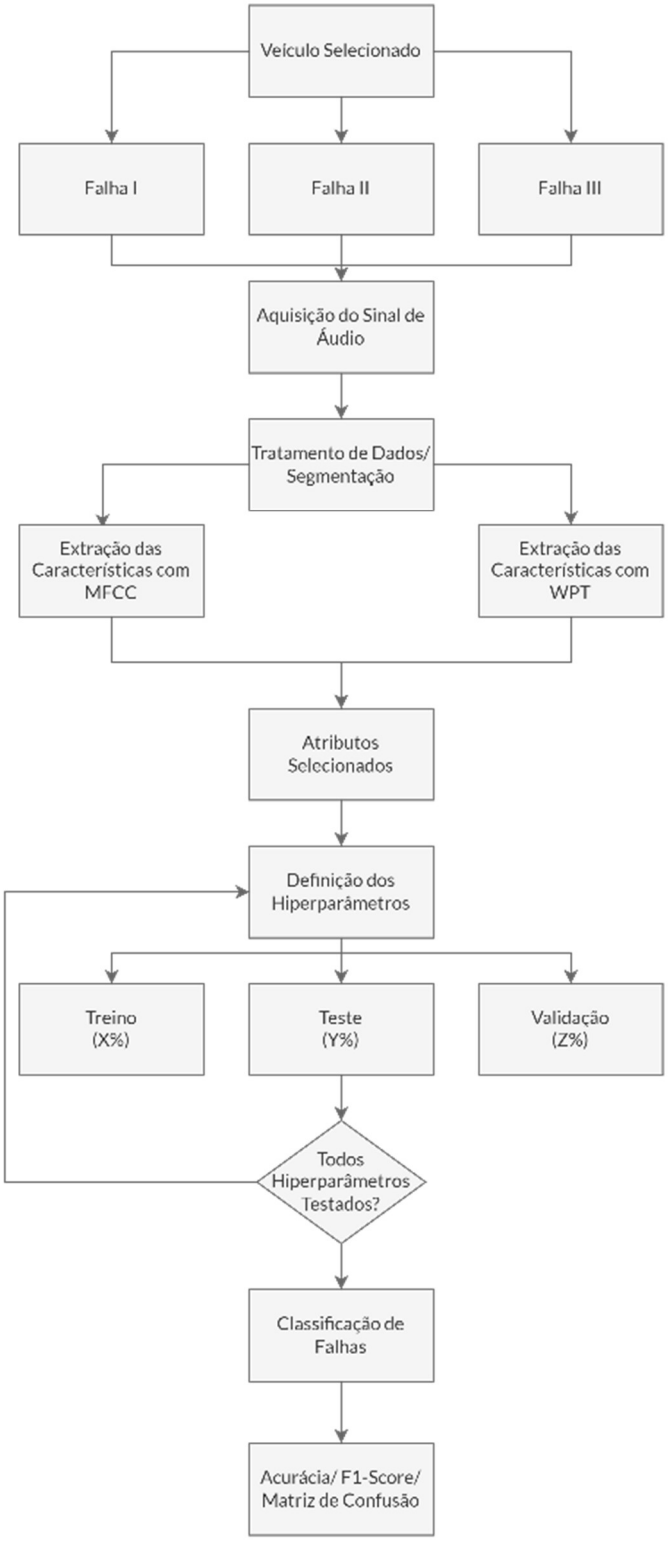
O presente trabalho utiliza a Transformada *Wavelet* para investigar falhas em componentes de um motor de ignição por compressão (Ayati, 2020). Características baseadas em estatísticas de uma Transformada *Wavelet* foram extraídas de gravações de áudios de motores e utilizadas para treinar classificadores. Nesse estudo, seis diferentes classificadores com suas configurações foram aplicados que são: RF, *Gradient Boosting*, SVM, MLP, CNN-LSTM e uma MFCC-CNN.

A Figura 20 ilustra o fluxograma proposto para a execução deste estudo que consiste em seis etapas. Na primeira etapa, o sinal de áudio é coletado para formar a base de dados. Na segunda etapa, os dados são segmentados em conjuntos menores, é um processo crucial em muitas tarefas de análise de dados aplicando modelos de ML. Ao segmentar os dados, o ruído pode ser reduzido e a precisão pode aumentar. Em seguida, a WPT ou MFCC são utilizados como métodos para a extração de características dos sinais segmentados. Na sequência, a seleção de caraterísticas, que é o processo de pré-processamento de dados, que envolve a escolha das características mais relevantes ou importantes para um modelo de ML é executada. O objetivo da seleção de atributos é reduzir a dimensionalidade dos dados e melhorar a precisão e a eficiência dos modelos. Por fim, os modelos de ML são selecionados para treinar e testar a capacidade de um determinado modelo de diagnosticar falhas em veículos que são baseados na informação de entrada, isto é, o sinal de áudio. Então as métricas estatísticas *F1-score* e acurácia foram utilizadas para a acurãcia do diagnóstico correto para as classes definidas.

Em relação as métricas estatísticas escolhidas, o *F1-score*, conforme Sasaki (2007), combina as métricas de precisão e *recall* em uma única medida. A precisão é a proporção de instâncias classificadas como positivas que são realmente positivas, enquanto o *recall* é a proporção de instâncias positivas que são corretamente identificadas. O *F1-score* é calculado a partir da média harmônica da precisão e do *recall*. A média harmônica dá mais maior ponderaçãp às métricas menores. A métrica varia entre 0 e 100%, sendo que 100% indicam um modelo com alta precisão e *recall*. O *F1-score* é particularmente útil quando as classes estão desequilibradas ou quando é importante ter um desempenho tanto na identificação correta dos positivos quanto dos negativos.

Por outro lado, a acurácia, é principalmente utilizada quando as classes estão bem equilibradas. Está métrica define a proporção de instâncias corretamente classificadas em relação ao total de instâncias.

Figura 20 – Fluxograma da proposição para a detecção de falhas.



Fonte: Tuleski, 2023.

5.1 ESTUDO DE CASO

Os motores de ignição por compressão são conhecidos por sua eficiência e robustez. No entanto, como qualquer máquina complexa, esses motores estão sujeitos a várias falhas que podem afetar seu desempenho e durabilidade. Por exemplo, as falhas mecânicas podem incluir problemas no pistão, biela, virabrequim, válvulas, anéis de pistão e outros componentes internos. Essas falhas podem resultar em vibrações anormais, ruídos metálicos ou batidas no motor. Outro exemplo de falha, podem ser problemas nos injetores, como entupimento, vazamento ou mau funcionamento, podem levar a um desempenho inadequado, incluindo falta de potência e emissões alterada. A qualidade do combustível também afeta a eficiência da combustão e, portanto, a assinatura de áudio do motor. A análise da assinatura de áudio desses modos de falha pode ajudar a diagnosticar de forma correta qual é o componente defeituoso.

Para a determinação do tamanho da amostra, ou número de oficinas a serem consultadas para selecionar as principais falhas representativas encontradas em casos reais, métodos estatísticos foram utilizados para o cálculo do tamanho da amostra. Nesse contexto, foi considerado conhecido o tamanho da população, portanto utilizou-se a equação de Neyman (1934), que leva em consideração o tamanho da população N, o nível de confiança Z, e o erro amostra desejado e. Matematicamente, a equação de Neyman é expressa por,

$$n = \frac{\frac{Z^2.p.(1-p)}{e^2}}{1 + \frac{[Z^2.p.(1-p)]}{e^2.N}} \tag{28}$$

Em que p representa a proporção esperada da característica de interesse na população. Além disso, é essencial considerar a variabilidade esperada na população ao determinar o valor de p (Thompson, 2012).

Substituindo os valores da Tabela 9 na equação anterior, estimou-se que a amostra da população, com arredondamento para cima, de $n = 50$ oficinas.

Tabela 9 - Parâmetros para determinação do tamanho amostral.

Parâmetros	***Descrição***	***Valor***
N	Tamanho da População	100 oficinas
z	Escore z	1,96 (95%)
e	Margem de Erro	1%

p	Desvio Padrão	50%

Fonte: Tuleski, 2023.

Com base em uma pesquisa realizada em campo com mais de 50 oficinas mecânicas da rede credenciada Bosch, especializadas em manutenção de veículos com motores de ignição por compressão, foram levantadas as principais falhas diagnosticadas recorrentemente. A Figura 21 apresenta a relação entre o valor para o cliente e a facilidade para a Bosch fornecer um diagnóstico preciso, considerando a *expertise* da empresa. O resultado daquela pesquisa mostrou que o principal interesse de diagnóstico é relacionado a falhas que ocorrem no injetor. Portanto, esse estudo traz como foco a identificação dessas falhas.

A coleta de dados em campo para a realização desse estudo se apresentou um desafio, principalmente relacionado a disponibilidade de veículos com falhas e ao risco associado de coletar dados desses veículos. Assim, foi proposto coletar sinais de áudios de veículos em três condições diferentes e que não apresentam riscos para a saúde do veículo. Essas condições estão expostas na Tabela 10 e foram as três classes escolhidas para a realização dos diagnósticos neste estudo.

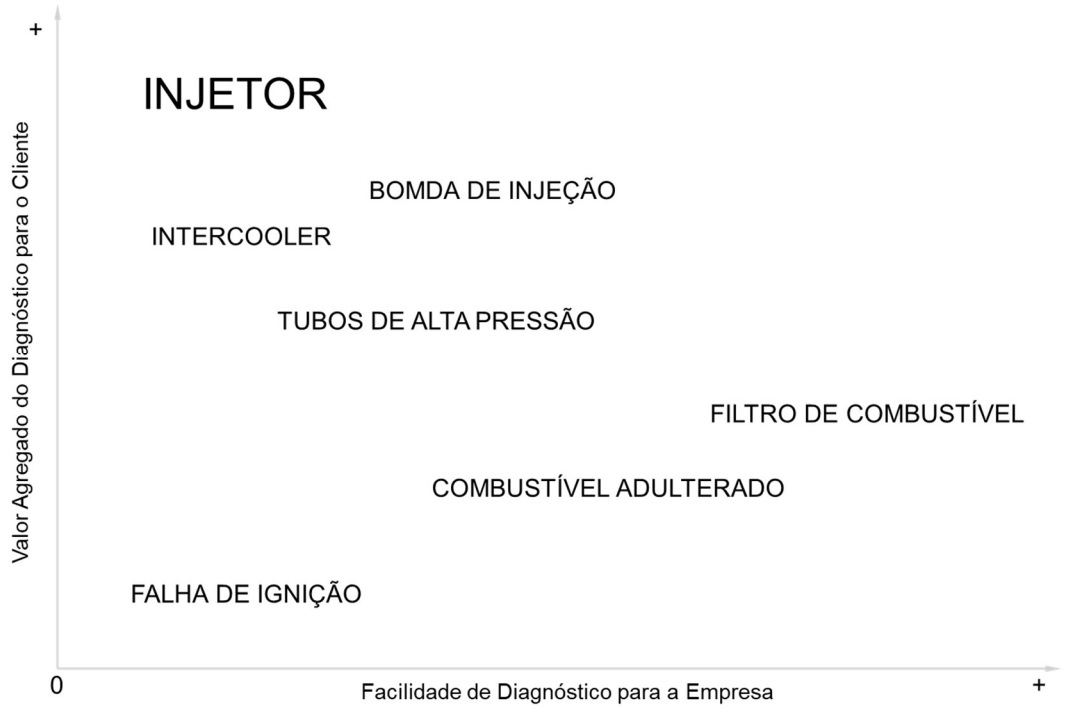
Os veículos escolhidos para a coleta de dados, como pode-se observar na Tabela 11, foram selecionados com base em vários critérios, sendo a disponibilidade o fator relevante. Além disso, outros critérios foram fator decisivo da escolha, como por exemplo, diferentes tipos de aplicação e motor. É importante ressaltar que o estudo foca em apenas veículos de aplicações com motores de ignição por compressão. Para o estudo, foram selecionados dois veículos de cada modelo, variando condição física, porém com mesmo ano de fabricação.

Tabela 10 - Definição das Classes

Classes	***Descrição de falhas***
Classe 1	Sem Anomalias
Classe 2	Falha no Injetor
Classe 3	Falha na Mangueira de Admissão

Fonte: Tuleski, 2023.

Figura 21 - Definição do foco da pesquisa.



Fonte: Tuleski, 2023.

Tabela 11 - Veículos utilizados para a coleta de dados e amostras.

Identificação	**Veículo automotivo**
	Ford Ranger 3.0 Powerstroke
	Fiat Toro 2.0 Turbo
	Iveco Daily 65-170

Fonte: Tuleski, 2023.

Considerando que o estudo busca viabilizar a reprodução do experimento por outros usuários, optou-se pelo uso do dispositivo *Iphone XR* para coleta dos dados, que possui um microfone direcional localizado na parte inferior do aparelho, conforme ilustra a Figura 22, modelo *MEMS*, com frequência de resposta de 20Hz a 20kHz, para registro dos áudios. Todos os áudios foram coletados por meio desse microfone.

Figura 22 - Microfone Localizado na parte inferior do dispositivo Iphone XR.

Fonte: Tuleski, 2023.

Com base no estudo de campo realizado para compreender a viabilidade prática da utilização da aplicação por um usuário final, os áudios foram gravados com um telefone móvel acessando diversos veículos com rotação de marcha lenta – aproximadamente entre 800 e 1000 RPM, de acordo com os dados descritos na Tabela 12.

As rotações escolhidas para captura do áudio dos motores foram influenciadas por três fatores.

i) O primeiro fator é relacionado a facilidade para o usuário conseguir capturar o áudio do veículo mesmo que precise sair do veículo, portanto a marcha lenta, permite que essa condição seja cumprida pois o usuário consegue deixar o veículo acionado em marcha lenta em condição segura sem precisar estar dentro do veículo.

ii) O segundo fator é relacionado a um dos impedimentos causados pela falha do injetor, a central de controle eletrônica (ECU) do veículo, quando detecta a perda de potência do motor, limita o número de rotações para garantir que não haja danos.

iii) E por fim, ambas rotações foram escolhidas para avaliar e compreender o desempenho dos modelos enquanto as suas respectivas assinaturas de áudio em situações opostas.

Tabela 12 - Variação dos dados coletados dos veículos automotivos.

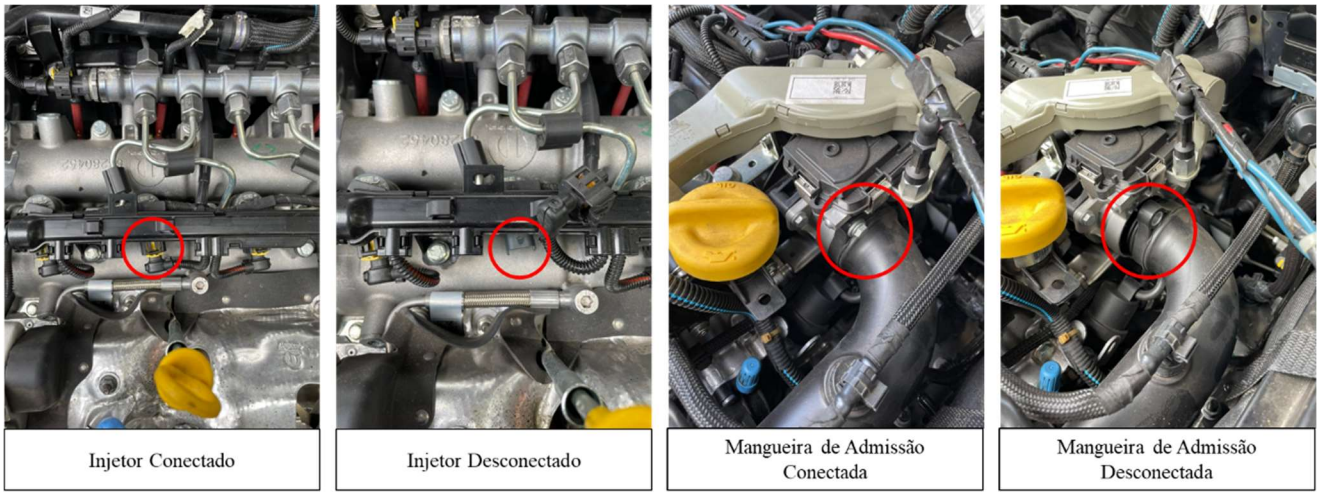
Veículos	**Rotações [RPM]**	**Falhas**
Ford Ranger	[Marcha Lenta, 2700]	Sem Falha, Injetor e Mangueira

Fiat Toro	[Marcha Lenta, 2400]	Sem Falha, Injetor e Mangueira
Iveco Daily	[Marcha Lenta, 3000]	Sem Falha, Injetor e Mangueira

Fonte: Tuleski, 2023.

Portanto, é importante notar que, o equipamento utilizado para a aquisição de dados não é dedicado ou específico para o propósito da aplicação. Os dados coletados contêm ruído ambiente, o que torna a classificação um desafio ainda maior. Para os veículos selecionados, três condições diferentes foram consideradas, como: sem anomalias, um dos cilindros com injetor desligado e mangueira de admissão desconectada conforme ilustrado na Figura 23.

Figura 23 - Exemplificação das falhas simuladas nos veículos utilizados.



Fonte: Tuleski, 2023.

A Figura mostra a posição de coleta dos áudios, que foram obtidas em ambiente externo, em três posições diferentes, duas externas, em frente ao veículo e ao lado do condutor, e outra interna, na posição do condutor do veículo.

As gravações foram realizadas com a taxa de amostragem de 22,05 kHz com arquivos no formato *m4a,* e posteriormente convertidas para o formato *wav.* Três modelos diferentes de veículos foram selecionados para a coleta de dados, duas camionetes e um caminhão. O total de áudios gravados estão disponíveis na

Tabela *13*.

Desde a configuração experimental, diferentes faixas são utilizadas para treinamento, validação e teste, a fim de evitar resultados tendenciosos. Além disso, para balancear as classes, 550 segundos de cada áudio foi segmentado com o tamanho de 1s e os dados foram normalizados.

Figura 27 - Posição de coleta dos dados nos veículos.

Fonte: Tuleski, 2023.

Tabela 13 - Duração dos áudios coletados em relação ao veículo e condição de falha.

Veículos	**Duração Total do Áudio por Condição de Falha**		
	Sem anomalias	Injetor desconectado	Mangueira de admissão desconectada
Ford Ranger I	10’02’’	10’05’’	10’01’’
Ford Ranger II	10’13’’	10’08’’	10’06’’
Fiat Toro I	10’02’’	10’04’’	10’01’’
Fiat Toro II	10’17’’	10’22’’	10’11’’
Iveco Daily I	10’03’’	10’59’’	10’56’’
Iveco Daily II	10’09’’	10’27’’	10’06’’

Fonte: Tuleski, 2023.

5.2 EXTRAÇÃO DOS ATRIBUTOS

Em consideração aos parâmetros da Transformada *Wavelet* utilizados para extrair os recursos para cada veículo analisado, foi considerado ao sexto nível de decomposição, com a geração de 2^6 pacotes de nós *Wavelet*, e db20 de famílias *Daubechies* como *Wavelet* mãe.

Em seguida, as características baseadas em estatísticas são extraídas de cada nó de pacote *Wavelet*. Após uma série de experimentos, encontrou-se que os recursos baseados em estatística que melhor funcionam para cada veículo sob análise pode variar, tendo assim sido necessário conduzir uma análise para cada veículo. Esse resultado para cada recurso extraído está disponível na Tabela 14.

Tabela 14 - Características/Recursos extraídos baseados em estatística para cada veículo.

Veículos	**Característica/Recurso Extraído**
Ford Ranger	Assimetria, média, comprimento de onda, contagem cruzada média, valor eficaz.
Fiat Toro	Assimetria, média, comprimento de onda, contagem cruzada média, valor eficaz.
Iveco Daily	Assimetria, média, comprimento de onda, contagem cruzada média, valor eficaz.

Fonte: Tuleski, 2023.

Após o procedimento de extração de características estatísticas, a seleção foi feita com base no método de filtro de matriz de correlação, por exemplo, as estatísticas mais relevantes extraídas por nó de pacote *Wavelet* são selecionadas pela eliminação de caraterísticas não desejáveis, como a com baixa correlação com a saída em relação as outras. Esse procedimento reduz o tamanho da entrada dos classificadores na ordem da centena para a dezena. O valor dos filtros para cada veículo analisado pode ser visto na Tabela 15.

Tabela 15 - Valores dos filtros de correlação para cada veículo.

Veículos	**Filtro de Entrada**	**Outros Filtros**
Ford Ranger	<10%	>99%
Fiat Toro	<10%	>95%
Iveco Daily	<30%	>99%

Fonte: O Autor, 2023.

Os hiperparâmetros relacionados à extração de MFCC estão exemplificados na Tabela 16. Os parâmetros do MFCC não são otimizados da mesma maneira que os hiperparâmetros de um modelo

de ML. Em vez disso, esses hiperparâmetros são definidos com base no conhecimento do domínio e na natureza do sinal de áudio que será tratado.

Tabela 16 - Hiperparâmetros utilizados para a extração de características com MFCC.

Hiperparâmetros	**Descrição**	**Faixa de Valores**
sample_rate	Taxa de amostragem do áudio	22.050
frame_size	Tamanho da janela	2.048
hop_size	Tamanho do passo	256
n_mfcc	Número de coeficientes	10
n_fft	Tamanho da FFT	2.048
hop_length	Tamanho do passo FFT	*constant*
n_mels	Número de filtros Mel	128
Window	Função de janelamento	*hamming*
preemphasis	Coeficiente de pré-ênfase	0,97

Fonte: Tuleski, 2023.

5.3 OTIMIZAÇÃO DOS HIPERPARÂMETROS

Após experimentos preliminares, os hiperparâmetros para cada modelo de ML foram otimizados utilizando a técnica *GridSearch*. As Tabelas 17 a 22 descrevem os hiperparâmetros utilizados para cada modelo de ML que são RF, *Gradient Boosting*, SVM, MLP, CNN, CNN-LSTM, respectivamente. Em todos os modelos foi utilizada a validação cruzada e o número de *folds* foi definido como $k = 5$.

Tabela 17 – Intervalos dos hiperparâmetros para o modelo RF.

Hiperparâmetros	**Descrição**	**Faixa de Valores**
n_estimators	Número de estimadores	[200, 400, 800, 1000]
max_depth	Profundidade máxima das árvores	[*None*, 10, 20, 30]
min_sample_split	Número mínimo de amostras por divisão	[2, 5, 10]

min_sample_leaf	Número mínimo de amostras por folha	[1, 2, 4]
max_features	Número máximo de atributos	*sqrt*
bootstrap	*Bootstrap sampling*	*True*
criterion	Critério de divisão	*Entropy*

Fonte: Tuleski, 2023.

Tabela 18 - Intervalos dos hiperparâmetros para o modelo *Gradient Boosting*.

Hiperparâmetros	**Descrição**	**Faixa de Valores**
n_estimators	Número de estimadores (árvores)	[200, 400, 800, 1000]
learning_rate	Taxa de aprendizado	[0,01, 0,1, 0,2]
max_depth	Profundidade máxima das árvores	[*None*, 10, 20, 30]
min_samples_split	Número mínimo de amostras por divisão	[2, 5, 10]
min_samples_leaf	Número mínimo de amostras por folha	[1, 2, 4]
Max_features	*Número máximo de atributos*	*None*
Subsample	Fração das amostras	1
Bins	Intervalos	255

Fonte: Tuleski, 2023.

Tabela 19 - Intervalos dos hiperparâmetros para o modelo SVM.

Hiperparâmetros	**Descrição**	**Faixa de Valores**
C	Parâmetro de regularização	[0.1, 1, 5, 10]
kernel	Função Kernel	[*linear, rbf, poly*]
degree	Grau do Kernel Polinomial	[2, 3, 4]
gamma	Coeficiente Kernel	*scale*
coef0	Termo independente no Kernel	0
shrinking	*Encolhimento*	*True*
probability	Cálculo de probabilidade	*False*
tol	Tolerência para parada	1e-3

class_weight	Peso das classes	*Balanced*

Fonte: Tuleski, 2023.

Foi estabelecido um limite de 1.000 iterações para o algoritmo de otimização utilizado na construção do modelo SVM. Esse limite determina o número máximo de vezes que o algoritmo pode iterar para encontrar o hiperplano de separação ótimo para os dados de treinamento. Ao atingir esse limite, o modelo SVM é finalizado mesmo que não tenha encontrado a solução ótima com base no parâmetro da tolerância definida.

Tabela 20 - Intervalos dos hiperparâmetros para o modelo MLP.

Hiperparâmetros	**Descrição**	**Faixa de Valores**
hidden_layer_sizes	Número de neurônios	[(100,), (50, 50), (100, 50)]
activation	Função de ativação	[*ReLU, Tanh*]
solver	Otimizador	[*sgd, adam, lbfgs*]
alpha	Parâmetro de regularização L2	[0,0001, 0,001, 0,01]
batch_size	Tamanho do lote	*Auto*
learning_rate	Taxa de aprendizado	*constant*
learning_rate_init	Taxa de aprendizado inicial	0,001
max_iter	Número máximo de iterações	100
shuffle	Embaralhar amostras	*Balanced*
Tol	Tolerância	1e-3

Fonte: Tuleski, 2023.

A função de perda utilizada foi a divergência de *Kullback-Leibler*, que mede a diferença entre duas distribuições de probabilidade, e o algoritmo de otimização utilizado foi o *Stochastic Gradient Descent* (SGD), que realiza a otimização de forma estocástica, ou seja, a cada passo do algoritmo, uma amostra aleatória do conjunto de treinamento é usada para atualizar os pesos da rede.

Tabela 21 - Intervalos dos hiperparâmetros para o modelo CNN.

Hiperparâmetros	**Descrição**	**Faixa de Valores**

input_shape	Formato da entrada	[128, 128, 3]
num_classes	Número de classes de saída	3
learning_rate	Taxa de aprendizado	[0,001, 0,01]
batch_size	Tamanho do lote	[16, 32, 64]
num_epochs	Número de épocas	[5, 10, 15]
dropout_rate	Taxa de *dropout*	[0,2; 0,5; 0,7]
filters	Número de filtros	[16, 32, 64]
kernel_sizes	Tamanho dos filtros	[(3, 3), (5, 5), (10, 10)]
pool_sizes	Tamanho das camadas de *pooling*	[(2, 2), (3, 3), (5, 5)]
dense_units	Unidades nas camadas conectadas	[128, 64]

Fonte: Tuleski, 2023.

Para treinar a rede, foi utilizado a perda de entropia cruzada como função objetivo e otimizador *Adaptive Momentum* (*Adam*) para atualizar os pesos da rede.

Tabela 22 - Intervalos dos hiperparâmetros para o modelo CNN-LSTM.

Hiperparâmetros	**Descrição**	**Faixa de Valores**
lstm_units	Camada de neurônios LSTM	[32, 64, 128]
lstm_dropout	Taxa de *dropout*	[0,3; 0,5; 0,7]

Fonte: Tuleski, 2023.

5.4 SEPARAÇÃO DA BASE EM TREINAMENTO-TESTE

A estratégia de separação da base de treinamento e teste foi formulada de forma a abordar duas análises diferentes. Inicialmente, os dados foram coletados somente de um veículo de cada modelo, posteriormente, verificou-se que treinar e utilizar para teste os dados coletados de um mesmo veículo poderia gerar viés nos resultados. Uma solução simples encontrada foi expandir a base de dados adicionando pelo menos mais um veículo de cada modelo. Portanto, ambas estratégias foram selecionadas para avaliar a influência da base de dados na capacidade dos modelos em classificar de forma correta evitando tendência.

A primeira estratégia consistiu em escolher aleatoriamente amostras de treinamento de cada segmento de dados, com uma relação treinamento-teste de 75%. Na sequência, a segunda estratégia, consistiu em utilizar os dados de um dos veículos para treinamento e outro para teste. As divisões da base treino-teste estão explicitas na Tabela 23. Para ambas as estratégias, foram calculadas a acurácia e *F1-score* para cada classificador. Além disso, a curva de precisão versus recuperação foi plotada para permitir uma análise visual dos resultados. Essa metodologia permitiu comparar o desempenho dos diferentes classificadores avaliados de forma mais robusta, já que os resultados foram baseados em 100 execuções independentes da metodologia. A escolha aleatória de amostras de treinamento também contribuiu para reduzir o efeito de possíveis vieses na seleção dos dados de treinamento na primeira estratégia.

Tabela 23 - Estratégia da divisão da base treino-teste

Veículos	***Quantidade de Veículos***	***Base Treino-Teste***
Ford Ranger	I Veículo	75%-25% (Estratégia I)
Ford Ranger	II Veículos	50%-50% (Estratégia II)
Fiat Toro	I Veículo	75%-25% (Estratégia I)
Fiat Toro	II Veículos	50%-50% (Estratégia II)
Iveco Daily	I Veículo	75%-25% (Estratégia I)
Iveco Daily	II Veículos	50%-50% (Estratégia II)
Todos	VI Veículos	75%-25%

Fonte: Tuleski, 2023.

CAPÍTULO 6
RESULTADOS E DISCUSSÃO

6.1 INTRODUÇÃO

Conforme comentado anteriormente, seis modelos de Machine Learning foram utilizados para os classificadores, RF*, Gradient Boosting,* MLP*,* SVM, CNN, CNN-LSTM e comparados com a abordagem MFCC-CNN. A escolha da abordagem MFCC-CNN como base comparativa em relação aos outros modelos vem do fato de ser bem estabelecido na literatura na aplicação de vibração, acústica e classificação de sinais de som conforme visto na referência (Henriquez, 2013).

Neste capítulo serão apresentados os principais resultados obtidos com estes modelos os quais são avaliados através do Tempo de execução, Acurácia, *F1-Score* e Matriz de Confusão.

6.2 HIPERPARÂMETROS OTIMIZADOS PARA CADA MODELO

Após os experimentos preliminares, os hiperparâmetros para cada modelo de ML foram obtidos utilizando a técnica *GridSearch*.

Para os modelos RF, *Gradient Boosting,* MLP*,* SVM e CNN-LSTM os hiperparâmetros otimizados obtidos estão descritos nas Tabelas 24, 25, 26, 27, 28 e 29 e 30 respectivamente.

Tabela 24 - Hiperparâmetros otimizados do modelo RF para cada veículo.

Veículos	**treino-teste**	***n_estimators***	***max_depth***	***min_sample_split***	***min_sample_leaf***
Ford Ranger	75%-25%	800	10	2	2
Ford Ranger	50%-50%	400	10	5	1
Fiat Toro	75%-25%	400	10	2	2
Fiat Toro	50%-50%	400	20	5	1
Iveco Daily	75%-25%	800	20	2	2
Iveco Daily	50%-50%	800	10	5	1

Todos	75%-25%	800	20	5	2

Fonte: Tuleski, 2023.

Tabela 25 - Hiperparâmetros otimizados do modelo *Gradient Boosting* para cada veículo.

Veículos	**treino-teste**	***n_estimators***	***Learning_rate***	***max_depth***	***min_sample_split***	***min_sample_leaf***
Ford Ranger	75%-25%	400	0,1	10	5	2
Ford Ranger	50%-50%	400	0,1	20	2	1
Fiat Toro	75%-25%	400	0,1	10	5	2
Fiat Toro	50%-50%	800	0,1	20	2	1
Iveco Daily	75%-25%	400	0,1	10	5	2
Iveco Daily	50%-50%	800	0,1	20	2	1
Todos	75%-25%	400	0,1	10	5	2

Fonte: Tuleski, 2023.

Tabela 26 - Hiperparâmetros otimizados do modelo SVM para cada veículo.

Veículos	**treino-teste**	***C***	***Kernel***	***Degree***
Ford Ranger	75%-25%	1	Poly	4
Ford Ranger	50%-50%	5	Poly	4
Fiat Toro	75%-25%	1	Poly	4
Fiat Toro	50%-50%	5	Poly	4
Iveco Daily	75%-25%	1	Poly	4

Iveco Daily	50%-50%	5	Poly	4
Todos	75%-25%	1	Poly	4

Fonte: Tuleski, 2023.

Tabela 27 - Hiperparâmetros otimizados do modelo MLP para cada veículo.

Veículos	**treino-teste**	***hidden_layer_size***	***activation***	***solver***	***alpha***
Ford Ranger	75%-25%	(50, 50)	ReLU	Adam	0,001
Ford Ranger	50%-50%	(50, 50)	ReLU	Adam	0,001
Fiat Toro	75%-25%	(50, 50)	ReLU	Adam	0,001
Fiat Toro	50%-50%	(50, 50)	ReLU	Adam	0,001
Iveco Daily	75%-25%	(50, 50)	ReLU	Adam	0,001
Iveco Daily	50%-50%	(50, 50)	ReLU	Adam	0,001
Todos	75%-25%	(50, 50)	ReLU	Adam	0,001

Fonte: Tuleski, 2023.

Tabela 28 - Hiperparâmetros ótimos do modelo CNN-LSTM para cada veículo (Parte I).

Veículos	**treino-teste**	***learning_rate***	***batch_size***	***num_epochs***
Ford Ranger	75%-25%	0,01	32	10
Ford Ranger	50%-50%	0,01	32	15
Fiat Toro	75%-25%	0,01	32	10
Fiat Toro	50%-50%	0,01	32	15
Iveco Daily	75%-25%	0,01	32	10
Iveco Daily	50%-50%	0,01	32	15
Todos	75%-25%	0,01	32	10

Fonte: Tuleski, 2023.

Tabela 29 - Hiperparâmetros ótimos do modelo CNN-LSTM para cada veículo (Parte II).

Veículos	**treino-teste**	***dropout_rate***	***filters***	***Kernel_size***	***Pool_size***
Ford Ranger	75%-25%	0,2	64	(5, 5)	(3, 3)
Ford Ranger	50%-50%	0,2	32	(5, 5)	(3, 3)
Fiat Toro	75%-25%	0,2	64	(5, 5)	(3, 3)
Fiat Toro	50%-50%	0,2	32	(5, 5)	(3, 3)
Iveco Daily	75%-25%	0,2	64	(5, 5)	(3, 3)
Iveco Daily	50%-50%	0,2	32	(5, 5)	(3, 3)
Todos	75%-25%	0,2	64	(5, 5)	(3, 3)

Fonte: Tuleski, 2023.

Tabela 30 - Hiperparâmetros ótimos do modelo CNN-LSTM para cada veículo (Parte III).

Veículos	**treino-teste**	***lstm_units***	***lstm_dropout***
Ford Ranger	75%-25%	64	0,3
Ford Ranger	50%-50%	32	0,3
Fiat Toro	75%-25%	64	0,3
Fiat Toro	50%-50%	32	0,3
Iveco Daily	75%-25%	64	0,3
Iveco Daily	50%-50%	32	0,3
Todos	75%-25%	64	0,3

Fonte: Tuleski, 2023.

6.3 DISCUSSÃO DOS RESULTADOS

Os resultados apresentados são para 3 modelos de veículos com motor de ignição por compressão diferentes, 2 camionetes e 1 caminhão. Os resultados foram separados por modelo de veículo para avaliar qual modelo obteve melhor desempenho e posteriormente usando dados de todos os veículos ao mesmo tempo para verificar a capacidade de generalização dos modelos de classificação.

6.3.1 FORD RANGER

Para a *Ford Ranger*, os dados coletados foram os com mais ruído no ambiente, entre os três modelos estudados nesse trabalho. Na sequência foi analisado os resultados obtidos para a primeira e segunda estratégia. As pontuações de desempenho dos modelos investigados em relação a primeira estratégia da divisão de treinamento-teste são apresentadas na Tabela 31.

Tabela 31 - Pontuação estatística para a Ford Ranger, em que μ e σ representam respectivamente média e desvio padrão (Estratégia I).

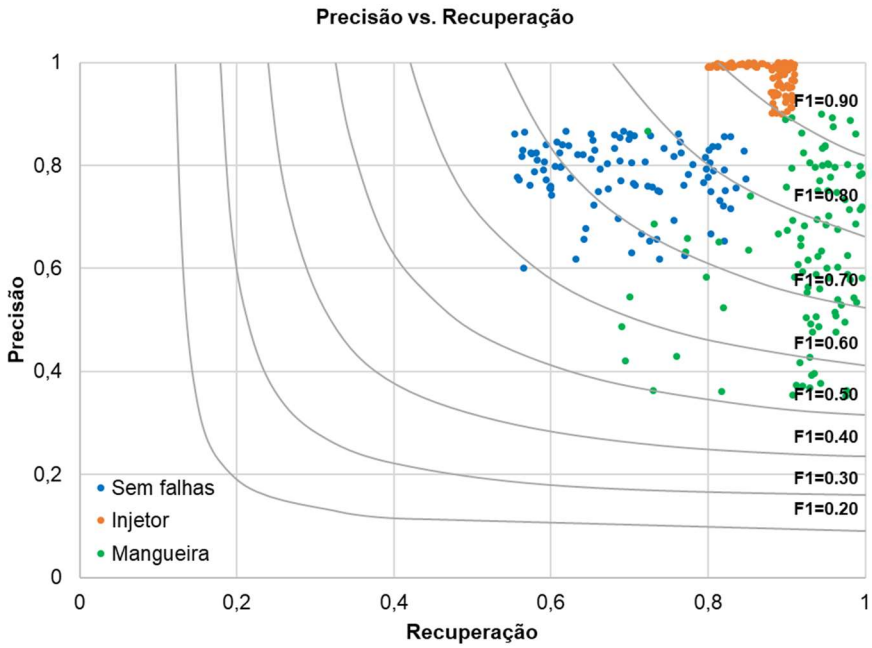
Modelos	**Tempo de Execução Relativo**	**Acurácia (%)**		**F1 Score**	
		μ	σ	μ	σ
MFCC-CNN	1	40,11	3,01	0,327	0,062
MLP	0,571	78,10	4,63	0,773	0,016
SVM	0,064	64,60	6,71	0,769	0,047
Gradient Boosting	0,096	55,00	1,04	0,446	0,080
RF	0,020	65,40	6,49	0,617	0,022
CNN	0,954	79,23	5,46	0,781	0,030
CNN-LSTM	0,966	81,12	2,55	0,803	0,080

Fonte: Tuleski, 2023.

Pela análise dos dados da Tabela 31, pode-se perceber que a CNN-LSTM, com acurácia de 81%, foi o modelo que apresentou o melhor desempenho de classificação, enquanto o MFCC-CNN, com acurácia de 40%, obteve o desempenho inferior. Além disso, pode-se perceber que houve uma redução de custo computacional (tempo relativo de execução) de aproximadamente 4% em relação a abordagem do MFCC-CNN.

Com a finalidade de analisar o desempenho da CNN-LSTM por classe, um gráfico de valores de precisão e recuperação foi feito para as 100 execuções conforme ilustrado na Figura 2824. Na Figura 2824, pode-se notar que a falha no injetor é a classe com a melhor pontuação de predição, enquanto as outras duas classes apresentam uma dispersão considerável no gráfico. As linhas de contorno representam a pontuação F1 para as combinações de precisão e recuperação.

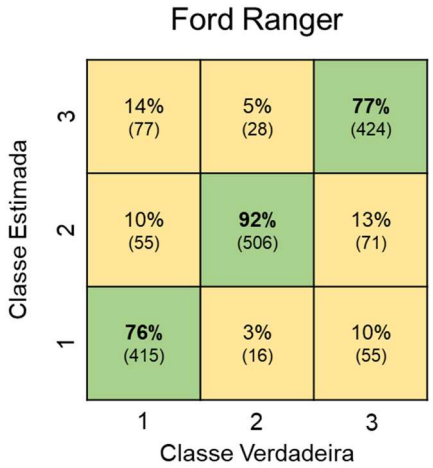
Figura 2824 - Pontuações de precisão e recuperação para cada condição de falha da Ford Ranger (Estratégia I).



Fonte: Tuleski, 2023.

Com base no modelo de melhor desempenho desenvolvido para a *Ford Ranger*, foi criada uma matriz de confusão para analisar os resultados em termos de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos. A Figura apresenta a matriz de confusão e permite observar que o modelo obteve uma classificação mais precisa para as falhas de injetor, enquanto a classificação menos precisa foi para a condição sem falhas e a mangueira desconectada.

Figura 29 - Matriz de confusão da Ford Ranger para método de melhor desempenho (CNN-LSTM) (Estratégia I).



Fonte: Tuleski, 2023.

As pontuações de desempenho dos modelos investigados em relação a segunda estratégia da divisão de treinamento-teste são apresentadas na Tabela 32.

Tabela 32 - Pontuação estatística para a Ford Ranger, em que μ e σ representam respectivamente média e desvio padrão (Estratégia II).

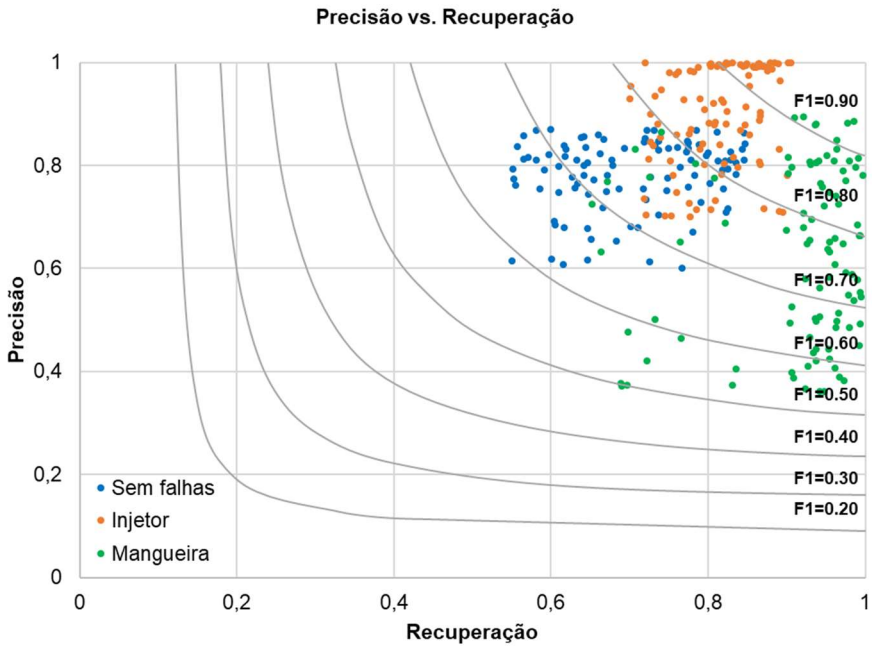
Modelos	**Tempo de Execução Relativo**	**Acurácia (%)**		**F1 Score**	
		μ	σ	μ	σ
MFCC-CNN	1	39,14	3,03	0,341	0,045
MLP	0,522	75,63	4,18	0,732	0,033
SVM	0,066	60,64	3,56	0,591	0,085
Gradient Boosting	0,096	52,19	1,71	0,479	0,024
Random Forest	0,036	65,71	5,91	0,636	0,066
CNN	0,854	77,56	4,36	0,836	0,047
CNN-LSTM	0,985	78,54	3,45	0,779	0,090

Fonte: Tuleski, 2023.

Pela análise da Tabela 32, pode-se perceber que a CNN-LSTM, com acurácia de 78%, foi o modelo que apresentou o melhor desempenho de classificação, enquanto o MFCC-CNN, com acurácia de 39%, obteve o desempenho mais baixo. Além disso, pode-se perceber que houve uma redução de custo computacional (tempo relativo de execução) de aproximadamente 2% em relação a abordagem do MFCC-CNN.

Com a finalidade de analisar o desempenho da CNN-LSTM por classe, um gráfico de valores de precisão e recuperação foi plotado para as 100 execuções conforme visto na Figura 25 onde pode-se notar que a falha no injetor é a classe com a melhor pontuação de predição, enquanto as outras duas classes apresentam uma dispersão considerável no gráfico.

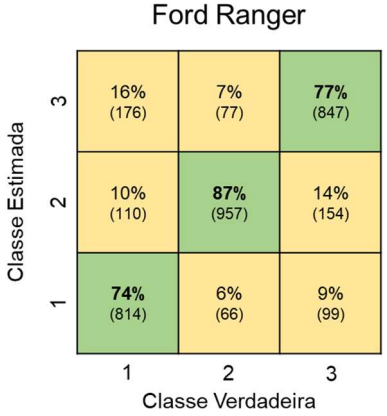
Figura 25 - Pontuações de precisão e recuperação para cada condição de falha da Ford Ranger. (Estratégia II).



Fonte: Tuleski, 2023.

Com base no modelo de melhor desempenho desenvolvido para a *Ford Ranger*, tendo em conta a divisão da base treino-teste seguindo a segunda estratégia, foi criada uma matriz de confusão para analisar os resultados em termos de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos. A Figura 26 apresenta a matriz de confusão e permite observar que o modelo obteve uma classificação mais precisa para as falhas de injetor, enquanto a classificação menos acurada permanece sedo para o motor sem falha e mangueira desconectada.

Figura 26 - Matriz de confusão da Ford Ranger para método de melhor desempenho (CNN-LSTM) (Estratégia II).



Fonte: Tuleski, 2023.

Finalmente, conforme mostra a Tabela 33, se compararmos os melhores modelos para cada estratégia, podemos observar que a primeira estratégia obteve uma acurácia melhor. Isso demonstra que a capacidade de aprendizado do modelo utilizando apenas a base de dados de um veículo pode refletir o viés de amostragem. Nesse caso, o viés de amostragem pode estar relacionado a amostra de dados utilizada para treinamento do modelo, quando essa amostra não é representativa o suficiente, isso pode gerar resultados tendenciosos. Esse tipo de viés geralmente acontece quando os dados coletados veem diretamente de um único veículo.

Tabela 33 – Comparação entre os resultados de Acurácia e *F1-Score* para cada estratégia de divisão de base-treino selecionada para a Ford Ranger.

Modelos	**Estratégia Treino-Teste**	**Acurácia (%)**		**F1 Score**	
		μ	σ	μ	σ
CNN-LSTM	75%-25%	81,12	2,55	0,803	0,080
CNN-LSTM	50%-50%	78,54	3,45	0,779	0,090

Fonte: Tuleski, 2023.

6.3.2 FIAT TORO

Para o veículo *Fiat Toro*, o sinal foi capturado com menos ruído do ambiente em comparação a *Ford Ranger*, mas ainda é possível notar a presença de ruído e regime transiente, ou seja, variação de rotação mesmo com o motor em marcha lenta. Os resultados de desempenho dos modelos, considerando a primeira estratégia, estão representados na Tabela 34.

Tabela 34 - Pontuação estatística para o Fiat Toro, em que μ e σ representam respectivamente média e desvio padrão (Estratégia I).

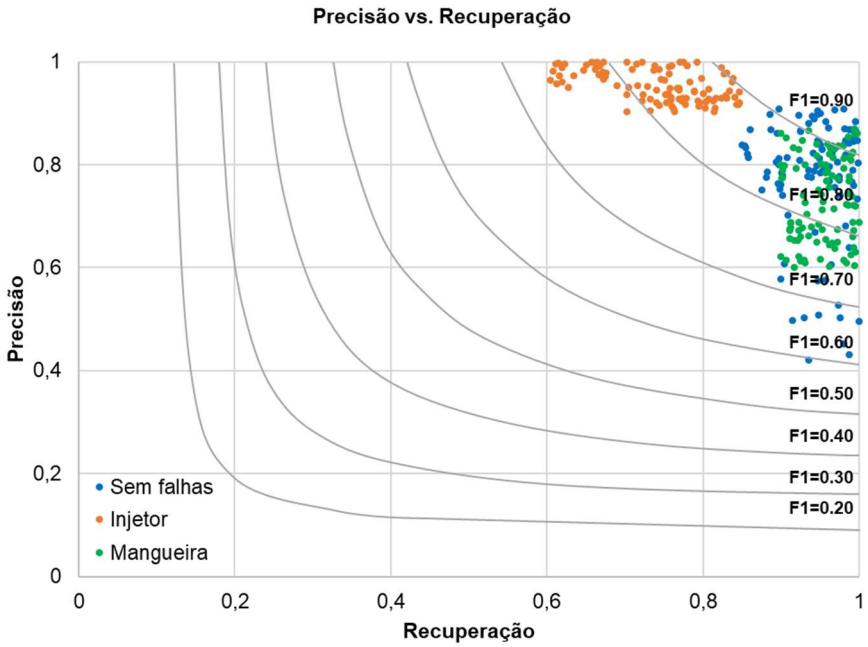
Modelos	**Tempo de Execução Relativo**	**Acurácia (%)**		**F1 Score**	
		μ	σ	μ	σ
MFCC-CNN	1	38,45	3,96	0,365	0,057
MLP	0,756	78,10	3,88	0,764	0,039
SVM	0,024	64,60	5,41	0,769	0,031
Gradient Boosting	0,022	52,46	2,04	0,468	0,045
Random Forest	0,033	61,23	6,49	0,654	0,040
CNN	0,923	81,23	4,12	0,802	0,012

CNN-LSTM	0,947	82,63	3,69	0,811	0,046

Fonte: Tuleski, 2023.

Pela análise da Tabela 34, pode-se notar que o CNN-LSTM novamente foi o modelo de ML em que se obteve os melhores critérios de desempenho, acompanhado pelo MLP, enquanto MFCC-CNN continuou sendo o pior. Também, existe uma redução no tempo computacional relativo de 6% em relação ao método CNN. Com o objetivo de analisar o desempenho do CNN-LSTM por classe, um gráfico de precisão e recuperação para as 100 iterações pode ser verificado na Figura 27.

Figura 27 - Pontuações de precisão e recuperação para cada condição de falha do Fiat Toro (Estratégia I).

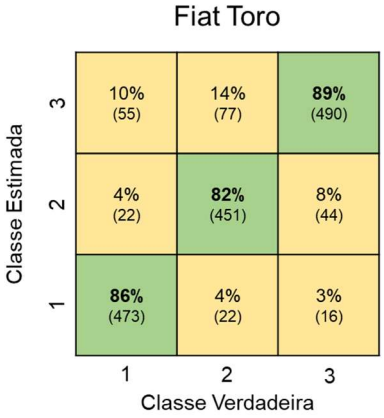


Fonte: Tuleski, 2023.

Como pode ser percebido na Figura 27, a condição sem falha e mangueira de admissão de ar desconectada foram as classes com melhores pontuações de predição. Os resultados são ligeiramente melhores do que o primeiro veículo, ainda que a predição contenha uma variância considerável.

Em conjunto com o gráfico *F1-score*, para o *Fiat Toro*, gerou-se também a matriz de confusão para o método que obteve maior desempenho, conforme Figura 28. Pode-se perceber que novamente a classificação da classe mangueira de admissão de ar desconectada obteve o melhor resultado e a classe injetor desconectado obteve o pior resutlado.

Figura 28 - Matriz de confusão do Fiat Toro para método de melhor desempenho (CNN-LSTM) (Estratégia I).



Fonte: Tuleski, 2023.

As pontuações de desempenho dos modelos investigados em relação a segunda estratégia da divisão de treinamento-teste são apresentadas na Tabela 35.

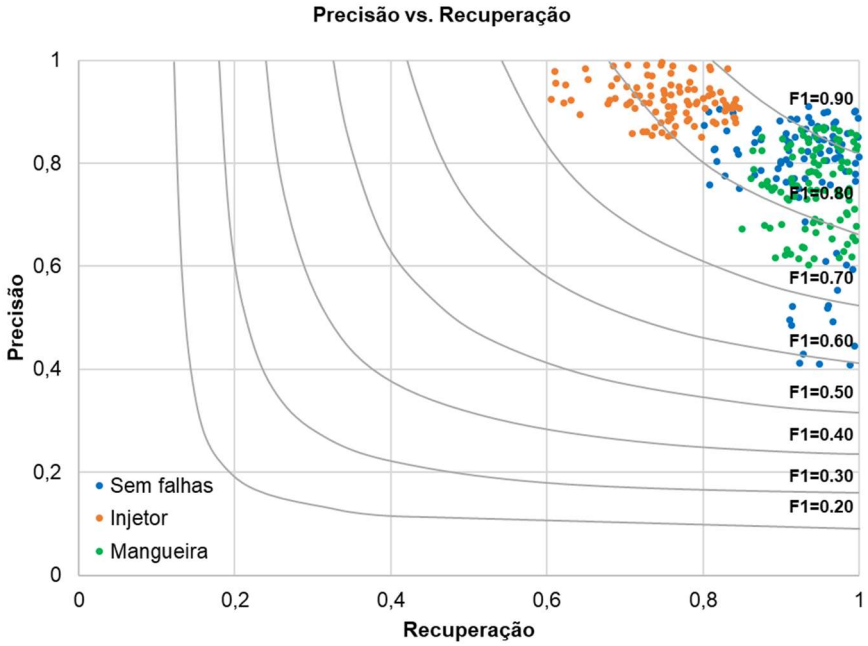
Tabela 35 - Pontuação estatística para a Toro, em que μ e σ representam respectivamente média e desvio padrão (Estratégia II).

Modelos	**Tempo de Execução Relativo**	**Acurácia (%)**		**F1 Score**	
		μ	σ	μ	σ
MFCC-CNN	1	37,12	5,36	0,326	0,045
MLP	0,856	75,01	3,45	0,726	0,065
SVM	0,046	62,03	4,56	0,615	0,014
Gradient Boosting	0,038	50,23	3,56	0,478	0,022
Random Forest	0,065	60,45	5,78	0,598	0,036
CNN	0,974	80,20	2,45	0,795	0,015
CNN-LSTM	0,985	81,65	3,67	0,806	0,054

Fonte: Tuleski, 2023.

Analisando a Tabela 35, pode-se notar que o CNN-LSTM novamente obteve os melhores critérios de desempenho, enquanto MFCC-CNN continuou sendo o pior. Também, existe uma redução no tempo computacional relativo de 2% em relação ao método MFCC-CNN. Com o objetivo de analisar o desempenho do CNN-LSTM por classe, um gráfico de precisão e recuperação para as 100 interações pode ser verificado na Figura .

Figura 34 – Pontuações de precisão e recuperação para cada condição de falha do Fiat Toro (Estratégia II).

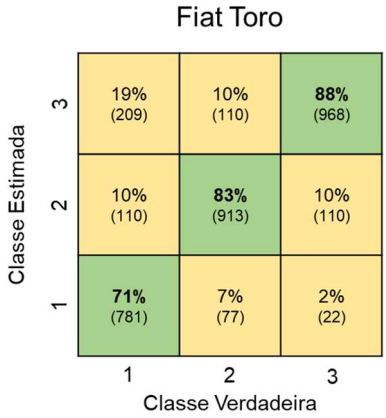


Fonte: Tuleski, 2023.

Como pode ser percebido na Figura , a condição de injetor desconectado e mangueira de admissão de ar desconectada foram as classes com melhores pontuações de predição. Opostamente a *Fiat Toro*, a falha de injetor desconectado não é bem diagnosticada.

Em conjunto com o gráfico *F1-score*, para a *Fiat Toro*, gerou-se também a matriz de confusão para o método que obteve maior desempenho, conforme Figura 29. Pode-se perceber que a classificação da classe injetor desconectado obteve o melhor resultado e a classe sem falha obteve o pior resultado.

Figura 29 – Matriz de confusão do Fiat Toro para método de melhor desempenho (CNN-LSTM) (Estratégia II).



Fonte: Tuleski, 2023.

Em conclusão, conforme mostra a Tabela 36, se compararmos os melhores modelos para cada estratégia, podemos observar que a primeira estratégia obteve uma acurácia melhor. Novamente as mesmas causas de viés observadas na *Ford Ranger* podem ter influenciado o modelo.

Tabela 36 – Comparação entre os resultados de Acurácia e *F1-Score* para cada estratégia de divisão de base-treino selecionada para o Fiat Toro.

Modelos	**Estratégia Treino-Teste**	**Acurácia (%)**		**F1 Score**	
		μ	σ	μ	σ
MFCC-LSTM	75%-25%	82,63	3,69	0,811	0,046
MFCC-LSTM	50%-50%	81,65	3,67	0,806	0,054

Fonte: Tuleski, 2023.

6.3.3 IVECO DAILY

Para o *Iveco Daily*, que apresenta os áudios com menos ruídos externos e com baixíssima variação de rotação em marcha lente. No entanto, ressalta a hipótese de o próprio sinal do veículo ser tão ruidoso que oculta ruídos advindos de outras fontes. Para esse veículo, os valores de performance também foram levantados na Tabela 37.

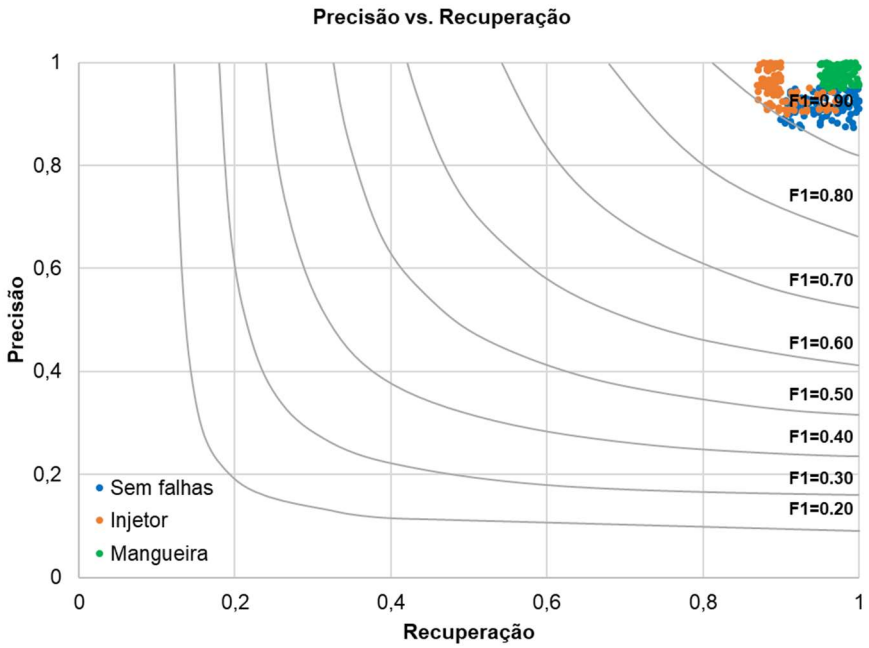
Tabela 37 – Pontuação estatística para o Iveco Daily, em que μ e σ representam respectivamente média e desvio (Estratégia I).

Classificador (Modelo)	**Tempo de Execução Relativo**	**Acurácia (%)**		**F1 Score**	
		μ	σ	μ	σ
MFCC-CNN	1	30,22	0,156	0,271	0,065
MLP	0,777	86,45	1,65	0,846	0,056
SVM	0,036	67,23	0,63	0,634	0,041
Gradient Boosting	0,026	95,64	1,56	0,945	0,025
Random Forest	0,014	96,36	1,07	0,957	0,036
CNN	0,923	96,12	1,02	0,951	0,056
CNN-LSTM	0,943	97,03	0,33	0,963	0,021

Fonte: Tuleski, 2023.

Na Tabela 37, pela sua análise, nota-se que RF é o modelo de aprendizagem de máquina que apresenta os melhores resultados desempenho de classificação, estritamente acompanhado pelo *Random Forest* e *Gradient Boosting*, enquanto, novamente o MFCC-CNN é o pior resultado. Para esse caso, o custo de processamento computacional ou tempo de execução relativo abaixou 6% em comparação com a CNN. Portanto, para a análise do desempenho da CNN-LSTM por classe, plotou-se o gráfico de precisão e recuperação após 100 interações conforme a Figura 30.

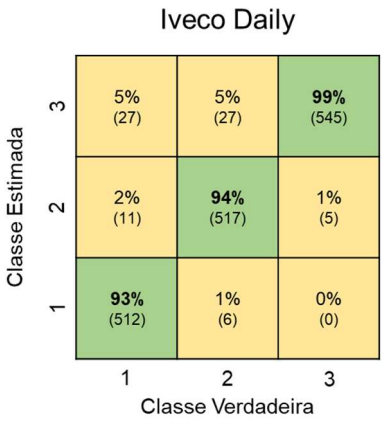
Figura 30 - Pontuações de precisão e recuperação para cada condição de falha do Iveco Daily (Estratégia I).



Fonte: Tuleski, 2023.

Como pode ser visto na Figura 30, esse veículo apresentou os melhores resultados de predição das classes consideradas, tendo previsto corretamente praticamente 100% das falhas do injetor desligado. Por fim, este é o caso com menor variação, o que mostra um procedimento de validação mais robusto. A matriz de confusão para o veículo Iveco Daily, baseada no método de melhor desempenho, é mostrada na Figura 3731. A classe de mangueira desconectada apresentou o maior número de verdadeiros positivos, com uma baixa taxa de classificação incorreta de falhas.

Figura 3731 - Matriz de confusão da Iveco Daily para método de melhor desempenho (CNN-LSTM) (Estratégia I).



Fonte: Tuleski, 2023.

Em seguida, os modelos foram novamente treinados e testados, porém seguindo a segunda estratégia, a performance do modelo está descrita na Tabela 38.

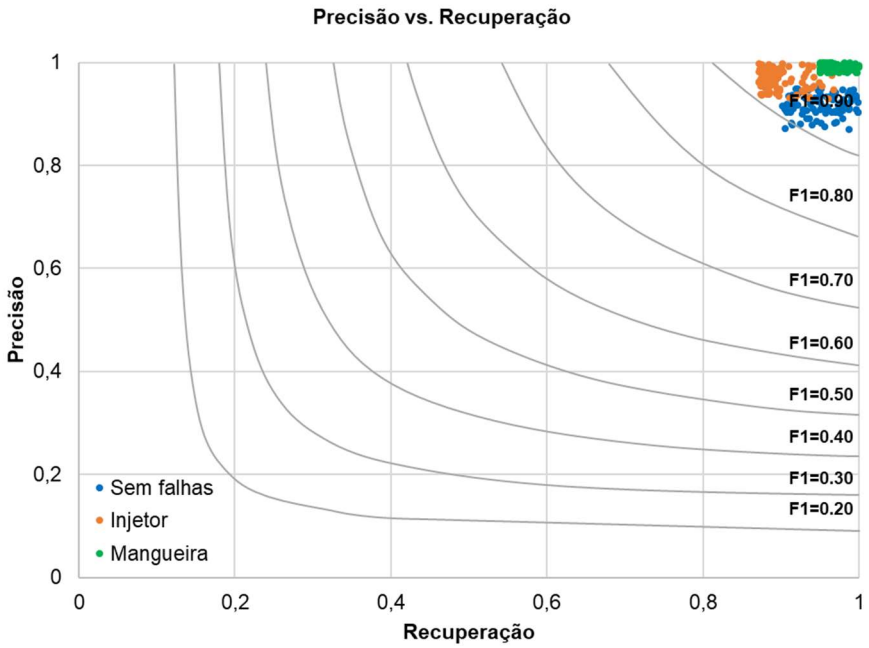
Tabela 38 - Pontuação estatística para o Iveco, em que μ e σ representam respectivamente média e desvio (Estratégia II).

Classificador (Modelo)	**Tempo de Execução Relativo**	**Acurácia (%)**		**F1 Score**	
		μ	σ	μ	σ
MFCC-CNN	1	29,36	0,45	0,285	0,077
MLP	0,856	84,12	1,22	0,832	0,042
SVM	0,064	66,53	0,56	0,624	0,060
Gradient Boosting	0,046	94,55	1,44	0,935	0,065
Random Forest	0,023	95,41	0,96	0,978	0,075
CNN	0,988	96,21	0,45	0,942	0,044
CNN-LSTM	0,989	96,36	0,43	0,958	0,012

Fonte: Tuleski, 2023.

Na Tabela 38, pela sua análise, nota-se que CNN-LSTM é o modelo de aprendizagem de máquina que apresenta os melhores resultados desempenho de classificação, enquanto, o MFCC-CNN é o pior resultado. Para esse caso, o custo de processamento computacional ou tempo de execução relativo abaixou 1% em comparação com a CNN. Portanto, para a análise do desempenho da CNN-LSTM por classe, plotou-se o gráfico de precisão e recuperação após 100 interações conforme a Figura *32*.

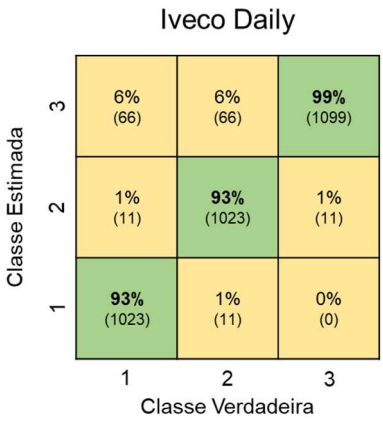
Figura 32 - Pontuações de precisão e recuperação para cada condição de falha do Iveco (Estratégia II).



Fonte: Tuleski, 2023.

Como pode ser visto na Figura *32*, esse veículo apresentou os melhores resultados de predição das classes consideradas, tendo previsto corretamente praticamente 100% das falhas do injetor desligado. Por fim, este é o caso com menor variação, o que mostra um procedimento de validação mais robusto. A matriz de confusão para o veículo Iveco Daily, baseada no método de melhor desempenho, é mostrada na Figura . A classe de mangueira desconectada apresentou o maior número de verdadeiros positivos, com uma baixa taxa de classificação incorreta de falhas.

Figura 39 - Matriz de confusão da Iveco Daily para método de melhor desempenho (CNN-LSTM) (Estratégia II).



Fonte: Tuleski, 2023.

Enfim, conforme mostra a Tabela 39, se compararmos os melhores modelos para cada estratégia, podemos observar que a primeira estratégia obteve uma acurácia melhor. Novamente a influência do viés pode ser um dos principais motivos da diferença de acurácia obtida em comparação as duas estratégias abordadas.

Tabela 39 - Comparação entre os resultados de Acurácia e *F1-Score* para cada estratégia de divisão de base-treino selecionada para o *Iveco Daily*.

Modelos	**Estratégia Treino-Teste**	**Acurácia (%)**		**F1 Score**	
		μ	σ	μ	σ
CNN-LSTM	75%-75%	97,03	0,33	0,963	0,021
CNN-LSTM	50%-50%	96,36	0,43	0,958	0,012

Fonte: Tuleski, 2023.

6.3.4 GENERALIZAÇÃO DOS MODELOS

Por fim, considerando os dados de todos os veículos, com o objetivo de verificar a capacidade de generalização dos modelos, seguindo somente a primeira metodologia com a divisão da base treino-teste em 75%-25% com amostras aleatórias, pode-se verificar os resultados obtidos de cada modelo na Tabela 40.

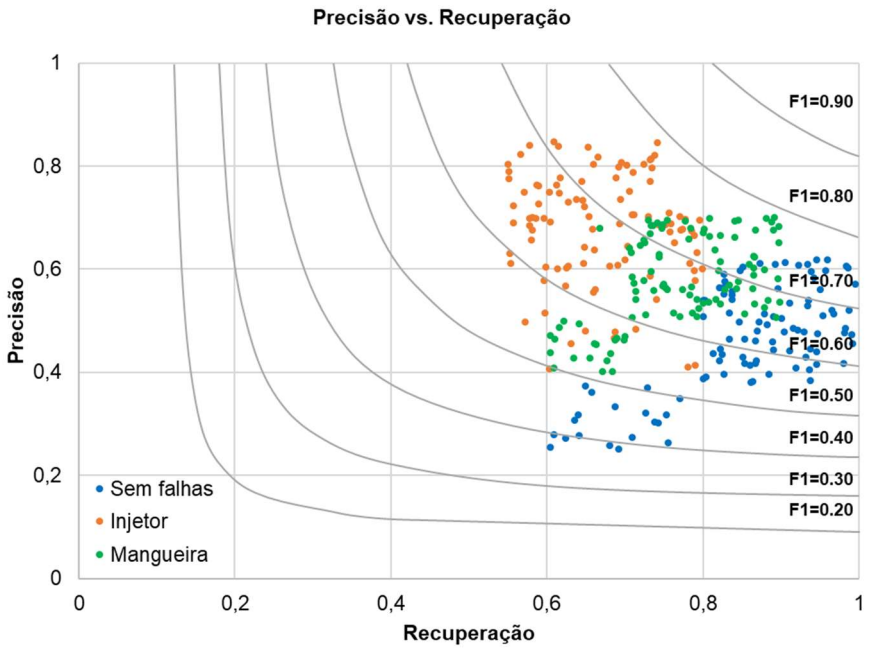
Tabela 40 - Pontuação estatística para o todo o conjunto de dados, em que μ e σ representam respectivamente média e desvio.

Classificador (Modelo)	**Tempo de Execução Relativo**	**Acurácia (%)**		**F1 Score**	
		μ	σ	μ	σ
MFCC-CNN	1	45,66	4,56	0,445	0,057
MLP	0,856	55,98	3,84	0,543	0,023
SVM	0,026	32,85	5,69	0,315	0,036
Gradient Boosting	0,021	37,22	2,56	0,365	0,069
Random Forest	0,017	34,69	3,87	0,335	0,041
CNN	0,942	66,12	2,58	0,658	0,023
CNN-LSTM	0,956	67,77	4,65	0,668	0,034

Fonte: Tuleski, 2023.

Na Tabela 40, observa-se que a rede neural CNN-LSTM é o modelo de aprendizagem de máquina que apresenta os melhores resultados de desempenho de classificação. Em geral os modelos de rede neural obtiveram melhores resultados o que demonstra sua melhor capacidade de generalização quando os dados são tratados de maneira conjunta. Portanto, para a análise do desempenho da CNN-LSTM por classe, plotou-se o gráfico de precisão e recuperação após 100 interações conforme a Figura *33*.

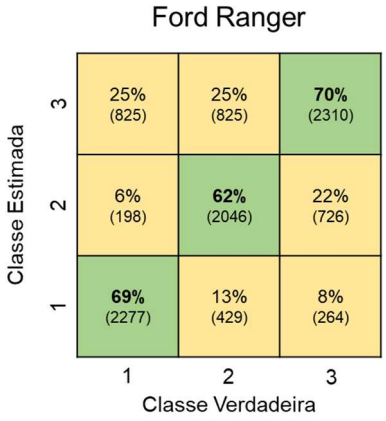
Figura 33 - Pontuações de precisão e recuperação para cada condição de falha de todo o conjunto de dados.



Fonte: Tuleski, 2023.

Como pode ser visto na Figura *34*, as classes obtiveram uma dispersão um pouco maior em relação aos outros modelos, isso provavelmente está correlacionado a dificuldade que o modelo encontra de classificar falhas quando os dados de todos os veículos são misturados. A matriz de confusão para os veículos generalizados, baseada no método de melhor desempenho, é mostrada na Figura *34*. A classe de mangueira desconectada apresentou o maior número de verdadeiros positivos, porém no geral todas as classes obtiveram resultados muito próximos.

Figura 34 - Matriz de confusão para todo o conjunto de dados para método de melhor desempenho (CNN-LSTM).



Fonte: Tuleski, 2023.

CAPÍTULO 7
CONCLUSÕES

Baseado nos estudos previamente encontrados na literatura, o trabalho buscou como principal objetivo realizar uma pesquisa sobre a viabilidade de diagnosticar falhas em motores de ignição por compressão através da aquisição de sinais de áudio. Em paralelo ao objetivo principal, os objetivos específicos ou secundários, como definição de uma metodologia para a coleta de dados, classificação das falhas encontradas em conjunto com o tratamento dos dados, definição dos parâmetros e métodos de ML, desenvolvimento de um modelo de IA para a classificação das falhas, melhoria no desempenho do modelo, otimização dos hiperparâmetros, redução de viés nos modelos e capacidade de generalização dos modelos, foram amplamente analisados.

A definição de uma metodologia robusta para a coleta de dados dos veículos foi definida com base em uma proposta que pudesse ser facilmente reproduzida, portanto os sinais de áudios foram coletados, em posições estratégicas dos veículos, com um dispositivo de fácil acesso aos possíveis usuário dessa aplicação, um *smartphone*.

Bem como as falhas selecionadas através de um estudo de campo com diversas oficinas mostrou quais eram as falhas mais reproduzíveis e que poderiam agregar mais valor em um primeiro momento de diagnóstico. Baseado em três situações distintas, os veículos selecionados para o estudo foram escolhidos de acordo com a disponibilidade e que representassem diferentes situações, ou seja, diferentes assinaturas de áudios devido a características do motor de ignição por compressão aplicado para cada caso.

Com base nos métodos mais utilizados na literatura, os principais modelos de aprendizado de máquina foram selecionados para avaliar qual obteria o maior desempenho, junto a isso, os hiperparâmetros foram otimizados caso a caso e no fim foi possível encontrar um modelo que obtivesse melhor resultados para cada veículo. Duas estratégias diferentes foram definidas para a divisão da base de treino-teste afim de avaliar a influência do viés nos resultados dos modelos de aprendizado de máquina selecionados. Observou-se que o melhor modelo foi a combinação de duas redes neurais CNN-LSTM, em que o resultado de acurácia foi maior em todos os casos.

Em conclusão, ao analisar o desempenho de cada modelo em diferentes veículos, os resultados sugerem que a abordagem mais adequada para diagnóstico de motores de ignição por compressão é a utilização de modelos treinados com uma base de dados proveniente de um mesmo veículo. Além disso, com o conjunto de dados disponíveis, indicando que um modelo único que funciona bem para todos os tipos de veículos ainda não é possível. Devido a disponibilidade inicial de veículos, para abranger mais classes, o autor recomenda a realização de uma coleta de dados em campo em parceria com oficinas e redes credenciadas para verificar a eficácia do método proposto no diagnóstico de um

número maior de falhas. Além disso, é importante investigar o potencial da abordagem em outros veículos automotivos, tais como por exemplo, motores a gasolina e etanol, que representam uma frota significativa do mercado brasileiro.

Para futuros estudos, recomenda-se que sejam utilizados conjuntos de dados maiores e mais variados, a fim de avaliar a capacidade da abordagem proposta em diferentes contextos. Tais como, serie interessante investigar a possibilidade de se utilizar outras técnicas de processamento de sinais ou ML, a fim de melhorar ainda mais a precisão da classificação. Por fim, seria importante considerar outras aplicações além do diagnóstico de falhas em motores a Diesel explorando o potencial da abordagem em outras áreas da engenharia ou da indústria.

REFERÊNCIAS BIBLIOGRÁFICAS

1. AHMED,R et al. **Automotive Internal-Combustion-Engine Fault Detection and Classification Using Artificial Neural Network Techniques**. IEEE Transactions on Vehicular Technology, 64:21-33, jan. 2015.
2. AL-BURGHARBEE, H; SALEH, N. **Rolling Element Bearing Fault Diagnosis uisng Spectral Analysis**. Measurement, 73: 170-179, 2015.
3. AMBARISH, D; BIJAN, M. **A comprehensive review of biodiesel as an alternative fuel for compression ignition engine**. Renewable and Sustainable Energy Reviews, 57, 799-821, 2016.
4. ANIS, A et al. **Investigating Pump Cavitation based on Audio Sound Signature Recognition using Artificial Neural Network.** Journal of Physics: Conference Series, 1-6, 2020.
5. AYATI, M et al. **Classification-based fuel injection fault detection of a trainset diesel engine using vibration signature analysis**. Journal of Dynamic Systems, Measurement and Control, 142(5), mar. 2020.
6. BELETE, D, M; MANJAIAH, D,H. **Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test reults.** International Journal of Computers and Applications, 44(1), 1-12, 2021.
7. BERGSTRA, J; BENGIO, Y. **Random Search for Hyper-Parameter Optimization**. Journal of Machine Learning Research, 13, 281-305, 2012.
8. BIN, L et al. **Analog-to-digital converters**. IEEE Signal Processing Magazine, 22:69-77, November 2005.
9. BISHOP, C M. **Neural networks for pattern recognition.** Oxford University Press. 1995.
10. BORRE, A et al. **Machine Fault Detection Using a Hybrid CNN-LSTM Attention-Based Model.** Sensors, 23(9), 451, 2023.
11. BOVIK, C. **Handbook of Image and Video Processing**. United States, Massachusetts, Academic Press, 2009.
12. BRACEWLL; RONALD, N. **The Fourier Transform and Its Applications.** United States, New York, McGraw, 1986.
13. BREIMAN, L. **Random Forests**. Machine learning, 45:5-32, 2001.
14. BREIMAN, L; FRIEDMAN, J; OLSHEN, R; STONE, C. **Classification and Regression Trees.** CRC Press, 1984.
15. CARREIRA, F, VELLOSO, P; ROSA, A. **A Support Vector Machine para detecção de desalinhamentos em turbinas eólicas.** Anais do Congresso Brasileiro de Engenharia Mecânica, Ceará, Maracanaú 2012.
16. CHEN, T; GUESTRIN, C. **Xgboost: A scalable tree boosting system.** In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 785-794, 2016.
17. COIFMAN, R et al. **Signal processing and compression with wavelet packets**. Wavelets and Their Applications, 442:363-379, 1994.
18. COOLEY, J.W; TURKEY, J.W. **An Algorithm for the Machine Computation of the Complex Fourier Series.** Mathermatics of Computation, 19, 297-301, 1965.

19. DAUBECHIES, I. **Ten lectures on wavelets.** Society for Industrial and applied mathematics, 1992.

20. DAVIS, B; MERMELSTEIN, P. **Comparison of parametric representations for monosyllabic word recognitions in continuously spoken sentences**. IEEE Transacations on acoustics, Speech, and Signal processing, 28:357-366, 1980.

21. DUMOULIN, V; VISIN, F. **A guide to convolution arithmetic for deep learning**. arCiv prepint arXiv, Jan. 2018.

22. FAKHARIAN, S; AHMADI, M; YILDIRIM, T. **A novel clustering approach based on frequency-domain features for identifying the component faults of an automobile suspension system.** Journal of Vibration and Control, 25:357-370, 2019.

23. FRIEDMAN, H. **Greedy function approximation: A gradient boosting machine**. Annals of Statistics, 1:1189-1232, 2001.

24. FUNGATI, M; SOARES, B; SILVA, C. **A transformada wavelet como ferramenta para análise de sinais em diferentes escalas de tempo e frequência**. Anais do Congresso International de Engenharia, 1:123-128, 2018.

25. GAETAN, F; OLGA, F. **Learnable Wavelet Packet Transform for Data-Adapted Spectograms**. IEEE International Conference on Acoustics, Singapore, Singapore, 22:24140-24149, abr. 2022.

26. GAO, X; YAN, R. Capítulo 5: **Wavelet Packet Transform**. In: GAO, X; YAN, R. Wavelets. 1ª. Edição. Nova Iorque: Editora Springer, 2011. 69-81.

27. GOODFELLOW, I et al. **Deep Learning.** MIT Press, United States, Massachusstes, 2016.

28. GOODFELLOW, I; BENGIO, Y; COURVILLE, A. **Deep Learning.** MIT Press, United States, Massachusstes, 2016.

29. GUPTA, R. **Interoperability and Standardization in On-Board Diagnose Systems**. Proceedings of the International Symposium on Automotive Technology, U.K, Croydon 321-335, 2019.

30. HASTIE, T; TIBSHIRANI, R; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction.** Springer, United States, New York, 2009.

31. HAYKIN, S. **Neural networks and learning machines.** Pearson, United States, New York 2009.

32. HE, P et al. **Core looseness fault identification model based on Mel spectogram-CNN**. Journal of Physics: Conference Series, 2137:1-7, dec. 2021.

33. HENRIQUEZ, P et al. **Review of Automatic Fault Diagnosis Systems Using Audio and Vibration Signals**. IEEE Transactions on Systems, Man and Cybernetics, 642-652, nov. 2013.

34. HOCHREITER, S; SCHMIDHUBER, J. **Long Shoort-Term Memory.** Neural Computation, 9(8), 1735-1780, 1997.

35. HOSMER, J D W; LEMESHOW, S; STURDIVANT, R X. **Applied logistic regression.** John Wiley & Sons. 2013.

36. HOU, L et al. **A fault diagnosis model of marine diesel cylinder based on modified genetic algorithm and multilayer perceptron**. Soft Computing, 24:7603-7613, set. 2019.

37. HUANG, J; TEMTES, C. **Delta-Sigma Data Converters: Theory, Design, and Simulation.** IEEE Pres, 1994.

38. HUANG, Y; BENESTY, J; ELKO, G. **Acoustic Array Systems: Theory, Impementation, and Application**. Springer, 2014.

39. JAFARI, M; MEHDIGHOLI, H; BEHZAD, M. **Vavle Fault Diagnosis in Internal Combustion Engines Using Acoustic Emission and Artificial Neural Network**. Shock and Vibration, 1:1-10, feb. 2014.

40. JENA, P; PANIGRAHI, N. **Motor bike piston-bore fault identification from engine noise signature analysis**. Applied Acoustics, 76:35-47, feb. 2014.

41. JOHN, D. **Advanced comrpession-ignition engines-understanding the in-cylinder process**. Proceedings of the Combustion Institute, 32(2), 2727-2742, 2009.

42. JOLLIFFE, T. **Principal Component Analysis.** Springer, 2002.

43. JONES, A et al. **OBD and Emission Control: Meeting Global Standards.** Journal of Automotive Technology, 45(2), 87-102, 2018.

44. KARPATHY, A et al. **Beyond Short Snippets: Deep Networks for Video Classification.**Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

45. KIANGALA, S; WANG, Z. **An effective adaptive customization framework for small manufacturing plants using gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 enviroment.** Machine Learning with Applications, 4:1-15, jun. 2021.

46. KOHAVI, R. **A study of cross-validation and bootstrap for accuracy estimation and model selection.** Proceedings of the 14th international joint conference on Artificial Intelligence. 2, 1995.

47. KUSUMAWARDHANI, A; PARIKESIT, D. **Market Potential Analysis of Spare Parts Business in the Automotive Industry in Indonesia.** International Journal of Engineering and Advanced Technology (IJEAT), 8(6S), 150-155, 2019.

48. LAGUARTA, J; HUETO, F; SUBIRANA B. **COVID-19 Artificial Intelligence Diagnosis Using Only Cough Recording.** IEEE Open Journal of Engineering in Medicine and Biology, 1:1-9, set. 2020.

49. LIANG, P. **Intelligent Fault Diagnosis of Rolling Element Bearing Based on Convolutional Neural Network and Freqeuncy Spectograms**. IEEE International Conference on Prognostics and Healt Management, jun. 2019.

50. LIAW, A; WIENER, M. **Classification and regression by Random Forest**. R News, 2:18-22, 2002.

51. LIMA, T. **Internacionalização do setor secundário catarinense: movimentos recentes no âmbito da indústria automotiva.** 2015. 84 p. Monografia: Departamente de Economia e Relações Internacionais – Universidade Federal de Santa Catarina, Florianópolis.

52. LUI, L; ZHANG, X; CHEN, Y. **Fault diagnosis of a vehicle's powertrain based on a convolutional neural network.** Sensors, 19:772, 2019.

53. LYONS, G. **Understanding Digital Signal Processing**. Pearson Education, 2011.

54. MALLAT, S. **A wavelet tour of signal processing: the sparse way**. Academic Press, 1998.

55. MASSOUDI, M; VERMA, S; JAIN, R. **Urban Sound Classification using CNN**. 6th International Inventive Computation Technologies (ICICT), jan. 2021.

56. MITRA, K. **Digital Signal Processing: A Computer-Based Approach.** McGraw-Hill Education, 2006.

57. MUSHTAQ, Z; SU, SF; TRAN, QV. **Spectral images based environmental sound classification using CNN with meaningful data augmentation**. Applied Acoustics, 172, jan. 2021.

58. NEYMAN, J. **On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection.** Journal of the Royal Statistical Society, 97(4), 558-606, 1934.

59. NOGUEIRA, S et al. **Prediction of the NOx and CO2 emissions from an experimental dual fuel engine using optimized random forest combined with feature engineering.** Energy, 280, 128066, 2023.

60. OPPENHEIM, A V; SCHAFER, R. **Discrete-Time Signal Processing.** Pearson Education, 2010.

61. PROAKIS, G; MANOLAKIS, G. **Digital Signal Processing: Principles, Algorithms, and Applications.** Prentice Hall, 2006.

62. RABINE, R; JUANG H. **Fundamentals of speech recognition.** Prentiece-Hall, Inc, 1993.

63. RAMTEKE, CHELLADURAI, H; AMARNATH, M. **Diagnosis and Classification of Diesel Engine Components Faults Using Time-Frequency and Machine Learning Approach**. Journal of Vibration Engineering & Technologies, 10: 175-192, aug. 2021.

64. RAMTEKE, CHELLADURAI, H; AMARNATH, M. **Diagnosis of liner scuffing fault of a diesel engine via vibration and acoustic emission analysis**. Journal of Vibration Engineering & Technologies, 8:815-833, oct. 2019.

65. RASSOUL, N et al. **Statistical Analysis of Profile Monitoring.** John Wiley & Sons, Inc. 2011.

66. RAVIKUMAR, N et al. **Fault diagnosis of internal combustion engine gearbox using vibration signals based on signal processing techniques**. Journal of Quality in Maintenance Engineering, 1355-2511, jun. 2020.

67. RISH, I. **An empirical study of the naive Bayes classifier.** In IJCAI workshop on empirical methods in artificial intelligence. Vol. 3, No. 22, 41-46.

68. SASAKI, Y. **The truth of the F-measure.** Teach Tutor Mater, 1(5), 1-6. 2007.

69. SHATNAWI, Y; AL-KHASSAWENEH, M. **Fault Diagnosis in Internal Combustion Engines Using Extension Neural Network**. IEEE Transactions on Industrial Electronics, 61:1434-1443, mar. 2014.

70. SHIBLEE, M; YADAV, K; CHANDRA, B. **Fault diagnosis of internal combustion engine using empirical mode decomposition and artificial neural networks**. International Conference on Intelligent Computing, 188-199, jul. 2017.

71. SHOOTHER, K et al. **The importance of Feature Processing in Deep-Learning-Based Condition Monitoring of Motors**. Mathematical Problems in Engineering, 1:1-23, mai. 2021.

72. SMITH, J. **OBD: A Critical Component in Modern Vehicles.** International Conference on Automotive Engineering, 123-136, 2015.

73. SMITH, J; JOHNSON, A; DAVIS, R. **Audio Signal Processing Techniques for Fault Identification and Diagnosis**. Journal of Acoustic Engineering, 45(2), 78-92, 2021.

74. SMITH, S W. **The Scientist and Engineerin's Guide to Digital Signal Processing**. California Technical Publishing, 1999.

75. SNOEK, J et al. **Practical Bayesian Optization of Machine Learning Algorithms**. Advances in Neural Information Processing Systems, 2012.

76. STEPHAN, H. **Time-to-Digital Converters.** Springer, 2010.

77. SUJESH, G; RAMESH, S. **Modeling and control of diesel engines: A systematic review.** Alexandria Engineering Journal, 57(4), 4033-4048, 2018.

78. SUN, J; YU, L; ZHANG, X. **A principal component analysis-based method for gearbox fault diagnosis**. Journal of Vibration and Control, 22:2874-2887, 2016.

79. SUN, W et al. **A Sparse Auto-encoder-Based Deep Neural Network Approach for Induction Motor Faults Classification**. Measurement, 89:171-178, jul. 2016.

80. SUN, Y et al. **Environment Sound Classification Using a Two-Stream CNN Based on Decision-Level Fusion**. Novel Modeling, Signal Processing and Machine Learning Techniques for Sensor Data, 1-15, abr. 2019.

81. SUTSKEVER, I et al. **Sequence to Sequence Learning with Neural Networks.** Advances in Neural Information Processing Systems, 27, 2014.

82. THANGARAJA, J et al. **A comprehensive review on the current trends, challenges and future prospects for sustainable mobilidty.** Renewable and Sustainable Energy Reviews, 157, 112073, 2022.

83. THOMPSON, K S. **Sampling.** 3rd, Wiley, 2012.

84. TOUME, W (2022). **Mercado de resposição deve crescer 7% até 2023, diz Sindipeças**. Automotive Business. Disponível: https://www.automotivebusiness.com.br/pt/posts/noticias/mercado-de-reposicao-deve-crescer-7-ate-2023-diz-sindipecas/. [21 jan. 2022].

85. UMAPATHY, K et al. **Audio Signal Processing Using Time-Frequency Approachs: Coding, Classification, Fingerprinting, and Watermarking.**EURASIP Journal on Advances in Signal Processing, 451695, 2010.

86. VAPNIK, V; CORTES, C. **Support-vector networks.** Machine Learning, 23:273-297, 1995.

87. VATTERLI, M; KOVACEVIC, J. **Wavelets and subband coding.** Prentice Hall, 1995.

88. WU, D; LIU, H. **An expert system for fault diagnosis in internal combustion engines using wavelet packet transform and neural network**. Expert Systems with Applications, 36:4278-4286, abr. 2019.

89. WU, H *et al.* **Vehicle sound signature recognition by frequency vector principal component analysis.** IEEE Instrumentation and measurement technology conference, 18-21, Março 1998.

90. YANG, M et al. **Audio-based fault diagnosis for belt conveyor rollers**. Neurocomputing, 397:447-456, jul. 2020.

91. YEN, G; LIN, K. **Wavelet Packet Feature Extraction for Vibration Monitoring**. IEEE Transactions on Industrial Electronics, 47:650-667, jun. 2000.

92. ZHANG, Q et al. **Fault Identification Based on PD Ultrasonic Signal Using RNN, DNN and CNN**. Condition Monitoring and Diagnosis (CMD), nov. 2018.

93. ZHAO, R et al. **Deep learning and its applications to machine health monitoring**. Mechanical Systems and Signal Processing, 115:213-237, jan. 2019.

94. ZHOU, Q et al. **Cough Recognition Based on Mel-Spectogram and Convolutional Neural Network**. Front Robot AI, 1-8, mai. 2021.

95. ZICO; ZECA**. 70 Anos da Melhor Música Sertaneja.** São Paulo, 2002, Vol. 3. Faixa 3.