Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

George Leandro dos Santos Pinto

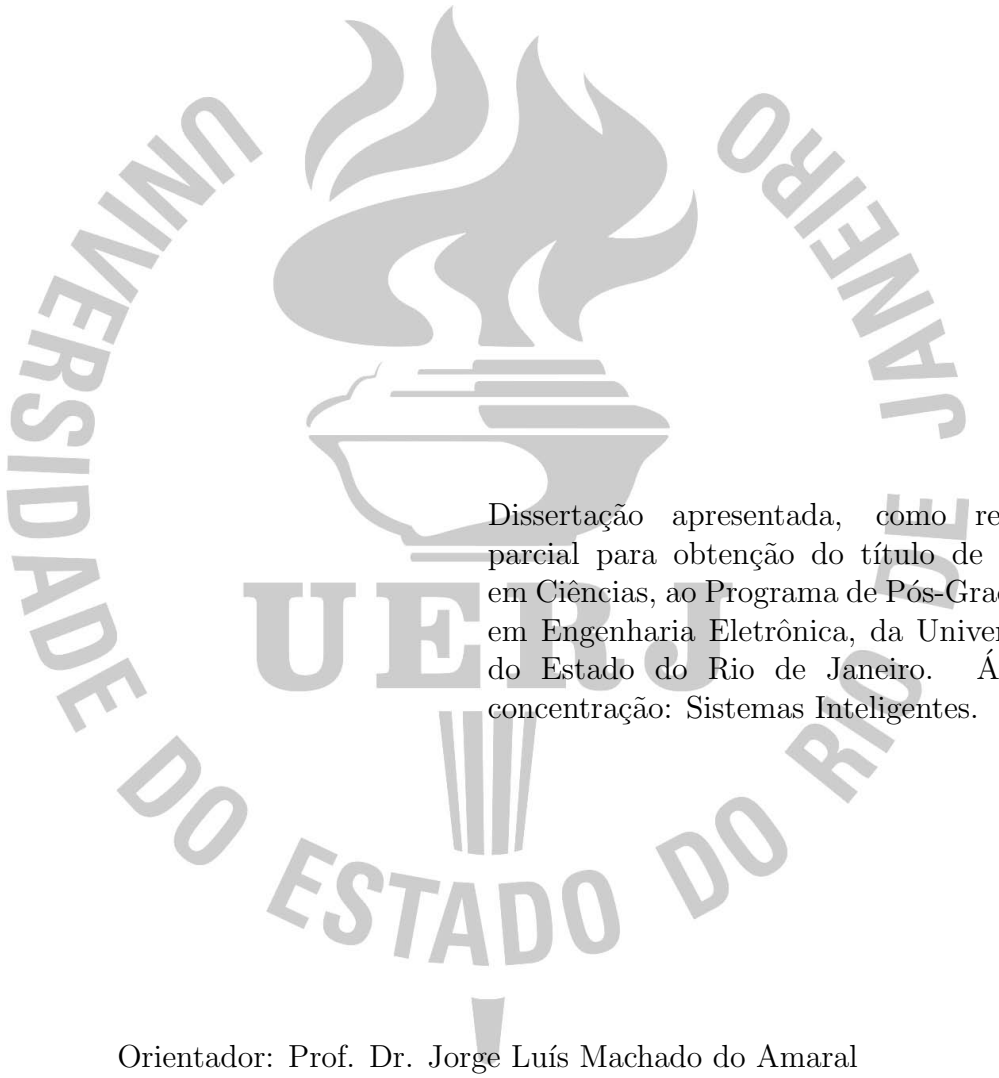
Um Sistema Não Intrusivo para Classificação da Severidade e Estimativa da Profundidade de Danos Causados por Corrosão Interna Usando a Técnica *Potential Drop* e Mapeamento Elétrico com Aplicação de Algoritmos de Aprendizado de Máquinas e Redes Neurais Convolucionais

Rio de Janeiro

2020

George Leandro dos Santos Pinto

Um Sistema Não Intrusivo para Classificação da Severidade e Estimativa da Profundidade de Danos Causados por Corrosão Interna Usando a Técnica *Potential Drop* e Mapeamento Elétrico com Aplicação de Algoritmos de Aprendizado de Máquinas e Redes Neurais Convolucionais



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre em Ciências, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes.

Orientador: Prof. Dr. Jorge Luís Machado do Amaral

Rio de Janeiro

2020

CATALOGAÇÃO NA FONTE

UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

P659 Pinto, George Leandro dos Santos.
Um Sistema não intrusivo para classificação da severidade e estimativa da profundidade de danos causados por corrosão interna usando a técnica potential drop e mapeamento elétrico com aplicação de algoritmos de aprendizado de máquinas e redes neurais convolucionais / George Leandro dos Santos Pinto. – 2020.
97f.

Orientador: Jorge Luís Machado do Amaral
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Engenharia eletrônica - Teses. 2. Redes neurais (Computação) - Teses. 3. Correntes elétricas - Teses. 4. Aço - Corrosão - Teses. 5. Método dos elementos finitos - Teses. I. Amaral, Jorge Luís Machado do. II. Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia. III. Título.

CDU 681.53

Bibliotecária: Júlia Vieira – CRB7/6022

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta tese, desde que citada a fonte.

Assinatura

Data

George Leandro dos Santos Pinto

Um Sistema Não Intrusivo para Classificação da Severidade e Estimativa da Profundidade de Danos Causados por Corrosão Interna Usando a Técnica *Potential Drop* e Mapeamento Elétrico com Aplicação de Algoritmos de Aprendizado de Máquinas e Redes Neurais Convolucionais

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre em Ciências, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes.

Aprovado em: 19 de novembro de 2020

Banca Examinadora:

Prof. Dr. Jorge Luís Machado do Amaral (Orientador)
Faculdade de Engenharia - UERJ

Prof. Dr. José Antônio da Cunha Ponciano Gomes
Universidade Federal do Rio de Janeiro - UFRJ

Prof. PhD. Gilson Alexandre Ostwald Pedro da Costa
Instituto de Matemática e Estatística - UERJ

Prof. Dr. Gil Roberto Vieira Pinheiro
Faculdade de Engenharia - UERJ

Rio de Janeiro

2020

DEDICATÓRIA

Dedico este trabalho a Deus e minha família, em especial à memória de minha vó Noêmia. Também deixo minha lembrança à equipe do LARISA.

AGRADECIMENTO

Em primeiro lugar, agradeço a Deus por me permitir chegar ao término deste trabalho cercado por pessoas que tanto amo. Agradeço à minha família pela compreensão quando não pude ser tão presente como gostaria, por todo o suporte e esforço pra que pudesse chegar aqui, destacando a disposição do meu pai, o zelo da minha mãe e o companheirismo do meu irmão.

Agradeço ao professor Jorge pela dedicação, comprometimento e confiança empenhados, sem os quais este trabalho não seria possível. Sou grato também ao professor Gil que trouxe valorosas contribuições e a parceria com LabCorr que proporcionou grandes trocas de conhecimento. Menciono também o Victor, que com sua vivência no CENPES e generosidade compartilhou importantes conhecimentos sobre o complexo mundo da corrosão, até então para mim desconhecido.

Sou grato a polivalente equipe do LARISA que tanto colaborou com a realização desta dissertação, seja no apoio técnico, fabricação de equipamentos ou nos momentos de descontração que se fazem tão necessários. Em especial ao Everton, Sávio, Noemi, Adriano, Patrícia e Hugo por transmitirem parte de suas experiências e conhecimentos. Também não posso deixar de mencionar os parceiros desde as salas de aula José Rodrigo, Elan e Allan.

O processo de uma dissertação de mestrado exige muitas habilidades e competências, porém esta contém um ingrediente atípico. A pandemia causada pelo Sars-CoV-2 nos fez ver o mundo de perspectiva e dimensões nunca experimentadas, foi necessário interromper nossas atividades presenciais, manter distanciamento de pessoas que amamos, cuidarmos afastados uns dos outros. Não foi um período fácil, época onde a proximidade de quem gostamos deu lugar a uma sensação de impotência, ansiedade e, ao mesmo tempo, de esperança que a normalidade voltaria logo. Eu e minha família fomos acometidos pela Covid-19, graças a Deus sobrevivemos. Isto nos dá força para continuarmos firmes na estrada da vida e tentarmos ser melhores a cada dia.

Por fim, agradeço à Fundação de Amparo à Pesquisa Carlos Chagas Filho do Estado do Rio de Janeiro - FAPERJ - pelo apoio prestado.

Combati o bom combate, completei a corrida, guardei a fé.

São Paulo

RESUMO

PINTO, George Leandro dos Santos. *Um Sistema Não Intrusivo para Classificação da Severidade e Estimativa da Profundidade de Danos Causados por Corrosão Interna Usando a Técnica Potential Drop e Mapeamento Elétrico com Aplicação de Algoritmos de Aprendizado de Máquinas e Redes Neurais Convolucionais.* 97 f. Dissertação (Mestrado em Engenharia Eletrônica) - Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, 2020.

Este trabalho apresenta um método não intrusivo para obtenção de informações sobre danos causados por corrosão interna em placas de aço inoxidável AISI-304 e classificá-los de acordo com sua severidade. Pela aplicação da técnica *Potential Drop* é extraído o mapeamento do gradiente de potencial elétrico, que é examinado por estratégias de processamento de imagens como operações para realce de bordas e segmentação. Com o emprego da Modelagem em Elementos Finitos foram realizadas simulações para gerar exemplos de placas danificadas, utilizando quatro classes de defeitos com geometria alveolar. O estágio de processamento de imagens atua como extrator de atributos que ao serem apresentados a diferentes regressores e classificadores possibilitam ser conhecidas a estimativa da profundidade dos danos e a classificação da sua severidade respectivamente. Com o regressor *Random Forest* foi obtido o erro absoluto médio de 0,0575 mm na estimativa da profundidade dos defeitos. Adicionalmente, com a aplicação de uma Rede Neural Convolucional foi alcançada a acurácia de 94,84% na classificação da gravidade dos danos.

Palavras-chave: Redes neurais convolucionais; Classificação; Regressão; Mapeamento elétrico; Técnica *potential drop*; Corrosão interna; Corrosão alveolar.

ABSTRACT

PINTO, George Leandro dos Santos. *A Non-Intrusive System for Classifying Severity and Estimating the Depth of Damage Caused by Internal Corrosion Using the Potential Drop Technique and Electrical Mapping with Application of Machine Learning Algorithms and Convolutional Neural Networks*. 97 f. Dissertação (Mestrado em Engenharia Eletrônica) - Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, 2020.

This work presents a non-intrusive method to obtain information about damages caused by internal corrosion in AISI-304 stainless steel plates and to classify them according to their severity. By applying the Potential Drop technique, the mapping of the electric potential gradient is extracted, which is examined by image processing strategies such as operations for enhancement of edges and segmentation. With the use of Finite Element Modeling, simulations were carried out to generate examples of damaged plates, using four classes of defects that can be considered as part of pitting corrosion. The image processing stage acts as an extractor of features that, when presented to different regressors and classifiers, make it possible to know the damage depth estimate and the severity classification respectively. With the Random Forest regressor, the mean absolute error of 0.0575mm was obtained in the estimate of the depth of the defects. Additionally, with the application of a Convolutional Neural Network, the accuracy of 94.84% was achieved in the classification of the severity of the damages.

Keywords: Convolutional neural network; Classification; Regression; Electrical mapping; Potential drop technique; Internal corrosion; Pitting corrosion.

LISTA DE FIGURAS

Figura 1 Algumas formas de corrosão 21
Figura 2 Algumas morfologias de corrosão puntiforme 22
Figura 3 Esquema básico para ensaios DCPD 23
Figura 4 - Típica junção de superfícies rugosas 24
Figura 5 Comparação densidades de corrente 25
Figura 6 Placa metálica simulada 27
Figura 7 Modelagem para placa defeituosa 29
Figura 8 Comparação do gradiente potencial elétrico 30
Figura 9 Esquema experimental montado 30
Figura 10 Comparação entre dados experimentais e simulados 31
Figura 11 Diagrama de desenvolvimento de um modelo tradicional de aprendizado de máquina 36
Figura 12 Exemplo de aplicação para regressão linear simples 38
Figura 13 Função sigmoide 38
Figura 14 Exemplo de classificação com k-NN 40
Figura 15 Hiperplano em duas dimensões 41
Figura 16 Exemplo de classificação com SVM 42
Figura 17 Exemplo transformação não linear em um conjunto de dados 43
Figura 18 Exemplo de regressão não linear com SVR 44
Figura 19 Árvore de decisão e o efeito do particionamento do conjunto de treinamento 45
Figura 20 Arquitetura típica de uma rede neural convolucional 49
Figura 21 Operação de convolução 51
Figura 22 Exemplo de aplicação do *padding* 52
Figura 23 Forma da função ReLU 53
Figura 24 Exemplos de *pooling* 54
Figura 25 Representação típica de conexões na camada totalmente conectada 55
Figura 26 Modelo proposto 56
Figura 27 Fluxo do processo de automatização das simulações 57
Figura 28 Comportamento do gradiente de potencial ao longo da linha dos eletrodos 57

Figura 29 Área utilizada para simulações 58
Figura 30 Curvas de isogradiente do potencial elétrico 59
Figura 31 Comparação em três dimensões do gradiente do potencial elétrico 59
Figura 32 Comparação mapeamento de gradiente do potencial elétrico e imagem elétrica para a placa com cinco alvéolos 60
Figura 33 Aplicação direta do limiar de Otsu 61
Figura 34 Fluxo do processamento de imagem 61
Figura 35 Limiar de Otsu como processo intermediário 62
Figura 36 Marcadores obtidos 62
Figura 37 Imagem elétrica processada 63
Figura 38 Ilustração de pontos de corte para extração de atributos do alvéolo de referência 64
Figura 39 Comparação das assinaturas de alvéolos com diferentes profundidades 65
Figura 40 Exemplo de imagem apresentada à rede convolucional 67
Figura 41 Arquitetura da rede Conv 2 67
Figura 42 Arquitetura da rede Conv 3 67
Figura 43 Distribuição dos alvéolos simulados e correspondentes regiões de influência obtidas 69
Figura 44 Resultados obtidos, sem normalização e sem seleção de atributos 72
Figura 45 Análise do erro com *Random Forest* sem seleção de atributos 72
Figura 46 Resultados obtidos, sem normalização com seleção de atributos 74
Figura 47 Análise do erro para *Gradient Boosting* e *Random Forest* 74
Figura 48 Resultados obtidos, com normalização sem seleção de atributos 75
Figura 49 Análise do erro com *Random Forest* sem seleção de atributos 76
Figura 50 Resultados obtidos, com normalização com seleção de atributos 77
Figura 51 Análise do erro com *Random Forest* com seleção de atributos 77
Figura 52 Ilustração para definição do Erro Crítico 79
Figura 53 Recall e acurácia, sem normalização sem seleção de atributos 80
Figura 54 Erro crítico, sem normalização sem seleção de atributos 80
Figura 55 Recall e acurácia, sem normalização com seleção de atributos 81
Figura 56 Erro crítico, sem normalização com seleção de atributos 81
Figura 57 Recall e acurácia, com normalização sem seleção de atributos 82

Figura 58 Erro crítico, com normalização sem seleção de atributos 83
Figura 59 Recall e acurácia, com normalização com seleção de atributos 84
Figura 60 Erro crítico, com normalização com seleção de atributos 84
Figura 61 Acurácias médias na validação 85
Figura 62 Recall e acurácia para o experimento com normalização e seleção de atributos e comparação com a melhor rede Conv 3 86
Figura 63 Erro crítico para o experimento com normalização e seleção de atributos e comparação com a melhor rede Conv 3 86
Figura 64 Recall e erro crítico, algoritmos tradicionais *versus* Conv 3 87
Figura 65 Erro crítico, algoritmos tradicionais *versus* Conv 3 87
Figura 66 Correlação linear dos atributos extraídos 97

LISTA DE TABELAS

Tabela 1 Comparação entre resultados modelado e experimental 31
Tabela 2 Descrição dos parâmetros de simulação 57
Tabela 3 Descrição do banco de dados 69
Tabela 4 Descrição de hiperparâmetros 70
Tabela 5 Desempenho dos regressores sem normalização e sem seleção de atributos 72
Tabela 6 Desempenho dos regressores sem normalização e com seleção de atributos 73
Tabela 7 Desempenho dos regressores com normalização sem seleção de atributos .. 75
Tabela 8 Desempenho dos regressores com normalização com seleção de atributos .. 76
Tabela 9 Desempenho dos regressores baseados em comitês de CARTs 77
Tabela 10 Desempenho dos classificadores sem normalização sem seleção de atributos 79
Tabela 11 Desempenho dos classificadores sem normalização com seleção de atributos 81
Tabela 12 Desempenho dos classificadores com normalização sem seleção de atributos 82
Tabela 13 Desempenho dos classificadores com normalização com seleção de atributos 83
Tabela 14 Modelos com maior acurácia média na validação 85
Tabela 15 Avaliação no teste para a rede Conv 3 85
Tabela 16 Modelo com maior desempenho individual 85
Tabela 17 Valores de hiperparâmetros utilizados na busca em grade para os regressores 97
Tabela 18 Valores de hiperparâmetros utilizados na busca em grade para os classificadores 98
Tabela 19 Valores de hiperparâmetros utilizados na busca em grade para as redes convolucionais 98

LISTA DE SIGLAS

ACPD	*Alternating Current Potential Drop*
CART	*Classification and Regression Trees*
CMY	Correlação Cooper-Mikic-Yovanovich
DC	*Direct Current*
DCPD	*Direct Current Potential Drop*
EA	Erro Absoluto
EDP	Equação Diferencial Parcial
END	Ensaio Não Destrutivo
EQM	Erro Quadrático Médio
FSM	*Field Signature Method*
ILSVRC	*ImageNet Large Scale Visual Recognition Challenge*
k-NN	*k-Nearest Neighbor*
MEF	Método dos Elementos Finitos
OG	Onda Guiada
PD	*Potential Drop*
PIG	*Pipeline Inspection Gizmo*
RE	Resistência Elétrica
RGB	*Red Green Blue*
RI	Região de Interesse
SRQ	Soma dos Resíduos ao Quadrado
SVM	*Support Vector Machine*
SVR	*Support Vector Regression*
VGG	*Very Deep Convolutional Networks for Large-Scale Image Recognition*
US	Ultrassom

SUMÁRIO

	INTRODUÇÃO	16
1	**CORROSÃO E A TÉCNICA *POTENTIAL DROP***	21
1.1	***Potential Drop***	23
1.1.1	*Direct Current Potential Drop* - DCPD	23
1.1.2	*Alternating Current Potential Drop* - ACPD	26
1.2	**Fundamentos de Modelagem em Elementos Finitos Aplicados à Técnica DCPD**	27
2	**TRABALHOS RELACIONADOS**	32
2.1	**Técnica *Potential Drop* e Modelagem em Elementos Finitos**	32
2.2	**Processamento de Imagens**	34
2.3	**Estimativa da Profundidade de Defeitos**	35
3	**APRENDIZADO DE MÁQUINAS E REDES NEURAIS CONVOLUCIONAIS**	36
3.1	**Regressões Linear e Logística**	37
3.2	***k-Nearest Neighbors***	39
3.3	***Support Vector Machine***	40
3.4	**Árvores de Decisão**	44
3.5	***Random Forest***	46
3.6	***Gradient Boosting***	47
3.7	**Redes Neurais Convolucionais**	48
3.7.1	Camada Convolucional	50
3.7.2	Camada de *Pooling*	53
3.7.3	Camada Totalmente Conectada	54
4	**MODELO PROPOSTO**	56
4.1	**Modelagem em Elementos Finitos**	56
4.2	**Processamento de Imagem**	58
4.2.1	**Extração de Atributos**	64
4.3	**Regressores e Classificadores**	66
4.4	**Redes Neurais Convolucionais**	66

5 **ESTUDO DE CASO** 69
5.1 **Estimativa da Profundidade dos Alvéolos** 70
5.1.1 Experimentos sem Normalização 71
5.1.2 Experimentos com Normalização 75
5.2 **Classificação da Gravidade dos Alvéolos** 78
5.2.1 Experimentos sem Normalização 79
5.2.2 Experimentos com Normalização 82
5.2.3 Experimentos com Redes Neurais Convolucionais 84
CONCLUSÃO 88
REFERÊNCIAS 91
APÊNDICE A 97

INTRODUÇÃO

A corrosão metálica pode ser encarada como um processo espontâneo que age constantemente sobre os materiais, podendo comprometê-los de tal forma que não sirvam mais aos propósitos à que se destinam. Estruturas de grande importância econômica estão sujeitas aos efeitos da corrosão como, por exemplo, oleodutos, gasodutos, adutoras, torres de transmissão de energia elétrica, instalações industriais, dentre outras (GENTIL, 2003). As perdas econômicas causadas pela corrosão podem ser diretas ou indiretas. As perdas diretas estão associadas à troca de equipamentos, componentes e estruturas corroídos. Já, as perdas indiretas podem atingir dimensões maiores, como a parada de uma planta industrial, perda de material ou contaminação (ambos em caso de transporte por dutos), podendo levar a desastres ambientais. Estima-se que os custos anuais para a indústria devido a corrosão cheguem ao expressivo valor de 2,5 trilhões de dólares, o que equivale a 3,4% do Produto Mundial Bruto (KOCH et al., 2016). Além das perdas financeiras, a corrosão pode levar à situações com ocorrência de acidentes sérios com potencial perda de vidas humanas. Sendo assim, a monitoração da corrosão, especialmente a que ocorre internamente em tubulações, constitui uma importante atividade para o gerenciamento da corrosão e da integridade de equipamentos. Através desta atividade é possível entender melhor o processo corrosivo e selecionar o método de controle mais adequado para cada sistema (AL-SHAMARI et al., 2014; ROBERGE, 2007).

Uma das etapas relevantes do monitoramento da corrosão interna é a de levantamento de taxas de corrosão, para determinação do potencial de corrosividade do fluido. Isto pode ser feito por meio de cupons e sondas, análises químicas do fluido e resíduos, dados de PIG (*Pipeline Inspection Gizmo*) instrumentados e histórico de falhas. Mais frequentemente são empregados cupons de perda de massa e sondas de resistência elétrica (RE), que são inseridos no interior do duto ou tubulação. Entretanto, essas técnicas apresentam algumas desvantagens. As sondas ER foram desenvolvidas com base em mecanismos de corrosão generalizados, ao invés de localizados (YANG, 2008). Mesmo os cupons de corrosão às vezes não são úteis para avaliar a corrosão localizada, porque eles são instalados em uma área mínima, considerando todo o comprimento e posição da tubulação pode não ser o mais apropriado. Estes métodos intrusivos necessitam de intervenções periódicas, seja para a troca do cupom de perda de massa e sua posterior

avaliação, ou para a substituição do elemento sensor da sonda de resistência elétrica. A periodicidade da troca do cupom e do elemento sensor da sonda RE varia de acordo com a corrosividade do fluido. Esta atividade é considerada crítica em unidades *off-shore* de produção de petróleo (NITG390, 2013; YANG, 2008). Devido ao elevado risco associado à atividade de troca de cupom e sondas intrusivas, além das dificuldades relacionados à logística, custo e treinamento de pessoal qualificado, os equipamentos não intrusivos vem se tornando mais uma alternativa para a determinação das taxas de corrosão (SILVA et al., 2017). Neste tipo de ensaio, também classificado como não destrutivos (END) não há danificação da peça em análise. Entretanto, a técnica da análise de cupons permite a avaliação de forma rápida e precisa de alterações nas características eletroquímicas de fluidos transportados em tubulações. Essas variações podem ser devidas ao próprio fluido ou à ações como a injeção de inibidores de corrosão, por exemplo.

A maioria dos equipamentos não intrusivos comerciais aplicados no monitoramento da corrosão interna possuem o princípio físico baseado em ultrassom (US). Feixes de ondas acústicas são introduzidos em um material para a detecção de descontinuidades superficiais e subsuperficiais. Essas ondas percorrem o material e sofrem alguma atenuação, então são captados os ecos e é feita a análise deles para definir e localizar descontinuidades. Medições ultrassônicas são afetadas por ruído durante estágios iniciais de processos corrosivos, mas podem dar resultados satisfatórios quando as dimensões das microfissuras aumentam (SPOSITO et al., 2010). Uma variação dos métodos baseados em ultrassom, os equipamentos de onda guiada (OG) possuem longo alcance, entretanto apresentam resolução inferior à fornecida pelos dispositivos US tradicionais, que por sua vez podem monitorar apenas um único ponto de cerca de 1 cm^2 de área circular. Um sistema de ondas guiadas usando transdutores eletromagnético e tomografia digital tem sido usado em campo, conforme relatado por (SIMONETTI et al., 2015). Esta solução traz como principal vantagem da tecnologia OG, como o monitoramento de longo alcance, com processamento específico de dados para melhorar sua resolução. Porém, a melhor resolução desse tipo de sistema atinge 0,1 mm e ainda é inferior às técnicas convencionais de sondas RE e US.

Métodos magnéticos não intrusivos também são empregados no acompanhamento de processos corrosivos em tubulações. Na técnica de campo magnético de fuga, aplicável apenas à materiais ferromagnéticos, que consiste na magnetização externa do material

em análise, a maior parte das linhas de fluxo magnético passará pelo material quando não há defeitos. Na presença de regiões defeituosas, onde a permeabilidade magnética decai, ocorre distorção do campo magnético. O que resulta na dobra de linhas de campo magnético que "vazam"para fora da superfície do material. Esta fuga, que costuma ser proporcional às dimensões dos defeitos encontrados, pode ser detectada por sensores de Efeito Hall (SHI et al., 2015; AMARAL et al., 2003).

Outra aplicação magnética é o uso de correntes parasitas (conhecidas também como *eddy currents*, do inglês). Uma bobina, principal elemento desta técnica, é excitada com corrente alternada e posicionada sobre a região inspecionada. A impedância da bobina é alterada segundo a reação do campo induzido. De acordo com a Lei de Lenz, um campo magnético induz (em um condutor elétrico) uma corrente na direção oposta àquela geradora do campo primário. Esta corrente induzida (parasita), por sua vez, leva a criação de um campo magnético induzido na bobina. Quando há a sobreposição dos campos primário e induzido, ocorre redução do fluxo magnético total ao qual a bobina está exposta, o que ocasiona diminuição na indutância da mesma se o material em análise não for ferromagnético (UDPA; MOORE, 2004). Ao passo que a indutância da bobina diminui, sua resistência aumenta. Isto ocorre devido as perdas que ocorrem no material em análise. Essas perdas são supridas pela fonte de energia primária. No caso de materiais ferromagnéticos, ocorre o aumento da indutância atribuído a maior permeabilidade magnética do material, pois ocorre o fenômeno conhecido como indutância mútua. Também ocorre aumento da resistência neste caso, por conta das perdas por correntes parasitas e histerese. A presença de descontinuidades na peça em análise provoca tanto redução como redistribuição das correntes parasitas e, consequentemente, na impedância da bobina (UDPA; MOORE, 2004).

Porém, há sistemas que se baseiam no método do campo elétrico como a técnica de *Potential Drop* (PD) que, de uma forma geral, consiste na aplicação de corrente elétrica e observação do potencial elétrico resultante na peça em análise. Tem sido usada na indústria para o monitoramento da corrosão interna, através da medição da perda de espessura. Outra aplicação comum desta técnica é a estimativa da profundidade de trincas em dutos, tubulações e equipamentos de processo. Também é empregada na realização de ensaios de laboratório para avaliar a propagação de trincas. Estes ensaios, geralmente, utilizam a técnica para estimar a velocidade de propagação de trincas para a determinação

da tenacidade à fratura de um material (YANG, 2008). Diferentemente das técnicas ultrassônicas e magnéticas, que permitem avaliações apenas durante a manutenção periódica de equipamentos, PD possibilita o monitoramento contínuo (*on-line*) de estruturas (RYU et al., 2020; SPOSITO et al., 2010).

Apesar de ser uma das técnicas não destrutivas mais antigas para o monitoramento da propagação de trincas e medição de espessura, poucos resultados quantitativos são apresentados na literatura, devido a necessidade de análise matemática ou de métodos computacionais. Alguns trabalhos, como (CORCORAN et al., 2020; SPOSITO; CAWLEY; NAGY, 2010; SPOSITO, 2009) mostram, através da modelagem em elementos finitos, que existe variação do potencial elétrico em torno de defeitos causados por corrosão em placas metálicas. A partir da medição do campo potencial na superfície da placa é possível ter uma noção das dimensões de defeitos existentes. Segundo (SPOSITO et al., 2010), existem indicações de que a técnica *Potential Drop* pode ser capaz de detectar danos em estágios anteriores do que outros métodos convencionais. Um passo desafiador seria a partir do mapeamento do potencial elétrico inferir as dimensões de defeitos e classificá-los de acordo com a sua gravidade.

No texto de (HOANG; DUC, 2019), embora seja tratado um problema de corrosão externa, são aplicadas técnicas de processamento de imagens com o objetivo de obter um extrator de características dos exemplos analisados. Seguindo esta linha, o mapeamento de potencial elétrico pode ser interpretado como uma "imagem elétrica", o que possibilita o emprego de uma abordagem de extração de atributos baseada em processamento de imagens para posterior apresentação a algoritmos regressores e classificadores. Além disso, a utilização de imagens permite aplicar redes neurais convolucionais para classificar defeitos de acordo com a sua gravidade. Atualmente, as redes neurais convolucionais vem tendo aplicações em diversas tarefas de aprendizado de máquina como detecção de objetos, segmentação, classificação, reconhecimento biométrico, veículos autônomos, aeronaves não tripuladas, diagnóstico médico dentre outras (KHAN et al., 2020; VALUEVA et al., 2020). Conforme escrito por (JAFARZADEH; CHEN; BOBARU, 2019), com os recentes avanços no campo do aprendizado de máquina, a utilização dessas ferramentas na extensão do alcance da modelagem de corrosão a fim de melhorar a previsibilidade, confiabilidade e universalidade dos modelos de corrosão é uma questão que permanece em aberto.

Na literatura, como discutido no Capítulo 2, há trabalhos que utilizam métodos de processamento de imagens ligados à necessidade de fotografias apenas para indicação da existência ou não de corrosão. Também há casos onde a caracterização de defeitos é baseada em expressões analíticas/aproximadas, por vezes restritas, o que prejudica a abrangência dessas técnicas. Assim sendo e observando a lacuna existente no setor de monitoramento *on-line* de corrosão interna do tipo localizada e morfologia alveolar, o objetivo deste trabalho é contribuir nesta área a partir da aplicação da técnica *Potential Drop* para obter o mapeamento do gradiente de potencial elétrico na superfície de uma placa de aço inox AISI-304. Adicionalmente, são empregadas técnicas de processamento de imagem e regressão para estimar a profundidade de defeitos. Também são utilizados classificadores tradicionais e redes neurais convolucionais, a fim de obter a identificação da severidade dos defeitos encontrados em termos da área de abertura.

Este documento está organizado da seguinte forma. No Capítulo 1 são descritas as principais formas de corrosão e também são apresentados os aspectos fundamentais da técnica *Potential Drop* em conjunto com sua modelagem via método dos elementos finitos. Trabalhos relacionados com esta dissertação, seja pela técnica *Potential Drop*, pelo uso de algoritmos de aprendizado de máquina, ou por modelagem em elementos finitos são mostrados no Capítulo 2. Já, no Capítulo 3 são mostrados os algoritmos clássicos de aprendizados de máquina utilizados e redes neurais convolucionais. O Capítulo 4 apresenta o modelo proposto, ao passo que no Capítulo 5 são descritos o processo de levantamento da base de dados utilizada, os experimentos realizados e, por fim, traz os resultados obtidos.

1 CORROSÃO E A TÉCNICA *POTENTIAL DROP*

Há diferentes tipos de corrosão que podem ser classificados considerando a aparência ou forma de ataque, bem como causas da corrosão e seus mecanismos. Dessa forma, pode-se especificar a corrosão segundo:

- a morfologia: uniforme, por placas, alveolar, puntiforme ou por pite, empolamento por hidrogênio, em torno de solda, filiforme, por esfoliação, grafítica dentre outras;
- os fatores mecânicos: sob tensão e/ou fadiga, por atrito, associada à erosão;
- o meio corrosivo: atmosférica, pelo solo, pela água do mar, pela presença de microrganismos, etc.

A análise da corrosão conforme a morfologia já traz grande auxílio para o esclarecimento do seu mecanismo de ação e na aplicação de medidas adequadas de proteção, conforme (GENTIL, 2003). Na Figura 1 são mostrados exemplos de corrosão e a classificação de acordo com sua forma.

Figura 1: Algumas formas de corrosão
Fonte: adaptado de GENTIL, 2003.

A corrosão uniforme ocorre em toda a extensão da superfície, causando perda uniforme de espessura de material, enquanto a corrosão alveolar e puntiforme são fenômenos

que ocorrem em pontos ou pequenas áreas localizadas da superfície metálica. No caso da corrosão alveolar, ocorre a formação de sulcos que apresentam fundo arredondado e profundidade geralmente menor que seu diâmetro. Já na corrosão puntiforme há a produção de cavidades de fundo anguloso e profundidade geralmente maior que seu diâmetro (GENTIL, 2003). Entretanto, na norma ASTM G46-94 (ASTM, 2018) por exemplo, não há distinção entre os termos *alveolar* e *pites*, sendo ambos tratados como pites (do inglês, *pitting*), conforme pode ser observado no trecho da norma mostrado na Figura 2. Então, ficam adotados neste trabalho defeitos localizados de fundos arredondados e profundidades menores que seus diâmetros, seguindo nomenclatura de *alvéolo*.

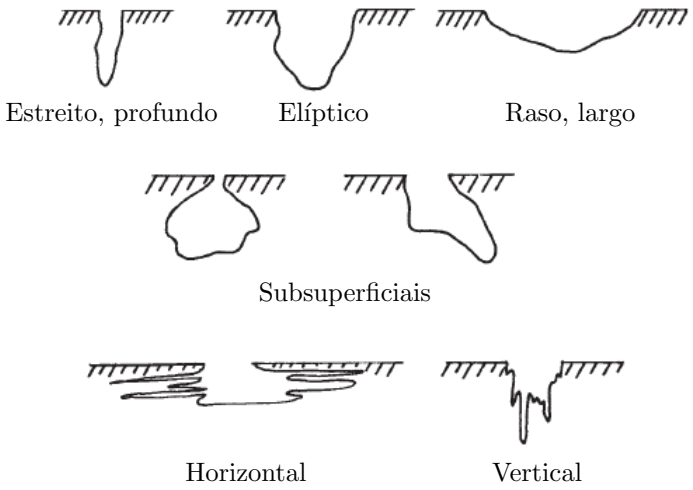


Figura 2: Algumas morfologias de corrosão puntiforme
Fonte: adaptado de ASTM, 2018.

Segundo (SCHWEITZER, 2010), é comum que a abertura do alvéolo seja recoberta pelo material resultante da corrosão, o que dificulta sua detecção durante inspeções visuais. Os alvéolos podem causar falhas estruturais devido à fragilização da estrutura, contribuindo para fenômenos como a trinca por exemplo.

A norma ASTM G46-94 (ASTM, 2018) define padrões de classificação de corrosão localizada considerando três aspectos: densidade do número de alvéolos/pites por área, área de abertura dos danos e profundidade dos defeitos. Neste trabalho é utilizada a classificação com base na área de abertura, isto engloba defeitos com área de $2,0\,mm^2$ à $24,5\,mm^2$, o que equivale afirmar que são observados danos das classes B2 à B5.

1.1 ***Potential Drop***

Neste item são apresentadas as duas formas básicas de utilização da técnica *Potential Drop*, que são com corrente contínua e corrente alternada, bem como suas principais diferenças.

1.1.1 *Direct Current Potential Drop* - DCPD

A utilização do DCPD para monitoração da origem de defeitos e sua evolução ao longo do tempo em laboratórios remonta os anos 1950 (NONDESTRUCTIVE..., 1959). Graças aos bons resultados que esta técnica proporciona, nos dias atuais possui aplicação na detecção de trincas sob fadiga, fragilização por hidrogênio e corrosão (MITCHELL; LINK; LJUSTELL, 2011). A montagem de um ensaio em DCPD utiliza o esquema a quatro fios. Conforme Figura 3, os eletrodos 1 e 4 são responsáveis por transportar a corrente contínua I que é injetada na amostra enquanto os eletrodos 2 e 3 colhem a diferença de potencial V. Note que o eletrodo 2 é móvel, o que possibilita realizar um mapeamento do potencial elétrico em parte da superfície da placa (PINHEIRO et al., 2019).

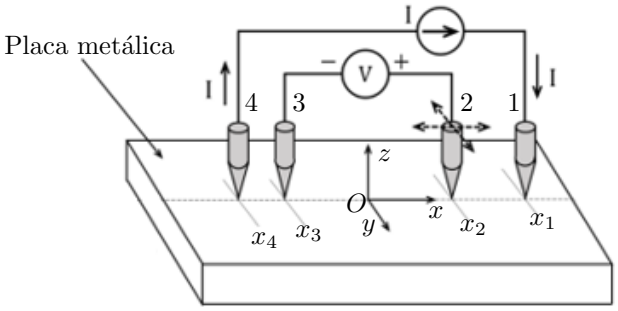


Figura 3: Esquema básico para ensaios DCPD
Fonte: adaptado de PINHEIRO et al., 2019.

A corrente injetada é constante e pode chegar a 50A em alguns equipamentos comerciais. Nas regiões em que há falhas na amostra analisada, a seção transversal é diminuída, com isso a resistência elétrica aumenta nesta região, causando aumento no potencial. Assim, em função da resistência elétrica é possível a obtenção de um mapeamento da peça e identificar a região onde está localizado o dano. Entretanto, deve-se lembrar que está sendo feita a medição de potencial elétrico em um objeto metálico, portanto um material de baixa resistividade. Isto causa leituras da ordem de microvolt, mesmo com

a utilização de correntes elevadas. Por conta deste baixo nível de tensão, pode ocorrer interferência da tensão termoelétrica nas medições, devida à junção elétrica de materiais diferentes (fios, eletrodos e amostra). Outro aspecto importante é a resistência de contato entre os eletrodos injetores de corrente e o objeto em análise. Altas resistências de contato ocasionarão elevadas perdas por efeito Joule, e, como a resistividade elétrica é dependente da temperatura, haverá influência deste fator no ensaio DCPD.

Segundo (YOVANOVICH, 2005; YOVANOVICH; MAROTTA, 2003) quando há união entre dois sólidos formam-se junções imperfeitas, isto deve-se ao fato de os materiais não serem perfeitamente lisos e planos. Então formam-se junções mecânicas que consistem em numerosos micro contatos discretos dependentes da aspereza das superfícies. Na Figura 4 esta situação microscópica é ilustrada, a partir das asperezas individuais é possível estabelecer uma aspereza efetiva α.

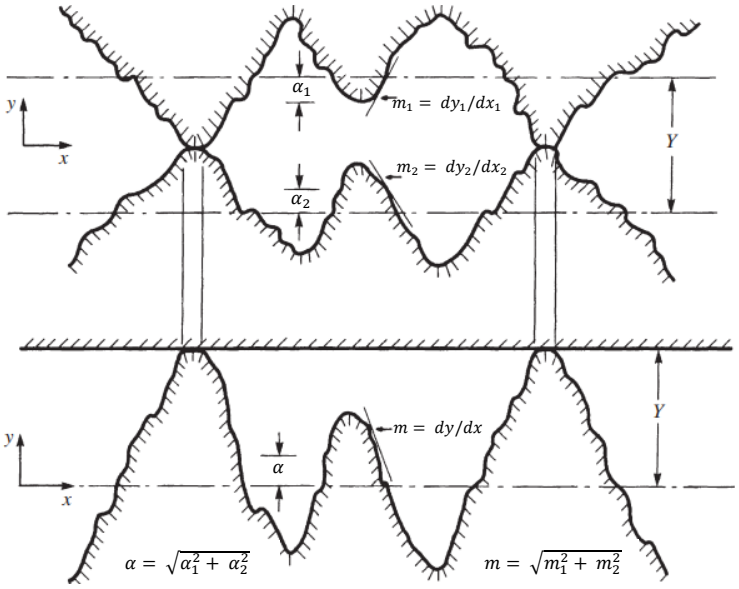


Figura 4: - Típica junção de superfícies rugosas
Fonte: adaptado de HEGAZY, 1985.

No *software* COMSOL®, que é utilizado neste trabalho, o contato elétrico é modelado levando em conta a condutância conjunta das duas superfícies em contato (entre placa e eletrodos). As densidades de corrente dos objetos em contato dependem das diferenças de potencial entre as áreas de contato, conforme equações 1 e 2; onde $\boldsymbol{n}$ é o vetor normal à superfície, $\mathbf{J}$ se refere à densidade de corrente e V o potencial elétrico. Os índices dizem respeito à respectiva superfície (COMSOL..., 2018).

$$\boldsymbol{n}\cdot\mathbf{J_1} = -h_c(V_1 - V_2) \qquad [A/m^2] \tag{1}$$

$$\boldsymbol{n}\cdot\mathbf{J_2} = -h_c(V_2 - V_1) \qquad [A/m^2] \tag{2}$$

A condutância h_c é calculada a partir do modelo da correlação Cooper-Mikic-Yovanovich (CMY) e válida para superfícies rugosas isotrópicas, foi formulada usando um modelo que assume a deformação plástica das asperezas superficiais (YOVANOVICH; MAROTTA, 2003). A Equação 3 mostra como a CMY relaciona h_c às asperezas e a pressão na área de contato.

$$h_c = 1,25\sigma_c \frac{m}{\alpha}\left(\frac{P}{H_c}\right)^{0,95} \qquad [S/m^2] \tag{3}$$

Onde H_c [Pa] é a microdureza do material mais macio e P [Pa] é a pressão de contato; a condutividade resultante σ_c é definida como a média harmônica das condutividades das superfícies de contato, como mostrado na Equação 4.

$$\sigma_c = \frac{2\sigma_1\sigma_2}{\sigma_1 + \sigma_2} \qquad [S/m] \tag{4}$$

Na Figura 5 é mostrada a comparação ilustrativa do comportamento esperado em DCPD para uma placa metálica de espessura T submetida à corrente elétrica I, através dos eletrodos 1 e 4. Em (a) é apresentada a placa sem defeito, com densidade de corrente $\mathbf{J_S}$ e em (b) é representada uma placa de mesmas dimensões, porém nesta situação é simulado um defeito semiesférico de abertura a e profundidade t. A densidade de corrente é $\mathbf{J_D}$ neste caso. O defeito produz um adensamento da distribuição de corrente no interior da placa, ocasionando consequentemente a variação do potencial elétrico superficial medido através dos eletrodos 3 e 4.

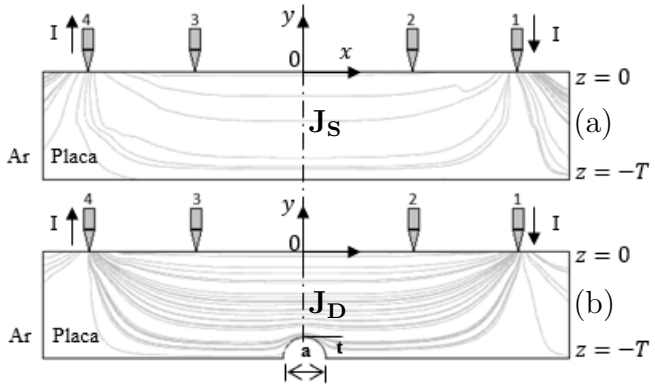


Figura 5: Comparação densidades de corrente
Legenda: (a) placa sem defeito, com densidade de corrente $\mathbf{J_S}$
(b) placa com defeito, com densidade de corrente $\mathbf{J_D}$

1.1.2 *Alternating Current Potential Drop* - ACPD

Neste trabalho a concentração é no uso de corrente contínua, entretanto é relevante apresentar o emprego de corrente alternada. A técnica ACPD é uma derivação da DCPD, o esquema de ensaio é o mesmo mostrado na Figura 3. Nela, a corrente injetada é alternada, esta simples mudança causa uma série de efeitos no sistema como um todo. Com o uso de corrente alternada surge a figura do efeito pelicular, fenômeno no qual a corrente tende a distribuir-se na superfície do condutor. A chamada profundidade de penetração, descrita pela Equação 5, indica o quão abaixo da superfície a corrente flui.

$$\delta = \frac{1}{\sqrt{f\pi\mu\sigma}} \qquad [m] \tag{5}$$

Onde:

f – frequência de operação [Hz];

μ – permeabilidade magnética do material [H/m];

σ – condutividade elétrica do material [S/m].

A utilização de corrente alternada faz com que não se pense mais apenas em termos da resistência elétrica da amostra, mas sim da sua impedância. Assim a impedância "vista pela fonte de corrente"é maior que a resistência isoladamente no caso da DCPD, com isso ocorre redução da magnitude elétrica necessária para a aplicação da ACPD em comparação ao método que usa corrente contínua. Tipicamente a corrente empregada é inferior a 1A (SPOSITO, 2009). A redução da corrente aplicada em relação ao DCPD também se dá pela utilização de amplificadores *lock-in* na medição do potencial elétrico, este dispositivo baseado na detecção sensível à fase possibilita a separação da componente de sinal em uma frequência específica, obtendo ganhos na relação sinal-ruído.

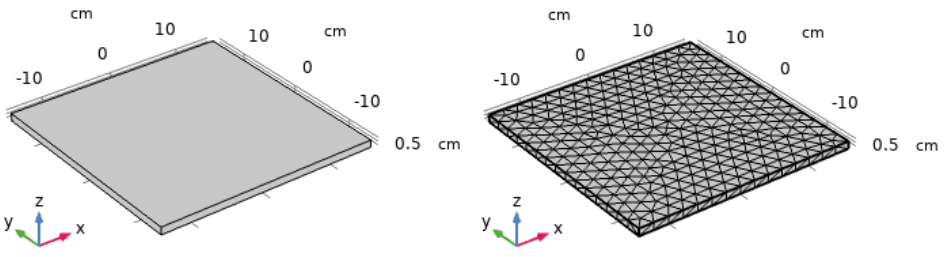
ACPD pode ser subdividida em duas classes: alta frequência e baixa frequência. Como a profundidade de penetração depende da permeabilidade magnética do material em questão, materiais ferromagnéticos possuem profundidade de penetração menor, então o efeito pelicular tem maior impacto. Um exemplo de material neste caso é o aço carbono, assim a utilização de ACPD de alta frequência pode ser considerada. Já para o aço inox AISI-304, por exemplo, que é não ferromagnético (permeabilidade magnética relativa unitária) o efeito pelicular pode ser desprezado, deve-se utilizar baixas frequências, regime também denominado quasi-DC (CORCORAN et al., 2020; TARNOWSKI, 2016).

1.2 Fundamentos de Modelagem em Elementos Finitos Aplicados à Técnica DCPD

Fenômenos da natureza nas mais diversas áreas são precisamente descritos por meio de leis físicas, com o emprego de diferentes campos da matemática como álgebra e equações diferenciais ou integrais, por exemplo. Para se obter boas representações do modelo físico estudado é preciso a utilização de uma ferramenta matemática adequada ao problema. Concentrando nos problemas descritos em termos do cálculo diferencial, grande parte deles são expressos via equações diferenciais parciais (EDP) (REDDY, 2006).

As soluções das EDPs para problemas e/ou geometrias complexos podem ser impossíveis de se atingir de forma exata, por isso são comumente obtidas de forma aproximada. Aí surge uma potencial possibilidade de uso do Método dos Elementos Finitos (MEF). Há décadas é utilizada a modelagem DCPD via Método dos Elementos Finitos, exemplos de trabalhos que fazem essa união são (RITCHIE; J., 1979; GANDOSSI et al., 2001).

No MEF, a geometria complexa do problema é representada como um conjunto de subdomínios geométricos simples, os ditos *elementos finitos* que normalmente são triangulares ou tetraédricos (no caso de três dimensões). Então, a geometria original passa a ser representada por uma malha (*mesh*) dos elementos finitos. Sobre cada elemento finito é aplicada a função que se deseja encontrar solução, usando a ideia básica de aproximação, na qual funções contínuas podem ser representadas por uma combinação linear polinomial. Os elementos finitos são interligados por meios de nós, os valores desses nós são obtidos pela satisfação da equação que rege o problema em questão (REDDY, 2006). Na Figura 6 temos a ilustração de uma placa metálica com sua representação antes e depois de sua discretização (geração da malha).



(a) placa antes de aplicar a malha (b) placa após discretização

Figura 6: Placa metálica simulada

O eletromagnetismo nos fornece duas equações fundamentais para a modelagem de campo elétrico e corrente estáticos: Equação da continuidade (6) e a Lei de Ohm na forma pontual (7) (HAYT; BUCK, 2013; COMSOL. . . , 2018), onde σ é a condutividade elétrica do material.

$$\nabla \cdot \mathbf{J} = 0 \tag{6}$$

$$\mathbf{J} = \sigma \mathbf{E} \qquad [A/m^2] \tag{7}$$

Pela Equação da continuidade obtém-se a densidade de corrente **J**, a partir daí, manipulando a Lei de Ohm na forma pontual temos uma expressão que relaciona o campo elétrico **E** com a densidade de corrente. Isto é mostrado na Equação 8.

$$\mathbf{E} = \frac{1}{\sigma}\mathbf{J} \qquad [V/m] \tag{8}$$

O termo $1/\sigma$ é comumente substituído por ρ, denotando a resistividade elétrica do material. A diferença de potencial (V) entre dois pontos (A e B) é definida como o trabalho necessário de uma fonte externa para mover uma carga de um ponto a outro em um campo elétrico, escrevendo este enunciado de forma matemática na Equação 9, temos:

$$V = -\int_B^A \mathbf{E} \cdot d\mathbf{L} = -\int_B^A \frac{1}{\sigma}\mathbf{J} \cdot d\mathbf{L} \qquad [V] \tag{9}$$

O potencial é um campo escalar e depende do caminho de integração **L**, da condutividade elétrica e da geometria do material; em coordenadas cartesianas então, o potencial é função de x, y e z. Desse modo, podemos tomar sua diferencial total:

$$dV = \frac{\partial V}{\partial x}dx + \frac{\partial V}{\partial y}dy + \frac{\partial V}{\partial z}dz \tag{10}$$

Por outro lado, temos que:

$$dV = -\mathbf{E} \cdot d\mathbf{L} = -(E_x dx + E_y dy + E_z dz) \tag{11}$$

E pela simples comparação concluímos:

$$\begin{aligned} \frac{\partial V}{\partial x} &= -E_x \\ \frac{\partial V}{\partial y} &= -E_y \\ \frac{\partial V}{\partial z} &= -E_z \end{aligned} \tag{12}$$

Então podemos escrever de forma compacta a expressão do gradiente do potencial em função do campo elétrico, mostrada na Equação 13.

$$\nabla V = -\mathbf{E} \qquad [V/m] \tag{13}$$

E assim, é possível relacionar o gradiente de potencial com a densidade de corrente, conforme Equação 14. Em outras palavras, a partir da observação do gradiente de potencial, podemos inferir o efeito da presença de defeitos sobre a densidade de corrente.

$$\nabla V = -\frac{1}{\sigma}\mathbf{J} \qquad [V/m] \tag{14}$$

Para fins de exemplificação e comprovação da aplicabilidade da técnica MEF neste trabalho, foi modelada uma placa de aço inox AISI-304 quadrada de 300 mm por 10 mm de espessura. O defeito utilizado consiste em um quadrado no centro da superfície inferior da placa de 40 mm por 3 mm de profundidade, a Figura 7 ilustra isto, com eletrodos injetores (indicados por I^+ e I^-) localizados nas coordenadas $(\pm 12, 0, 0)\, cm$ e corrente de 10A. Já o aparato experimental foi montado semelhante ao mostrado na Figura 3 e é mostrado na Figura 9. Note que os eletrodos de tensão foram suprimidos, isto é possível na simulação, pois o *sofwtare* utilizado permite obter diretamente o mapeamento do gradiente de potencial.

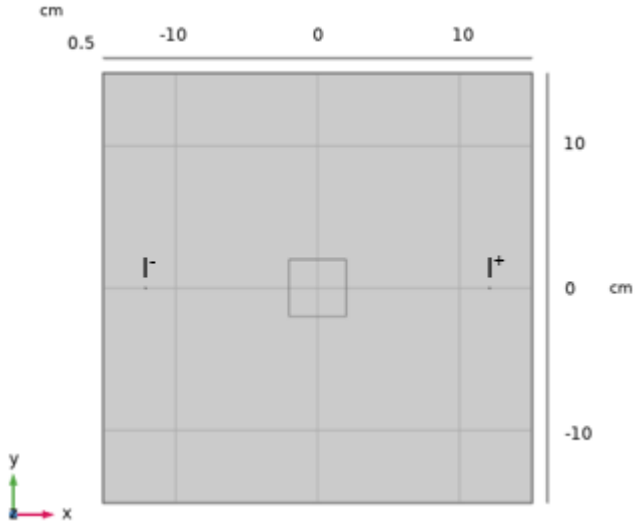
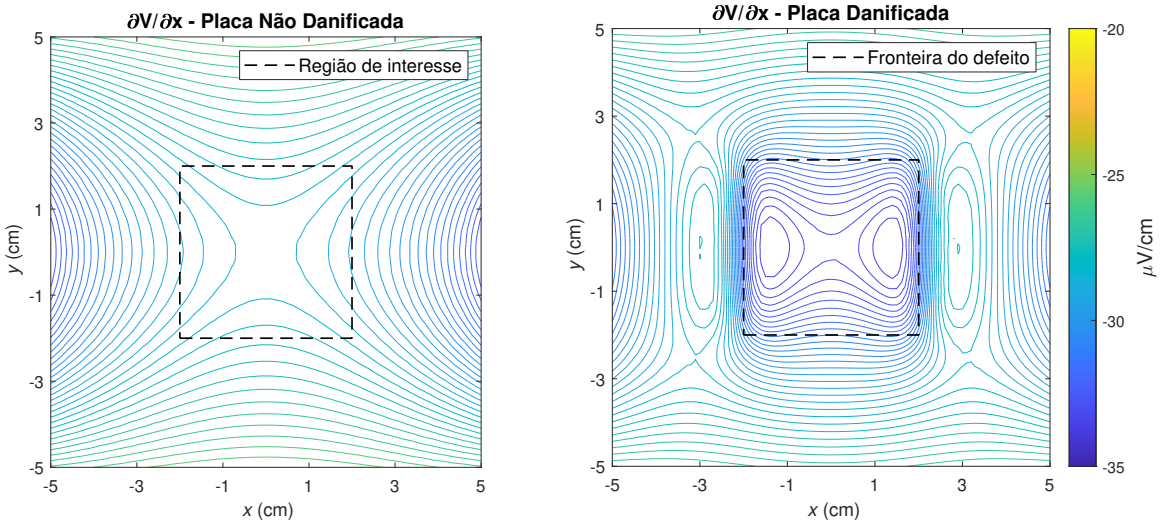


Figura 7: Modelagem para placa defeituosa
Nota: Vista superior da peça modelada com efeito de transparência aplicado de modo a possibilitar a visualização do defeito quadrado na superfície inferior.

Os mapeamentos simulados da componente x do gradiente de potencial superficial da placa sem defeito mostrada na Figura 6 e da placa defeituosa são ilustrados na Figura

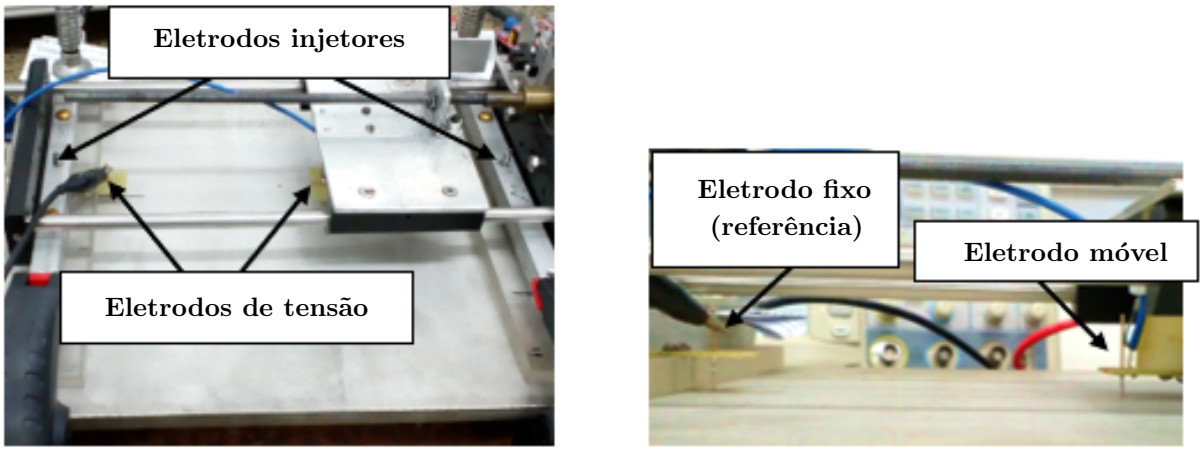
8. Na Figura 8 (a) é apresentada a situação da placa sem defeito (Figura 6), enquanto na Figura 8 (b) tem-se a presença do defeito quadrado, conforme indicações. Tanto na vizinhança do dano como na sua região interior ocorre uma perturbação do gradiente de potencial quando comparado à região equivalente da placa não defeituosa.



(a) placa sem defeitos (b) placa com defeito quadrado

Figura 8: Comparação do gradiente potencial elétrico

Utilizando o esquema de eletrodo móvel proposto por (PINHEIRO et al., 2019), optou-se por avaliar apenas a componente x do gradiente do potencial ao longo da linha onde os eletrodos estão dispostos, uma vez que a aplicação da corrente se dá ao longo do eixo x e o ponto de interesse é o centro da placa $(x, y) = (0, 0)$. As dimensões tanto da placa como do defeito são as mesmas que as usadas na modelagem, bem como o posicionamento dos eletrodos, corrente injetada e material da placa.



(a) visão geral (b) detalhe dos eletrodos de tensão

Figura 9: Esquema experimental montado

O deslocamento do eletrodo móvel ao longo do eixo x foi realizado com passo de $1\,mm$ no intervalo $-35\,mm \leq x \leq 35\,mm$. A diferença de potencial em cada ponto foi medida 30 vezes, o que resulta na aquisição 2.130 valores, assim cada ponto resultante nas curvas comparativas de potencial entre modelagem e experimento na Figura 10, é resultado do respectivo valor médio observado. A componente x do gradiente de potencial em $(x, y) = (0, 0)$ foi numericamente calculada como sendo os coeficientes de inclinação das curvas de potencial elétrico neste ponto. Os valores encontrados são mostrados na Tabela 1, veja que a diferença percentual dos resultados obtidos entre a modelagem e a montagem experimental foi de 0,96%, indicando assim a validade da modelagem utilizada. Em outras palavras, este resultado confirma a simulação via MEF como meio válido para obtenção de quantidade significativa de exemplos que podem ser apresentados à técnicas de aprendizado de máquinas, superando a escassez de amostras reais.

Tabela 1: Comparação entre resultados modelado e experimental

Placa com defeito	$\partial \mathbf{V}/\partial \mathbf{x}\vert_{\mathbf{x,y=0}}$ $(\mu V/cm)$
Modelagem	-33,21
Experimental	-33,53

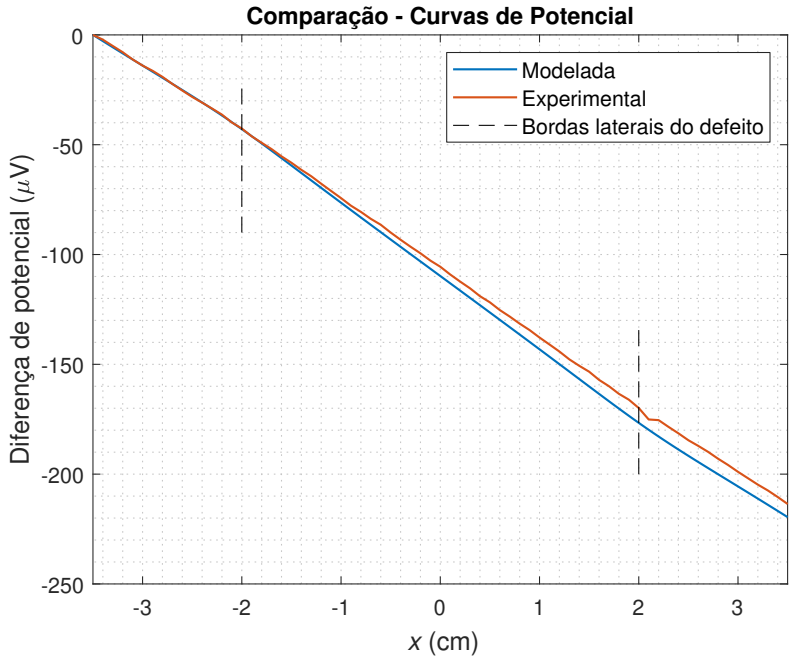


Figura 10: Comparação entre dados experimentais e simulados

2 TRABALHOS RELACIONADOS

O crescente aumento do poder computacional e da capacidade de realizar simulações cada vez mais complexas alteraram o panorama citado por (SPOSITO, 2009) que relatava uma escassez de trabalhos na área de monitoramento de corrosão interna utilizando a técnica *Potential Drop*. Agora é possível observar na literatura uma quantidade maior de trabalhos neste tópico.

Vamos também apresentar exemplos de trabalhos que utilizam os classificadores mais tradicionais do aprendizado de máquinas para explorar informações acerca da corrosão experimentada por materiais. Aqui também se faz necessário apresentar literatura relacionada ao processamento de imagens e redes convolucionais para embasar o modelo proposto nesta dissertação

Portanto, na revisão bibliográfica realizada ao longo deste trabalho, pode-se considerar que a intercessão dele com a literatura disponível se dá em três eixos principais: Técnica *Potential Drop* e Modelagem em Elementos Finitos; Processamento de Imagens (incluindo redes neurais convolucionais) e Estimativa da Profundidade de danos causados pela corrosão localizada.

2.1 Técnica *Potential Drop* e Modelagem em Elementos Finitos

Em (RYU et al., 2020) foi feito o estudo sobre a perda de massa em tubulações de aço carbono devido ao fenômeno conhecido como corrosão acelerada por fluxo. Este fenômeno encontra condições de ocorrência em plantas industriais nucleares e petrolíferas. No caso deste trabalho, a abordagem foi feita em plantas nucleares e envolveu a técnica *potential drop* com corrente contínua, modelagem em elementos finitos e uma abordagem elementar de *Machine Learning*. O sinal obtido via *potential drop*, indica mudanças na resistência elétrica, e é utilizado como atributo em uma regressão linear que tem como objetivo prever a curva do sinal DCPD, visto que cada forma de dano possui um curva característica associada. Os sinais DCPD utilizados foram obtidos através de simulações em elementos finitos. A perda de massa é modelada como uma região de tubos de aço carbono internamente desbastadas e a referência é um tubo modelado sem o desbaste. Supondo que cada valor de diferença de potencial tenha informações sobre a profundidade e largura do desbaste na posição, um total de 1024 valores de DCPD foram obtidos em

64 pontos de medição dispostos no sentido circunferencial do tubo, levando em conta 16 formas de defeitos. Empregando a corrente de 10A à 10 cm de distância do ponto de medição, visando mitigar a influência da vizinhança dos eletrodos injetores, onde a corrente não está uniformemente distribuída.

(CORCORAN et al., 2020) apresenta uma comparação entre sistemas DCPD, ACPD e *quasi*-DC utilizando placas de aço S275. A modelagem em elementos finitos permite aos autores avaliar a profundidade de penetração em diferentes frequências de operação. Com validação por meio de experimentos práticos, os autores afirmam que em frequências suficientemente baixas, o efeito pelicular é desprezível e, portanto, as medições se comportam efetivamente como em corrente contínua, portanto, menos sensíveis à permeabilidade magnética e são estáveis em materiais ferromagnéticos.

O método de assinatura de campo (FSM - do inglês, *Field Signature Method*) discutido por (GAN et al., 2016) é uma variação da técnica *potential drop* e tem sido aplicado ao monitoramento *on-line* da corrosão de tubos de metálicos. Para o emprego desta estratégia de monitoramento são soldados na tubulação eletrodos de medição de modo a formar uma matriz, podendo ser utilizada tanto para observação de corrosão uniforme como localizada. Dado que os pontos da matriz de medição são modelados como resistores, há diferentes percursos possíveis para corrente elétrica. Para ilustrar a distribuição da densidade de corrente neste tipo de situação os autores realizam abordagem com modelagem em elementos finitos.

Em (DOREMUS et al., 2015) o método DCPD é utilizado para monitorar a iniciação e propagação de uma trinca que se inicia a partir de anomalias de superfície durante um teste de fadiga em estruturas de motores de aeronaves. A diferença de potencial medida é dividida por um potencial de referência, com isso é obtido uma normalização que elimina os efeitos das propriedades dos materiais e do valor da corrente de entrada. A modificação da condutividade elétrica do material devido à alta temperatura também é explicada pela diferença de potencial normalizada. Estas curvas normalizadas são obtidas por meio da modelagem em elementos finitos e são utilizadas para obter informações sobre a profundidade das trincas. Com testes experimentais os autores alcançaram redução do erro 8,2% para 6% na estimativa da profundidade das trincas, em comparação com métodos anteriores. Este erro é definido como a diferença entre os resultados simulados e as medidas experimentais.

Outro trabalho que relaciona *Potential Drop* e modelagem em elementos finitos, (SPOSITO; CAWLEY; NAGY, 2010), com o propósito de determinar a máxima profundidade de defeito. Neste caso foi usado ACPD de baixa frequência (10 Hz). O estudo baseou-se em simulações de placas de aço inox AISI-304 quadradas de 360 mm por 10 mm de espessura, variando as dimensões dos defeitos, que eram quadrados e centralizados na superfície inferior das placas. A modelagem em elementos finitos validou a obtenção de uma expressão analítica para o cálculo da espessura remanescente. Também foi possível obter um mapeamento de corrosão tanto na modelagem quanto experimentalmente utilizando a fórmula obtida. O defeito artificialmente criado possui a profundidade de 30% da espessura da placa e a profundidade máxima estimada foi 28,5% e 28,1% para a simulação em elementos finitos e dados experimentais respectivamente.

2.2 **Processamento de Imagens**

Com o propósito de classificar regiões externas como corroída ou não corroída, em (ATHA; JAHANSHAHI, 2017) foram testadas as redes pré treinadas VGG16 (SIMONYAN; ZISSERMAN, 2014) e ZF Net (ZEILER; FERGUS, 2013), também foram avaliados redes próprias menores e com treinamento mais rápido. Foi realizada uma abordagem investigativa de espaço de cores como alternativa ao tradicional RGB. A rede VGG16 com retreinamento da camada totalmente conectada foi a que se mostrou mais robusta e isto foi atribuído às características de suas camadas de convolução e não ao aumento no tamanho da camada totalmente conectada, apresentando nas métricas *Recall*, *Precison* e *F1-score* valores superiores a 96%.

Em outra abordagem, (Vriesman et al., 2019), a aplicação de redes neurais convolucionais se deu com o intuito de se obter uma forma automática de extração de atributos e classificação relacionados à corrosão de tubulações usadas na indústria termoelétrica. Os dados utilizados são provenientes de fotografias internas de tubos, que com base na textura são classificadas previamente como: não corroída, corrosão de grau médio e corrosão severa. Com a utilização da técnica de *data augmentation*, o conjunto de dados chegou à 2500 imagens e empregando uma arquitetura baseada na rede AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) obteve a acurácia de 99,2% na classificação de corrosão interna.

Ainda baseado na análise de texturas, porém dedicado à observação de corrosão

externa de tubulações, em (HOANG; DUC, 2019) utiliza-se um conjunto de dados composto por 2000 imagens, extração de atributos por análise estatística dos canais da imagem como média, desvio padrão, curtose e entropia, dentre outras técnicas descritoras de textura, resultando em 78 características extraídas. Aplicando o classificador SVM (*Support Vector Machine*) com a ferramenta *Differential Flower Pollination* de otimização para a busca de hiperparâmetros, foi realizada a distinção entre amostra corroída e não corroída. A acurácia média obtida foi 92,81% no conjunto de teste (composto por 30% dos dados totais).

2.3 **Estimativa da Profundidade de Defeitos**

Como já mencionado, a união entre aprendizado de máquinas e modelagem de processos corrosivos é um campo em eclosão, principalmente em problemas de regressão. Portanto, a literatura ainda apresenta poucos trabalhos nesta área, porém a publicação de (SEGHIER et al., 2020) representa um caso significativo de aplicação do aprendizado de máquinas na predição de profundidade de defeitos ocasionados pela corrosão puntiforme em tubulações subterrâneas. Com a utilização das técnicas de otimização por Algoritmos Genéticos e Enxame de Partículas visando a obtenção de hiperparâmetros ótimos para o algoritmo SVR (*Support Vector Regressor*). Este regressor é alimentado com um conjunto de dados composto por 259 amostras, sendo 80% para treinamento e 20% para teste, contendo informações físico-químicas a respeito do solo onde a tubulação está instalada, bem como seu tempo de exposição. De modo que este escrito atingiu o incremento de desempenho de 29,72% quando comparado à trabalhos predecessores que utilizaram expressões analíticas para cálculo da profundidade de danos.

3 APRENDIZADO DE MÁQUINAS E REDES NEURAIS CONVOLUCIONAIS

Neste trabalho foram empregados sete algoritmos tradicionais no aprendizado de máquinas, considerando suas abordagens para regressão (estimativa da profundidade dos alvéolos) e classificação da gravidade dos danos encontrados (posteriormente comparada com os resultados obtidos com a utilização de redes neurais convolucionais). Os modelos foram baseados nas Regressões Linear e Logística, *k-Nearest Neighbor* (k-NN), Vetor de Suporte (do inglês, *Support Vector Machine* - SVM), Árvores de Decisão, *Random Forest* e *Gradient Boosting*. Suas principais características tanto como classificadores ou regressores são descritas ao longo deste Capítulo. Porém, independentemente do algoritmo analisado, todos estão inseridos no mesmo processo de desenvolvimento mostrado na Figura 11.

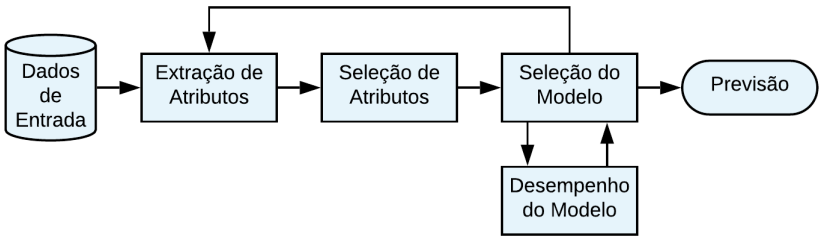


Figura 11: Diagrama de desenvolvimento de um modelo tradicional de aprendizado de máquina

A fase de extração de atributos recebe os dados brutos que alimentam o sistema, nela, por exemplo, são realizadas as operações de filtragem e pré-processamento se necessárias. Na sequência, existe a possibilidade de nem todos os atributos disponibilizados pelo extrator serem relevantes ao problema em questão. Então, visando a redução de uma complexidade desnecessária do modelo (JAMES et al., 2014) surge a etapa de seleção de atributos onde a escolha poder ser feita com base em propriedades estatísticas, métodos de busca ou por especialistas com conhecimento da situação em análise. Por sua vez, no estágio de seleção do modelo ocorre o treinamento propriamente dito, onde são ajustados hiperparâmetros e obtidas as previsões. Entretanto, note que é um processo iterativo com base no desempenho do modelo. A métrica de desempenho pode levar em conta fatores como generalização e qualidade das previsões, velocidade de treinamento, escalabilidade ou interpretabilidade da solução obtida. Caso os critérios estabelecidos não sejam atingidos pode ser necessário retornar ao passo inicial de extração de atributos.

3.1 Regressões Linear e Logística

Um dos primeiros métodos de aprendizado supervisionado, a regressão linear tem papel importante tanto prático como conceitual. Para o caso com apenas uma variável, a predição da regressão linear pode ser representada como mostrada na Equação 15; onde $\hat{y}_i$ é a predição, β_0 e β_1 são os parâmetros a serem determinados e x é a variável de entrada.

$$\hat{y} = \beta_0 + \beta_1 x \tag{15}$$

Considerando um conjunto de dados X com N exemplos, o valor predito para o i-ésimo valor de X é definido por $\hat{y}_i = \beta_0 + \beta_1 x_i$. A partir disso é possível definir uma medida de erro que relaciona os valores reais e os preditos pelo modelo, dada por $e_i = y_i - \hat{y}_i$. E na sequência é definida a soma dos resíduos ao quadrado (SRQ) conforme Equação 16.

$$SRQ = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} (y_i - \beta_0 - \beta_1 x_i)^2 \tag{16}$$

Assim, o objetivo é minimizar o erro cometido pelo modelo. Com a aplicação do método dos mínimos quadrados, a função de custo SRQ pode ser minimizada e obtermos os valores ótimos para os parâmetros β_0 e β_1, como mostrado nas Equações 17a e 17b onde $\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ e $\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$ (JAMES et al., 2014). Na Figura 12 é mostrado um exemplo de aplicação da regressão linear simples.

$$\beta_1 = \frac{\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{N}(x_i - \overline{x})^2} \tag{17a}$$

$$\beta_0 = \overline{y} - \beta_1 \overline{x} \tag{17b}$$

Entretanto, para situações onde há múltiplas variáveis a obtenção dos parâmetros é computacionalmente custosa por envolver álgebra matricial da ordem N^3, tornando o modelo não escalável. Na Equação 18 é apresentada a solução analítica para múltipla regressão, onde $\boldsymbol{\beta}$ é a matriz de parâmetros, $\mathbf{X}$ os exemplos observados e $\mathbf{Y}$ os valores alvos. A estimativa obtida é dada por $\hat{\mathbf{Y}} = \boldsymbol{\beta}^T \mathbf{X}$ (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

$$\boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \tag{18}$$

Para reduzir o custo computacional, são obtidos modelos sub ótimos de regressão múltipla por meio da descida do gradiente.

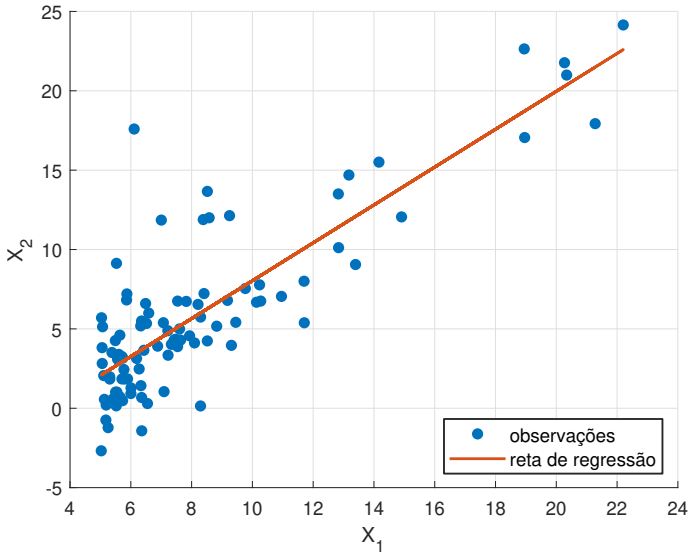


Figura 12: Exemplo de aplicação para regressão linear simples

No caso da regressão logística objetiva-se obter a probabilidade de um vetor de atributos de entrada pertencer a uma classe. Ou seja, trata-se de classificação. Este modelo é baseado na função sigmoide ou logística mostrada na Equação 19 e cuja forma é apresentada na Figura 13. A função sigmoide retorna valores entre 0 e 1, portanto pode ser utilizada para expressar a probabilidade de um exemplo pertencer a uma classe ou à outra, supondo classificação binária.

$$\theta(s) = \frac{e^s}{1 + e^s} \tag{19}$$

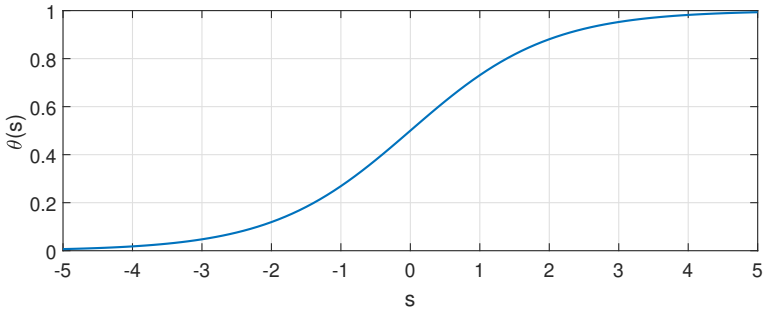


Figura 13: Função sigmoide

Após manipulação da Equação 19 obtemos as expressões mostradas nas Equações 20a e 20b. Tal como na regressão linear, a regressão logística é modelada de forma linear

fazendo $s = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$, generalizando para p variáveis.

$$\frac{\theta(s)}{1-\theta(s)} = e^s \tag{20a}$$

$$ln\left(\frac{\theta(s)}{1-\theta(s)}\right) = s \tag{20b}$$

Técnicas como a estimativa de máxima verossimilhança são usadas para estimar os parâmetros β do modelo. Este método determina os valores dos parâmetros do modelo que maximizam a função de custo definida pela Equação 20b (JAMES et al., 2014). Para isso a descida do gradiente também pode ser empregada, sendo preciso multiplicar a função de custo por (-1) por se tratar de um método de minimização.

Ambos modelos apresentados nesta seção são paramétricos, ou seja assumem alguma distribuição para o conjunto de dados, nos casos a hipótese feita é a de distribuição linear. Esta característica retrata modelos de treinamento simples, entretanto para conjuntos de dados que não podem ser linearmente modelados o desempenho destas técnicas pode ser insatisfatório.

3.2 *k-Nearest Neighbors*

O algoritmo *k-Nearest Neighbors* (k-NN) ou k-Vizinhos mais Próximos é do tipo não paramétrico, ou seja não faz hipóteses sobre a distribuição do conjunto de dados de treinamento; isso permite que o modelo encontre padrões naturais em vez de tentar ajustar os dados em uma distribuição pré definida e potencialmente enviesada. De funcionamento simples, porém com capacidade para fornecer resultados próximos do ótimo (LANTZ, 2015; JAMES et al., 2014), sua ação é baseada na distância entre um ponto de teste (não rotulado) e os pontos de treinamento (rotulados). Ou seja, dados um ponto de teste e um número positivo k (ímpar, a fim de evitar empates), o algoritmo identifica os k pontos vizinhos nos dados de treinamento que estão mais próximos da amostra de teste.

Como classificador, o k-NN atribui ao ponto de teste a classe à qual a maioria das instâncias mais próximas pertencem; já como regressor é atribuída a média dos valores dos k pontos mais próximos, devido a característica não paramétrica, o algoritmo fornece uma abordagem alternativa e mais flexível para realizar a regressão. Na Figura 14 é apresentado um exemplo de classificação binária com k-NN. Note a importância da escolha do valor para k, com $k = 1$ o ponto de teste seria rotulado como pertencente à Classe B,

enquanto usando $k = 3$ o mesmo ponto pertenceria à Classe A. Outra questão relevante, é a disparidade entre os valores dos atributos, um procedimento comum é redimensionar todos os dados para equilibrá-los (JAMES et al., 2014; DUDA; HART; STORK, 2000).

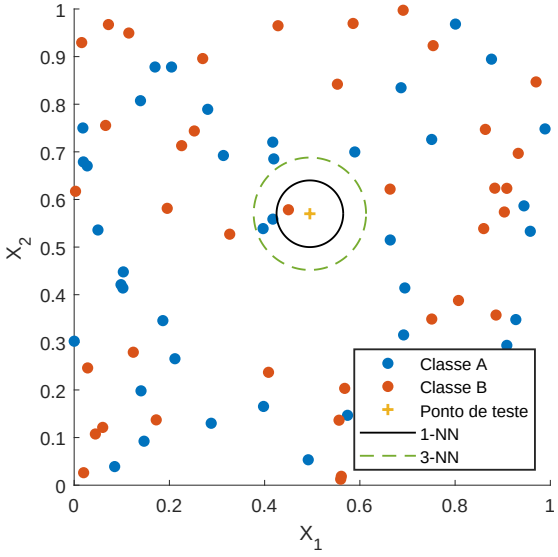


Figura 14: Exemplo de classificação com k-NN

O k-NN é considerado um algoritmo *preguiçoso*, uma vez que não há aprendizado do ponto de vista do aprendizado de máquinas. Em vez disso, ele armazena os dados de treinamento literalmente. Isso permite que a fase de treinamento, ocorra muito rapidamente. A consequência é que o processo de fazer previsões tende a ser relativamente lento em comparação ao treinamento. Devido à forte dependência das instâncias de treinamento, o aprendizado preguiçoso também é conhecido como aprendizado baseado em instâncias (LANTZ, 2015).

3.3 *Support Vector Machine*

Em um espaço de dimensão p, um hiperplano é definido como um subespaço de dimensão $p - 1$. Por exemplo, em um espaço bidimensional, o hiperplano se reduz a uma reta. Na Equação 21 é mostrada a forma geral para um espaço p-dimensional, onde o vetor de atributos é dado por $\mathbf{X} = (X_1, X_2, \ldots, X_p)^T$ (JAMES et al., 2014).

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \tag{21}$$

No caso bidimensional é possível visualizar que o hiperplano $1 + 2X_1 + 3X_2 = 0$ atua como fronteira de decisão, conforme mostrado na Figura 15. Os pontos em azul atendem

à condição $1 + 2X_1 + 3X_2 > 0$ e a região em vermelho representam $1 + 2X_1 + 3X_2 < 0$.

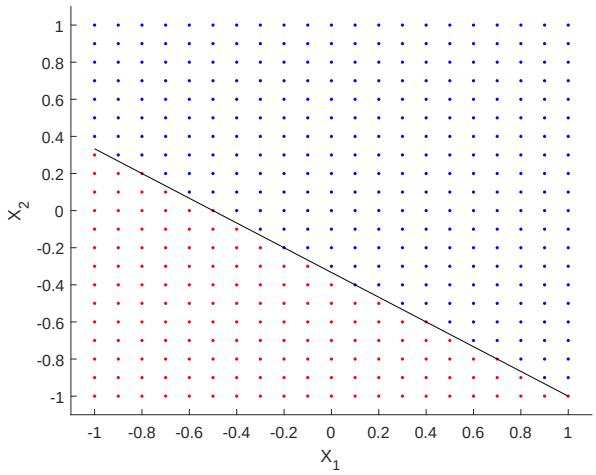


Figura 15: Hiperplano em duas dimensões
Fonte: adaptado de JAMES et al., 2014.

O conceito de margem pode ser definido a partir da distância (perpendicular) de cada observação de treinamento a um dado hiperplano de separação; a menor dessas distâncias é a distância mínima das observações ao hiperplano e é conhecida como margem. A margem máxima é o hiperplano de separação para o qual a margem é maior - ou seja, é o hiperplano que tem a distância mínima mais distante para as observações de treinamento. O algoritmo *Support Vector Machine* - SVM ou Máquina de Vetor de Suporte classifica uma observação de teste dependendo de qual lado do hiperplano ela está. O objetivo do treinamento do SVM é encontrar o hiperplano de separação com a maior margem; espera-se que quanto maior a margem, melhor generalização do classificador (JAMES et al., 2014; DUDA; HART; STORK, 2000).

Considerando um exemplo de classificação binária linearmente separável, na Figura 16 está ilustrada a classificação com o SVM. As classes de observações são mostradas em azul e em vermelho. Os dois pontos vermelhos e o ponto azul que se encontram nas linhas tracejadas são os vetores de suporte, e, as margens são indicadas por setas. As regiões $\mathcal{R}_1$ e $\mathcal{R}_2$ indicam a regra de decisão feita pelo classificador com base neste hiperplano de separação.

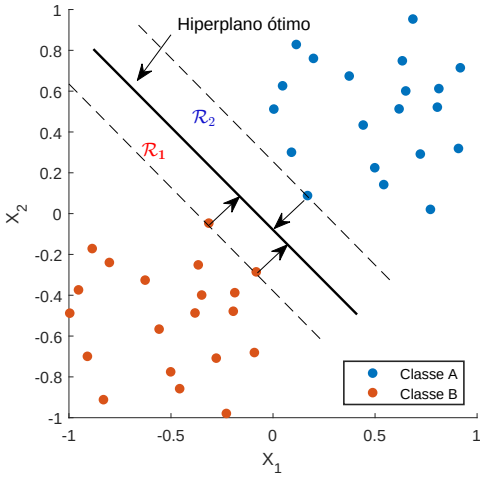


Figura 16: Exemplo de classificação com SVM
Fonte: adaptado de DUDA; HART; STORK, 2000.

Segundo (JAMES et al., 2014), o funcionamento do SVM pode ser compreendido como a resolução do problema com restrições mostrado nas Equações 22a - 22d, considerando um conjunto com n pontos e de dimensão p:

$$\text{maximixar}_{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_i,\ldots,\epsilon_n,M} \quad M \tag{22a}$$

$$\text{sujeito a} \sum_{j=1}^{p} \beta_j^2 = 1, \tag{22b}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \tag{22c}$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C, \tag{22d}$$

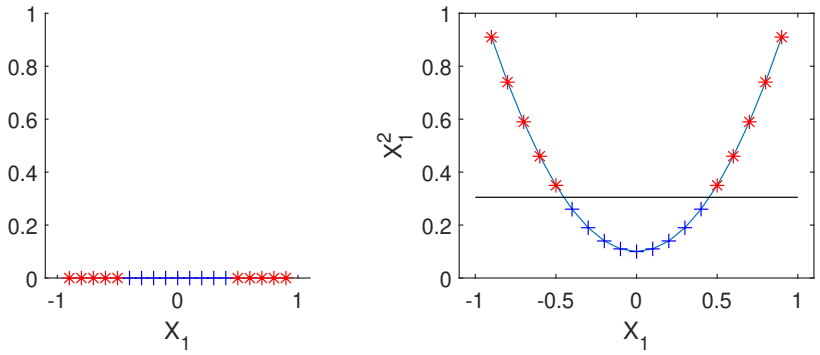
onde M é a margem; $\epsilon_i \cdots \epsilon_n$ são variáveis de folga que permitem que alguns pontos de treinamento estejam do lado errado da margem ou do hiperplano, tornando o classificador mais flexível e C é um parâmetro de ajuste não negativo.

Uma vez resolvido o problema de otimização com restrições, as variáveis soltas retratam onde o respectivo ponto está localizado em relação ao hiperplano ou à margem. Por exemplo, $\epsilon_i = 0$ indica que a i-ésima amostra está no lado correto da margem; quando $\epsilon_i > 0$ o respectivo ponto está no lado errado da margem, diz-se que ele violou a margem. Enquanto, $\epsilon_i > 1$ indica que a observação está no lado errado do hiperplano.

O hiperparâmetro C limita o somatório das variáveis soltas, determinando a quantidade e a severidade toleradas das violações tanto de margem como de hiperplano. Quando C é pequeno, buscamos margens estreitas que raramente são violadas; isso equivale a um classificador que é altamente ajustado aos dados, tendendo ao sobre ajuste (*overfitting*).

Por outro lado, quando C é maior, a margem é mais ampla e permitimos mais violações. Em outras palavras, C permite o equilíbrio *bias-variância* e na prática seu valor ideal é obtido via validação cruzada (JAMES et al., 2014).

Entretanto, nem sempre é possível obter uma separação linear diretamente, na Figura 17 (a) é mostrada uma situação na qual não é possível obter um hiperplano de separação, para contornar este caso usa-se a estratégia de elevar a dimensionalidade do problema; na Figura 17 (b) é ilustrada a transformação $X_1 \rightarrow X_1^2$, neste no espaço os pontos são linearmente separáveis. Entretanto, encontrar a transformação adequada não é uma tarefa trivial. O SVM permite utilizar funções (*kernels*) que realizam esta transformação como parte do treinamento, os mais comumente empregados são *Radial Basis Function* - RBF, Polinomial e o Gaussiano.



(a) espaço unidimensional, pontos não linearmente separáveis

(b) espaço bidimensional, pontos linearmente separáveis

Figura 17: Exemplo transformação não linear em um conjunto de dados

Uma extensão do algoritmo é o *Support Vector Regression* - SVR, dedicado para problemas de regressão. Neste caso, além das características citadas para o SVM, o SVR procura que minimizar alguma função de custo que relaciona os valores predito e real para o conjunto de treinamento. À medida que o parâmetro de regularização C aumenta, o peso para a minimização da função de custo aumenta. Um tubo flexível de raio mínimo é formado simetricamente em torno da função estimada, de modo que os valores absolutos dos erros menores que um certo limite sejam ignorados acima e abaixo da estimativa. Desta forma, os pontos fora do tubo são penalizados, mas aqueles dentro do tubo, acima ou abaixo da função, não recebem penalidade. Uma das principais vantagens do SVR é que sua complexidade computacional não depende da dimensionalidade do espaço de entrada. Além disso, possui excelente capacidade de generalização, com alta precisão de

predição. (AWAD; KHANNA, 2015; JAMES et al., 2014). Na Figura 18 é mostrado um exemplo de regressão não linear utilizando SVR.

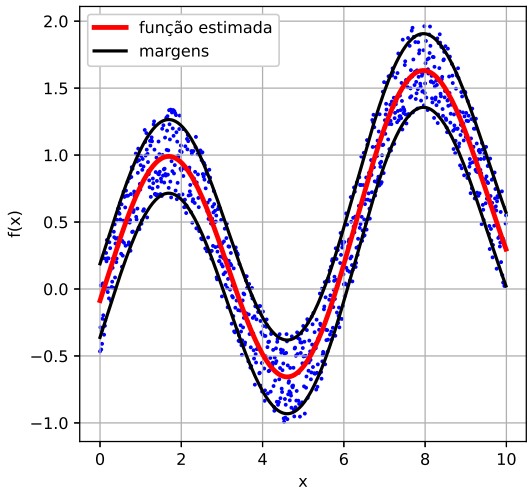
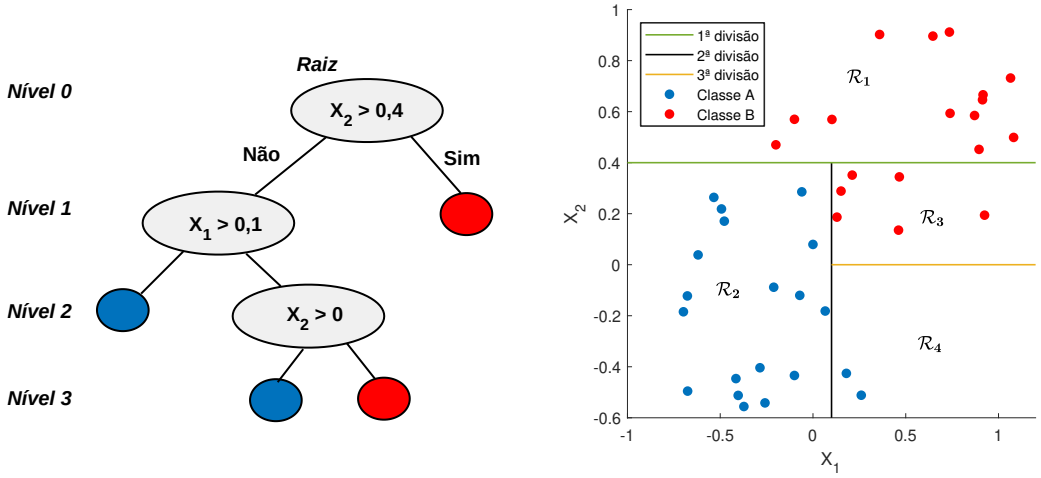


Figura 18: Exemplo de regressão não linear com SVR

Dentre as características das SVM/SVR estão a baixa influência de dados ruidosos e a resistência ao *overfitting*, por outro lado apresenta treinamento lento particularmente em conjuntos com grande número de exemplos e/ou de atributos. E a obtenção de modelos do tipo *caixa preta*, que não permitem ou dificultam a interpretabilidade.

3.4 Árvores de Decisão

As árvores de decisão ou CART (do inglês, *Classification and Regression Trees*) são modelos não paramétricos e com alta interpretabilidade de como determinada solução é encontrada. Baseadas na naturalidade intuitiva de se classificar um padrão por meio de uma sequência de perguntas, as árvores de decisão podem ser representadas como uma sequência de perguntas disposta em um grafo acíclico e orientado, conforme Figura 19(a). Por convenção, o nó *raiz* é o topo da árvore e é conectado direcionalmente por *ramos* a outros nós. Estes são conectados de modo similar até atingir os nós terminais, as *folhas*. Os ramos do nó raiz correspondem aos diferentes valores possíveis. Com base na resposta, seguimos o ramo apropriado para um nó subsequente ou descendente. Assim, os nós internos representam um teste de atributo, os ramos indicam os resultados desses testes e a folhas caracterizam os rótulos da classe ou distribuição (no caso de regressão) (DUDA; HART; STORK, 2000).



(a) estrutura da árvore resultante (b) particionamento realizado

Figura 19: Árvore de decisão e o efeito do particionamento do conjunto de treinamento

Os exemplos de treinamento são particionados recursivamente com base nos atributos selecionados, como mostrado na Figura 19(b) onde cada região representa uma folha da árvore na Figura 19(a); a atribuição da classe é baseada naquela de maior ocorrência em dada região. Esta seleção, comumente, deriva de propriedades estatísticas como o ganho de informação, por exemplo. Para tal, duas abordagens podem ser empregadas, o índice de Gini e a entropia definidos nas Equações 23 e 24 respectivamente. Sendo K o número de classes e P_k a probabilidade de se observar a classe k nos nós em análise.

$$G = \sum_{k=1}^{K} P_k(1 - P_k) \tag{23}$$

$$E = -\sum_{k=1}^{K} P_k \cdot log_2 P_k \tag{24}$$

Estas métricas estão relacionadas com o grau de aleatoriedade da variável alvo, ou seja é uma medida de *impureza*. Quanto maior o número de elementos da mesma classe no k-ésimo nó, menores são os valores do índice de Gini e de entropia. O ganho de informação GI de uma divisão pode ser calculado levando em conta tanto o índice de Gini como a entropia, conforme apresentado nas Equações 25a e 25b.

$$GI_G = G|_{nó\ pai} - G|_{nós\ \ filhos} \tag{25a}$$

$$GI_E = E|_{nó\ pai} - E|_{nós\ \ filhos} \tag{25b}$$

No caso de regressão, para cada ponto de teste que cai na região R_j, fazemos a

mesma previsão, que é simplesmente a média dos valores de treinamento em R_j. Neste tipo de problema, o objetivo é por meio do particionamento minimizar a soma residual dos quadrados definida na Equação 26. Onde $\overline{y}_{Rj}$ é média das observações de treinamento dentro região R_j e y_i o valor alvo (JAMES et al., 2014).

$$SRQ = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \overline{y}_{Rj})^2 \tag{26}$$

O crescimento da árvore até que cada nó folha corresponda à impureza mais baixa, pode levar ao caso extremo em que cada folha corresponde a um único ponto de treinamento, incorrendo em *overfitting*; portanto, não se espera boas generalizações em problemas ruidosos. Por outro lado, se a divisão for interrompida muito cedo, o erro nos dados de treinamento não é suficientemente baixo e, portanto, o desempenho pode ser prejudicado. Por isso, a profundidade p da árvore deve ser controlada, para tal uma abordagem tradicional é usar a técnica de validação cruzada. Ou seja, a árvore é treinada usando um subconjunto dos dados, com o restante mantido como um conjunto de validação. Continuamos dividindo os nós em camadas sucessivas até que o erro nos dados de validação seja minimizado (DUDA; HART; STORK, 2000).

As árvores de decisão apresentam importantes vantagens como a capacidade de lidar com atributos numéricos, nominais e dados ausentes. Também permite a exclusão de atributos irrelevantes, além da interpretabilidade já mencionada. Por outro lado, o particionamento retangular pode limitar o desempenho com atributos correlacionados.

3.5 ***Random Forest***

Modelo introduzido por (BREIMAN, 2001), a floresta aleatória (do inglês, *Random Forest*) é baseada em árvores de decisão, porém fornecem uma melhoria em relação à estas por meio da técnica conhecida como *bagging* (*bootstrap agreggation*) que reduz a variância final. *Bagging* é uma técnica em que os modelos básicos são desenvolvidos alterando o conjunto de treinamento para cada modelo básico (JAMES et al., 2014).

Em um determinado conjunto de treinamento T contendo n amostras, m conjuntos de treinamento são desenvolvidos, cada um também com n exemplos, por amostragem com substituição. Cada conjunto de treinamento $T_1, T_2, T_3, \ldots, T_m$ terá a mesma quantidade de amostras que o conjunto de treinamento original T. Como eles são amostrados

com substituição, podem conter observações duplicadas. Esse procedimento caracteriza o *bootstrap*. Cada conjunto de treinamento amostrado é então usado para a preparação de um modelo básico. Isto leva à produção de um conjunto de modelos base e a previsão de cada modelo é agregada para um modelo do tipo comitê, onde cada modelo básico é uma árvore de decisão. Essa combinação de *bootstrap* e agregação é chamada de *bagging* (KOTU; DESHPANDE, 2015).

A estratégia de construção de uma floresta aleatória usa um conceito semelhante ao usado no *bagging*. Ao decidir sobre a divisão de cada nó em uma árvore de decisão, o *random Forest* considera apenas um subconjunto aleatório de todos os atributos no conjunto de treinamento. Para reduzir o erro de generalização, o algoritmo utiliza dois níveis de aleatoriedade, escolha de exemplos de treinamento e seleção de atributos, no funcionamento interno de cada modelo base (KOTU; DESHPANDE, 2015).

Depois que todas as n árvores da floresta são construídas, para cada nova amostra, todas as árvores preveem uma classe e votam na classe com pesos iguais. A classe mais prevista pelas árvores base é a predição da floresta, isto no caso de classificação. Para regressão, a previsão final é a média dos valores previstos pelas árvores base (BREIMAN, 2001).

De modo geral, os principais ganhos com o emprego do *random Forest* estão na capacidade de lidar com dados ruidosos ou ausentes, categóricos e numéricos; além de selecionar apenas os atributos mais relevantes ao problema. Entretanto, por se tratar de um comitê há redução na interpretabilidade quando comparadas à árvore de decisão.

3.6 *Gradient Boosting*

Assim como no *random Forest*, o algoritmo *gradient boosting* apresentado por (FRIEDMAN, 2000) também é baseado em comitê de árvores de decisão. Outra maneira de incrementar o desempenho das árvores de decisão é com o emprego da técnica conhecida como *boosting* (reforço); este termo se refere a uma família de algoritmos capaz de converter modelos "fracos"em um modelo "forte"(ZHOU, 2012).

No *bagging* cada árvore é construída em um conjunto de dados obtido via *bootstrap*, portanto independente das outras árvores. Enquanto no *boosting* as árvores são obtidas sequencialmente: cada árvore é treinada usando informações de árvores previamente desenvolvidas utilizando uma estratégia que visa minimizar o erro anterior (JAMES et al.,

2014). Em outras palavras, existem duas distinções principais. Primeiro, no *boosting* os conjuntos de dados sub amostrados, sem substituição, são construídos especificamente para gerar modelos complementares. Em segundo lugar, em vez de dar a cada modelo um voto com o mesmo peso, o *boosting* dá ao voto de cada modelo fraco um peso com base em seu desempenho anterior. Modelos com melhor desempenho têm maior influência sobre o predição final do conjunto (LANTZ, 2015).

Em seguida, a estratégia do *gradient boosting* se concentra em minimizar uma função de custo diferenciável que leva em conta a previsão e seu valor real. Para tal é utilizada a descida do gradiente. A cada iteração uma árvore de decisão é adicionada para ajustar os erros cometidos pelo comitê. Os principais hiperparâmetros do modelo são o número de árvores n, o número de divisões em cada árvore e taxa de aprendizado η, que controla o passo de atualização na na função objetivo. Segundo (JAMES et al., 2014), η muito pequeno pode exigir grande quantidade de árvores para obter um bom desempenho.

O *gradient boosting* também conta com a seleção implícita de atributos, porém apresenta dificuldades de escalabilidade por se tratar de um modelo sequencial e seu treinamento é mais lento.

3.7 **Redes Neurais Convolucionais**

Desde a década de 1980 já havia interesse na busca de algoritmos que lidassem diretamente com imagens, em 1989 o primeiro caso exitoso de rede neural convolucional foi a ConvNet por (LeCun et al., 1989), usando o treinamento baseado no algoritmo *backpropagation* cujo objetivo foi o reconhecimento de dígitos manuscritos. Anos mais tarde foi proposta, como aprimoramento da ConvNet, a rede convolucional LeNet-5 (Lecun et al., 1998) obtendo bons resultados no reconhecimento óptico de caracteres e impressões digitais, fazendo com que fosse utilizada em sistemas bancários (KHAN et al., 2020).

Entretanto, as redes convolucionais tiveram um período de estagnação entre 2000 e 2011, perdendo espaço para o SVM, principalmente devido ao reduzido tamanho dos conjuntos de treinamento e baixo poder computacional disponíveis até então. Pesquisadores da Universidade de Stanford, nos Estados Unidos, consolidaram dois grandes conjuntos de dados conhecidos como ImageNet (com 15 milhões de imagens) e PASCAL 2010 VOC, o que aliado ao desenvolvimento computacional, novas estratégias de otimização e de arqui-

teturas possibilitou o retorno das redes neurais convolucionais. Como exemplo dessa nova era, a AlexNet em 2012 por (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) obteve taxa de erro na classificação de 15,3% contra os 26,2% da segunda colocada na competição 2012-ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*). Isto abriu portas para o que ficaria conhecido como redes neurais convolucionais profundas, que desde então, têm sido cada vez mais empregadas (KHAN et al., 2020). A partir disso houve rápido desenvolvimento na área de redes convolucionais, o que resultou em arquiteturas como R-CNN (GIRSHICK et al., 2013), GoogleNet (SZEGEDY et al., 2014), ResNet (HE et al., 2015) dentre outras. Recentemente, com as redes U-Net (RONNEBERGER; FISCHER; BROX, 2015) e FCN (LONG; SHELHAMER; DARRELL, 2014) surgiu o campo conhecido como segmentação semântica, onde se atribui um rótulo categórico a cada *pixel* de uma imagem, o que desempenha um papel importante na compreensão da imagem.

Apesar de não haver definição consensual de limite para considerar uma rede neural profunda (*deep*) ou rasa (*shallow*) as redes neurais convolucionais acabam sendo consideradas como profundas, por possuírem mais de duas camadas (NIELSEN, 2015). Esse tipo de algoritmo é composto por camadas sequenciais, como pode ser observado no exemplo de arquitetura típica mostrada na Figura 20, ilustrando as camadas de convolução, *pooling* e classificação (camada totalmente conectada).

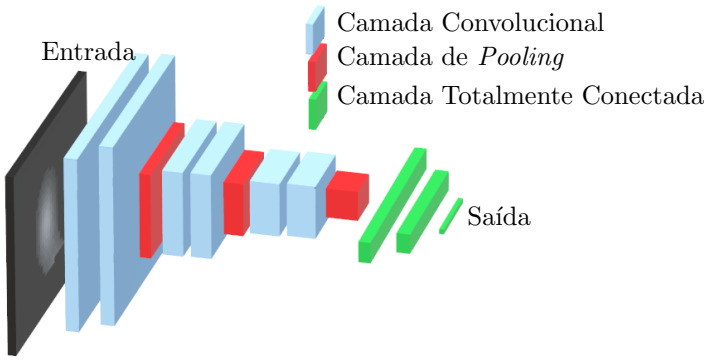


Figura 20: Arquitetura típica de uma rede neural convolucional
Fonte: adaptado de MAZUROWSKI et al., 2018.

Considerando imagens como matrizes de duas dimensões, uma imagem colorida pode ser entendida como se houvesse três matrizes bidimensionais empilhadas, uma para cada cor básica (vermelho, verde e azul); ao passo que para uma imagem em tons de cinza seria necessário apenas uma matriz para representá-la. Com isso, uma imagem colorida também pode ser encarada como um volume tridimensional (largura, altura e canais de

cor), assim a típica arquitetura convolucional pode ser interpretada como uma sequência de camadas que transformam um volume de entrada em pontuações de classe.

Uma rede convolucional é capaz de capturar com sucesso as dependências espaciais em uma imagem através da aplicação de filtros relevantes, ou seja é uma forma automática de extração de atributos; esta característica de redes profundas possibilita a descoberta de atributos que antes poderiam não ser alcançados por um especialista no problema em questão. A arquitetura convolucional executa um melhor ajuste ao conjunto de dados da imagem devido à redução no número de parâmetros envolvidos e na reutilização de pesos. A redução do número de parâmetros é uma comparação entre a aplicação de rede neural artificial tradicional totalmente conectada como a *Perceptron* para uma mesma tarefa. Por exemplo, considerando como entrada uma imagem 64x64x3, uma rede *Multilayer Perceptron* com 10 neurônios e uma camada escondida resultaria em 122.890 pesos para serem aprendidos, incluindo *bias*. Esta grande quantidade de parâmetros favorece a ocorrência de *overfitting*. Conforme descrito nas seções a seguir, o número de total pesos a serem aprendidos pela rede convolucional depende do tamanho dos filtros aplicados, porém pelas características das operações de convolução e *pooling* será menor que caso fosse utilizada uma rede neural tradicional (STANFORD, 2019; SAHA, 2018).

Algumas das camadas que compõem uma rede convolucional contêm hiperparâmetros e parâmetros tais quais as camadas convolucionais e totalmente conectada; pois elas realizam transformações em função não apenas das ativações no volume de entrada, mas também dos parâmetros (pesos dos neurônios). Por outro lado, a camada de *pooling* implementa uma função fixa, ou seja, apresenta apenas parâmetros (STANFORD, 2019). A seguir cada camada é analisada.

3.7.1 Camada Convolucional

A camada de convolução é responsável, dentre outras funções, por realizar a operação de convolução com o objetivo de extrair características (atributos) da imagem de entrada. A operação de convolução utiliza duas matrizes, a primeira é chamada matriz de entrada que consiste na própria imagem e a segunda, é o que se conhece como filtro (ou do inglês, *kernel*). Quando se utiliza o termo "convolução MxNxP"fica implícito que o filtro empregado tem dimensões MxN e P é o número de filtros. Na Figura 21 é mostrada a ilustração do processo de convolução.

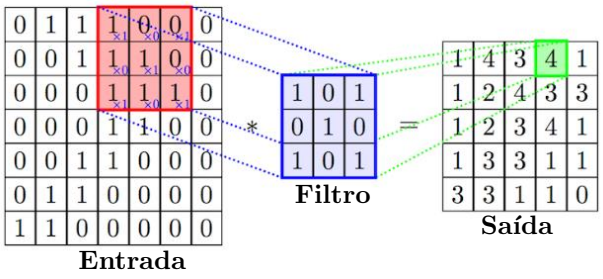


Figura 21: Operação de convolução
Fonte: adaptado de CHATTERJEE, 2017.

A operação convolucional é realizada pelo deslizamento do filtro sobre a matriz de entrada, cujo deslocamento é dado pelo hiperparâmetro "passo"(do inglês, *stride*). O filtro convolucional funciona dividindo a imagem em pequenas fatias, comumente conhecidas como campos receptivos. A divisão de uma imagem em pequenos blocos ajuda na extração de atributos. À medida que o filtro é deslizado sobre o volume de entrada, produz-se um mapa de ativação bidimensional (ou mapa de atributos) que fornece as respostas desse filtro em todas as posições espaciais. Intuitivamente, a rede aprenderá filtros que são ativados quando encontram algum tipo de atributo visual, como bordas por exemplo. Podem ser empregados diversos filtros em uma mesma camada convolucional e cada um deles produzirá seu mapa de ativação bidimensional próprio. Empilhando esses mapas de ativação ao longo da dimensão de profundidade (dada pelo número de filtros) é produzido o volume de saída. Então, obtém-se uma medida de semelhança entre o filtro e determinada parte da imagem, já que a convolução apresenta valores maiores nestas regiões (STANFORD, 2019; KETKAR, 2017). Veja, que na verdade, não se trata de uma operação matricial, mas sim do produto interno entre o valor do pixel representado na matriz de entrada e o valor da correspondente posição do filtro. Retomando a Figura 21 tem-se:

linha 1: $1{\cdot}1 + 0{\cdot}0 + 0{\cdot}1 = 1$;

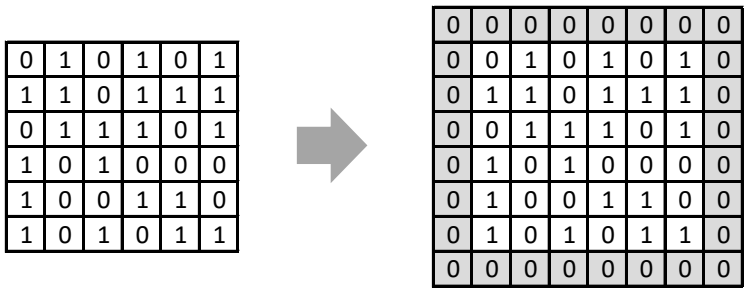
linha 2: $1{\cdot}0 + 1{\cdot}1 + 0{\cdot}0 = 1$;

linha 3: $1{\cdot}1 + 1{\cdot}0 + 1{\cdot}1 = 2$.

Somando estes resultados parciais, obtém-se o valor "4"em sua posição devida no mapa de atributos. Note também, que o filtro causa redução na dimensionalidade da matriz de saída quando comparada à entrada.

Por outra característica da convolução, as bordas da entrada serão "vistas"apenas uma vez, enquanto os valores intermediários terão o filtro deslizado sobre si diversas vezes,

isto faz com que se perca informações acerca da fronteira da matriz de entrada. Para resolver este problema, utiliza-se o artifício do preenchimento (do inglês, *padding*), cujo tamanho é um hiperparâmetro, consistindo na colocação de zeros ao redor das bordas matriz de entrada, conforme mostrado na Figura 22 com preenchimento de tamanho unitário. Assim, o preenchimento também pode ser utilizado para controlar o tamanho do mapa de atributos, compensando a redução provocada pela operação de convolução.



(a) matriz original (b) após a aplicação do preenchimento

Figura 22: Exemplo de aplicação do *padding*

Segundo (BUDUMA, 2017), é aconselhável que os filtros sejam pequenos, algo como 3 x 3 ou 5 x 5, podendo ser utilizado filtros 7 x 7 apenas na primeira camada convolucional, pois ter mais filtros pequenos aumenta o poder de representação e diminui o número de parâmetros. Também é sugerida a utilização de passo unitário, para que se capture todas as informações úteis do mapa de atributos. E, ainda, a utilização do preenchimento para manter as dimensões de saída iguais às da entrada.

Outra atribuição da camada convolucional é a aplicação da função de ativação, que se dá como fator de decisão no aprendizado de padrões complexos graças a aplicação de não-linearidades sobre o mapa de atributos. Diferentes funções de ativação já foram usadas em redes convolucionais, como por exemplo sigmoide, tangente hiperbólica e *Rectified Linear Unit* (ReLU) bem com suas variantes *leaky* ReLU, ELU e PReLU. Entretanto, as funções de ativação baseadas na ReLU e ela própria tem sido amplamente utilizadas por terem o efeito de desvanecimento do gradiente mitigado, favorecendo a aplicação do algoritmo de treinamento *backpropagation* (KHAN et al., 2020). A forma da função ReLU é mostrada na Figura 23.

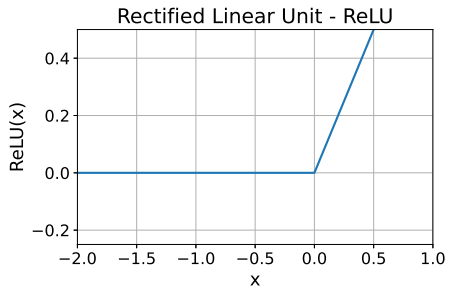


Figura 23: Forma da função ReLU

3.7.2 Camada de *Pooling*

Uma vez extraídos os mapas de atributos pelos passos anteriores de convolução e ativação, a localização exata das características obtidas perde importância desde que sua posição aproximada em relação ao restante da imagem seja preservada. Isto significa que pode ser realizada uma subamostragem que mantenha as relações espaciais relativas das características já obtidas. Com isso, a camada de *pooling* tem como objetivo reduzir o tamanho do mapa de características e principalmente proporcionar certo grau de invariância espacial. Dentre os efeitos da utilização da camada de *pooling* estão a redução da quantidade de parâmetros, do custo computacional e da ocorrência de *overfitting* (KHAN et al., 2020; STANFORD, 2019).

A ideia por trás do *pooling* é dividir os mapas de atributos em blocos de tamanhos iguais e em seguida criar um mapa de atributos condensado. Especificamente, cria-se uma célula para cada bloco e calcula-se o máximo (max *pooling*) ou a média (med *pooling*) e transporta-se esse valor resultante para a célula correspondente do mapa da atributos condensado. O *pooling* considerando a média foi historicamente empregado, entretanto vem perdendo espaço para o *pooling* máximo que tem apresentado melhores resultados. Um exemplo destas operações é mostrado na Figura 24. A camada de *pooling* possui dois hiperparâmetros, o tamanho sendo mais comumente usado 2x2 e o passo de deslizamento sobre o mapa de atributos cujo valor mais utilizado é 2 (STANFORD, 2019; SAHA, 2018; BUDUMA, 2017).

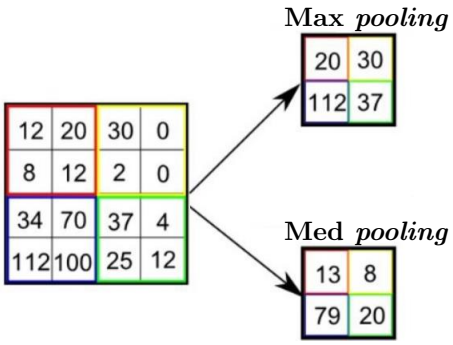


Figura 24: Exemplos de *pooling*
Fonte: adaptado de SAHA, 2018.

3.7.3 Camada Totalmente Conectada

A camada totalmente conectada, consiste na aplicação de uma rede multicamadas Perceptron tradicional que utiliza os atributos extraídos pelas camadas convolucionais para gerar uma decisão como a atribuição de rótulos, por exemplo. Contendo na última camada a função de ativação *softmax*, seu treinamento se dá pelo algoritmo *backpropagation*. Em outras palavras, a adição da camada totalmente conectada é uma maneira de aprender combinações não lineares de características complexas, conforme representado pela saída da camada convolucional (KHAN et al., 2020; STANFORD, 2019; SAHA, 2018). Ao passar pela última camada de *pooling*, o correspondente volume de saída é "achatado"(do inglês, *flatten*) e convertido para um vetor coluna que alimentará a rede Perceptron, na Figura 25 é representada a arquitetura típica de uma camada totalmente conectada para especificação de quatro classes (rótulos).

Nesta camada, por se tratar na realidade de uma rede multicamadas Perceptron, as técnicas preventivas da ocorrência de *overifitting* empregadas lá também são válidas aqui. Uma delas é a parada antecipada (do inglês, *early stopping*), consiste em determinar o ponto de treinamento onde a capacidade de generalização da rede começa a cair. Ao longo das épocas, o erro de treinamento decai, porém para amostras não apresentadas ao modelo (conjunto de validação) em dado momento o erro de validação passa a aumentar, então aí o treinamento é interrompido e o número adequado de épocas para o treinamento é obtido. O hiperparâmetro *patience* determina o número de épocas no qual é feito esse monitoramento dos erros de treino e validação. Outra estratégia de regularização é a conhecida como *Dropout*, introduzida por (SRIVASTAVA et al., 2014), que segundo a probabilidade p anula ativação de neurônios escolhidos aleatoriamente durante o treina-

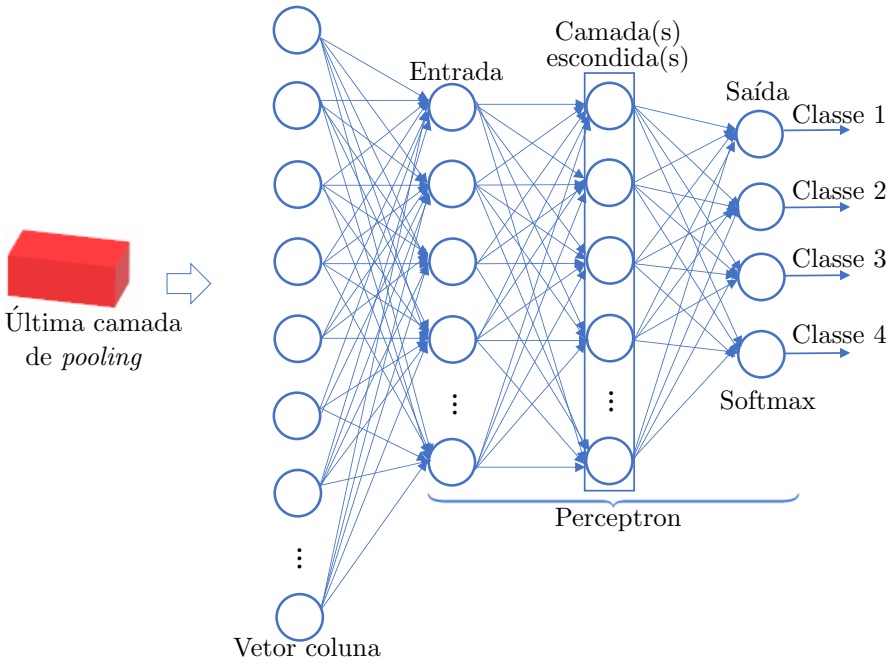


Figura 25: Representação típica de conexões na camada totalmente conectada

mento. Ou seja, a cada vez é treinada uma rede diferente, essa restrição força a rede a aprender características mais robustas, em vez de depender da capacidade preditiva de um pequeno subconjunto de neurônios na rede (IOFFE; SZEGEDY, 2015).

Por outro lado, a técnica *Data Augmentation* age diretamente sobre uma causa central do *overifitting*, a pequena quantidade de dados disponíveis no conjunto de treinamento. Supondo que mais informações podem ser extraídas do conjunto de dados original por meio da ampliação artificial do conjunto de dados. O aumento dos dados ocorre com a utilização de alterações na imagem original, como rotação vertical/horizontal, escalonamento, distorção ou até mesmo transformações de cores (SHORTEN; KHOSHGOFTAAR, 2019).

4 MODELO PROPOSTO

O sistema proposto é composto por etapas principais bem definidas: modelagem em elementos finitos, processamento de imagem, extração de atributos e aplicação de algoritmos de aprendizado de máquina (classificadores e regressores) para obtenção da classificação da severidade de defeitos e da profundidade destes defeitos, respectivamente. Também integra o modelo a utilização de redes neurais convolucionais para a predição da severidade dos defeitos, na Figura 26 é mostrado o esquema com os estágios fundamentais. Note que o trecho tracejado indica modelagem em elementos finitos, porém em uma aplicação prática haveria o mapeamento do potencial elétrico com a técnica *Potential Drop* da amostra sob investigação.

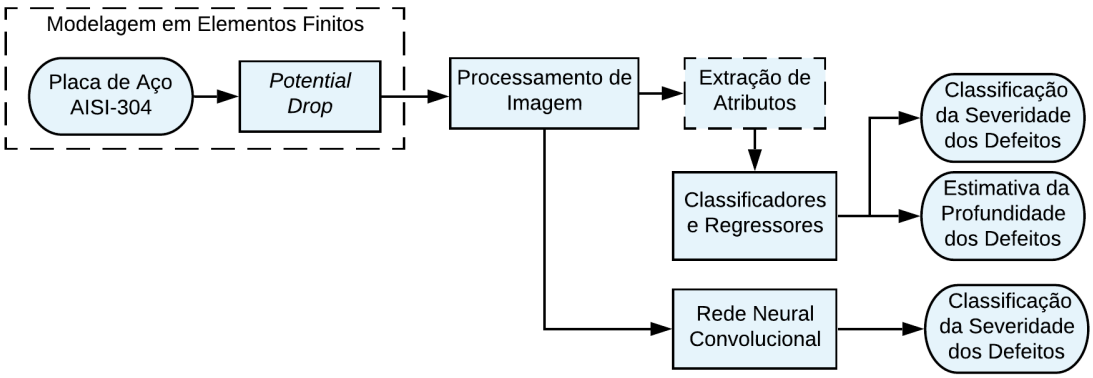


Figura 26: Modelo proposto

4.1 Modelagem em Elementos Finitos

As simulações realizadas durante o desenvolvimento deste trabalho foram obtidas de forma automatizada utilizando o *software* COMSOL®, através de sua interface com o MATLAB®. Este processo exige a criação de um modelo base, onde são definidos parâmetros relacionados à geometria e malha, algoritmo de resolução e tolerância desejada. Conforme Figura 27, este procedimento é interativo; ao ser considerado como satisfatório, o modelo é convertido para um algoritmo em MATLAB®, o que resulta no código base. Esta análise de desempenho é feita com base no tempo de convergência e no comportamento gráfico das grandezas de interesse, como potencial elétrico e seu gradiente. Em seguida, o código base é utilizado em uma rotina principal que permite determinar o número de simulações desejadas, especificar as características dos defeitos (dimensões e

localizações), dentre outros ajustes. Na Tabela 2 estão descritos os principais parâmetros de modelagem.

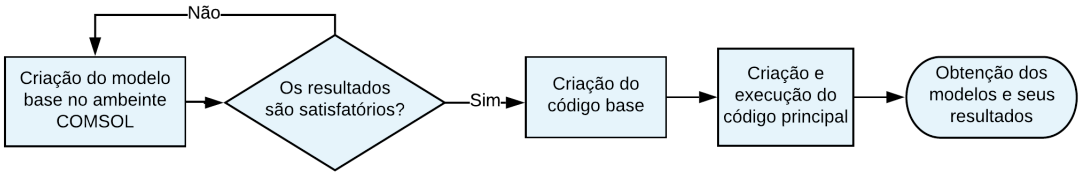


Figura 27: Fluxo do processo de automatização das simulações

Tabela 2: Descrição dos parâmetros de simulação

Material da placa	Aço inox AISI-304
Dimensões da placa	300x300x10 mm
Corrente contínua injetada	10A
Posição dos eletrodos injetores	$(x, y) = (\pm 120, 0)\, mm$

Conforme mencionado por (RYU et al., 2020), é necessário manter alguma distância entre os pontos de medição e os eletrodos de corrente, pois na vizinhança dos eletrodos injetores — entorno dos pontos $(x, y) = (\pm 120, 0)$ para o arranjo empregado, a densidade de corrente pode não estar uniformemente distribuída. Isto implicaria também na não uniformidade do gradiente de potencial nesta região, como pode ser visto na Figura 28, o que levaria à análises equivocadas. Portanto, convém estabelecer um limite de proximidade entre os alvéolos simulados e os pontos de contato dos eletrodos injetores.

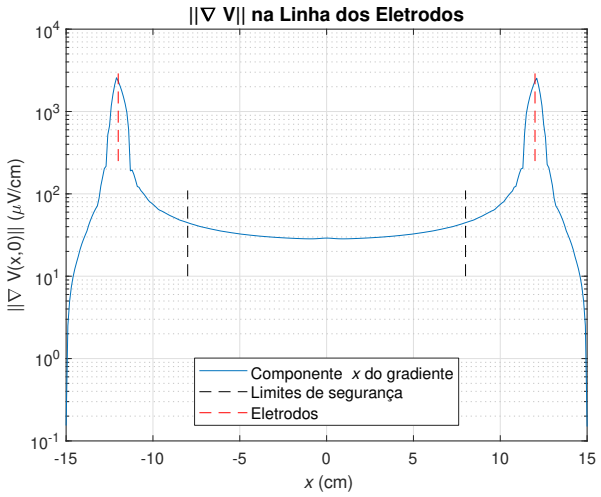


Figura 28: Comportamento do gradiente de potencial ao longo da linha dos eletrodos

Em testes preliminares verificou-se que a área de interferência dos eletrodos se estendia ao longo de um raio de $3\,cm$, assim por segurança, ficou determinado o valor de afastamento de $4\,cm$ os pontos de contato dos eletrodos injetores como limites de segurança, onde observa-se um perfil mais suave no gradiente de potencial. Na Figura 29 são ilustradas as áreas da placa utilizadas para as simulações (hachuradas), em branco as regiões evitadas. Visando mitigar a influência das bordas da placa, os eletrodos injetores são dispostos sobre o eixo x e à uma distância de $3\,cm$ das laterais da peça; quanto aos alvéolos simulados, pela mesma razão, embora sejam realizadas simulações considerando a área hachurada, há preferência pela região central da placa. Do ponto de vista dos modelos de aprendizado, é interessante que haja diversidade entre as amostras, por isso são mantidas simulações em regiões mais afastadas do centro.

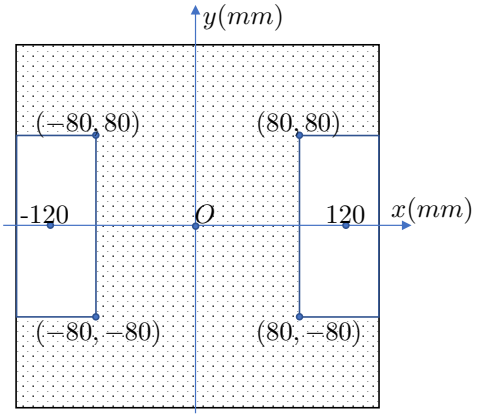
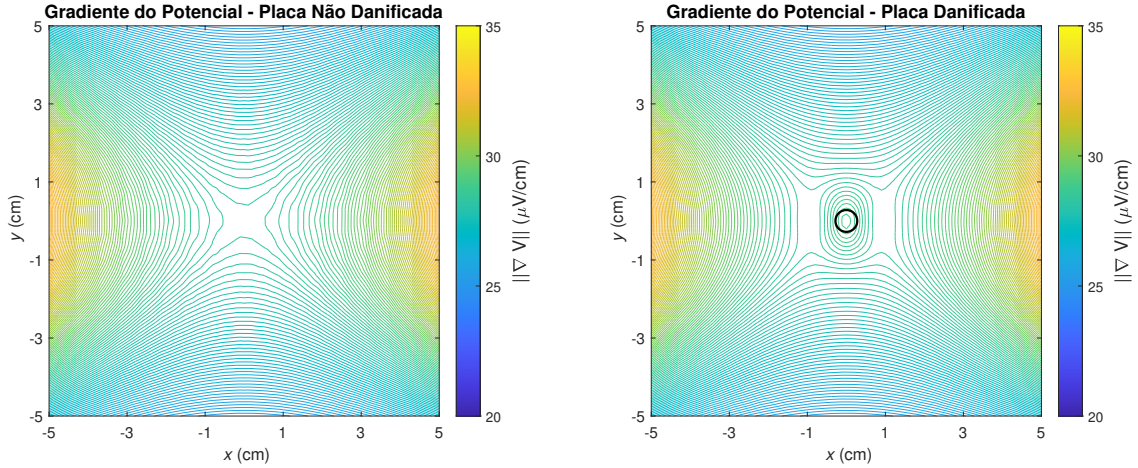


Figura 29: Área utilizada para simulações

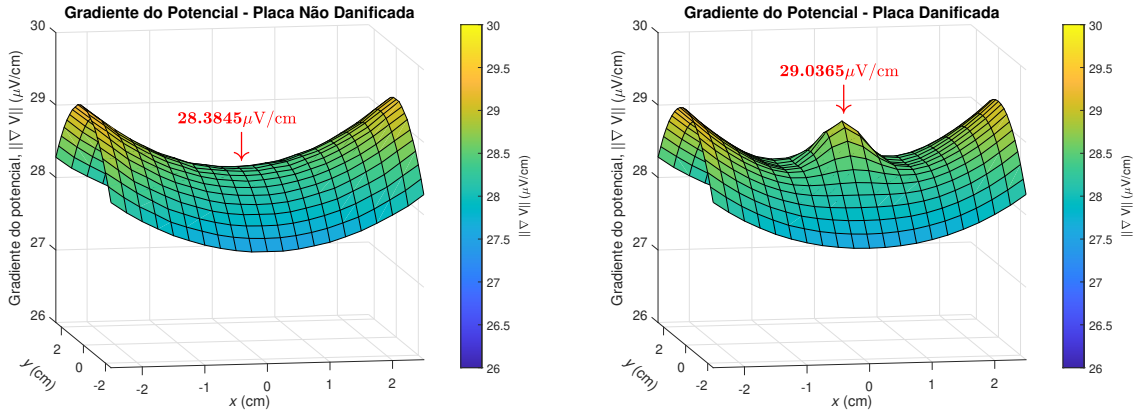
4.2 **Processamento de Imagem**

Iniciando com um caso simples, apresentando apenas um alvéolo no centro da placa metálica na Figura 30 é mostrada a comparação da magnitude do gradiente de potencial entre uma placa sem defeito (a) e outra danificada (b), cuja localização do alvéolo é indicada pelo círculo em preto. Na Figura 31 pode ser feita a comparação por meio de gráficos em três dimensões, os valores no centro da placa estão em destaque. Como pode ser verificado, não apenas a área do alvéolo é evidenciada, mas toda a sua vizinhança é eletricamente influenciada pela presença do defeito, originando o que foi denominada como região de influência.



(a) placa sem defeitos (b) placa com um defeito

Figura 30: Curvas de isogradiente do potencial elétrico



(a) placa sem defeitos (b) placa com um defeito

Figura 31: Comparação em três dimensões do gradiente do potencial elétrico

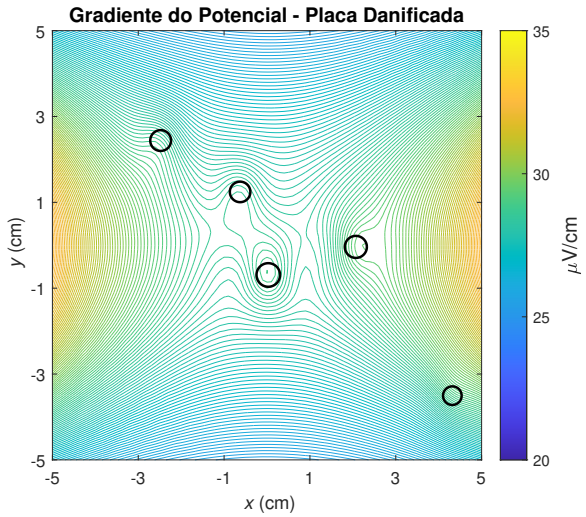
O estágio inicial no processamento de imagens é a obtenção do que foi denominado como "imagem elétrica", este tipo de imagem representa a assinatura dos defeitos. Entretanto, primeiramente é necessário definir o conceito de assinatura, esta é obtida pela divisão pontual entre o mapeamento do gradiente da placa com defeito pelo da placa sem defeito, conforme mostrado na Equação 27. Onde ∇V_{cd} e ∇V_{sd} se referem às situações com defeito e sem defeito respectivamente. Observando a Figura 31, a variação na magnitude do gradiente entre as situações com e sem defeito na região de influência é numericamente pequena, então a razão entre esses valores é próxima da unidade.

$$Assinatura(x,y) = \begin{cases} \dfrac{|\nabla V_{cd}(x,y)|}{|\nabla V_{sd}(x,y)|}, & \text{se } \dfrac{|\nabla V_{cd}(x,y)|}{|\nabla V_{sd}(x,y)|} \geq 1 \\ 1, & \text{caso contrário} \end{cases} \tag{27}$$

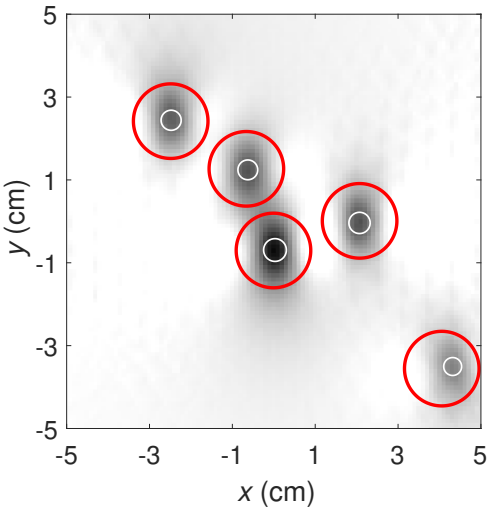
A imagem elétrica é uma representação em escala de cinza e foi idealizada de modo que dissimilaridades entre as assinaturas fossem indicadas pelos mais tons escuros. Em outras palavras, quanto uma região apresenta tom mais próximo do preto, maior a alteração em relação à placa sem defeito. A definição matemática da imagem elétrica é apresentada na Equação 28. Sendo $Assinatura_{ref}$ o valor da assinatura em $(x, y) = (0, 0)$ para o caso do alvéolo com profundidade de 2,79 mm mostrado na Figura 30 (b) que é a referência de máximo defeito, com respeito à área de abertura, (24,5 mm^2) e limiar da classe B5 indicado pela norma ASTM G46-94 (ASTM, 2018).

$$Imagem\ elétrica(x, y) = \left(1 - \frac{Assinatura(x, y) - 1}{Assinatura_{ref} - 1}\right) \cdot 255 \tag{28}$$

Tratando agora de um caso com múltiplos defeitos, na Figura 32 (a) é apresentada a distribuição do valor absoluto do gradiente de potencial superficial em uma placa de aço inox AISI-304 apresentando 5 defeitos, conforme indicações e cujas profundidades variam entre 2,17 mm e 2,75 mm. A imagem elétrica correspondente deste caso é mostrada na Figura 32 (b). Conforme pode ser observado, a localização de cada defeito é indicada pelos círculos brancos; a região de influência de cada alvéolo (ilustradas por círculos em vermelho) pode refletir em uma interação mútua entre os defeitos e dificultar a distinção entre eles.



(a) Mapeamento do gradiente de potencial superficial



(b) Imagem elétrica, regiões de influência em destaque

Figura 32: Comparação mapeamento de gradiente do potencial elétrico e imagem elétrica para a placa com cinco alvéolos

Portanto, a direta aplicação de um método de segmentação, como o limiar de Otsu (Otsu, 1979) que assume um simples discriminante e considera apenas um objeto não traria resultados adequados. Um exemplo desta situação é mostrado na Figura 33; então, para contornar este caso é proposta a abordagem para a segmentação de imagens elétricas composta pelos seguintes operadores: dilatação, transformada de distância euclidiana e transformação (segmentação) de *watershed*. Na Figura 34 é mostrado o fluxo de trabalho empregado no processamento de imagens elétricas.

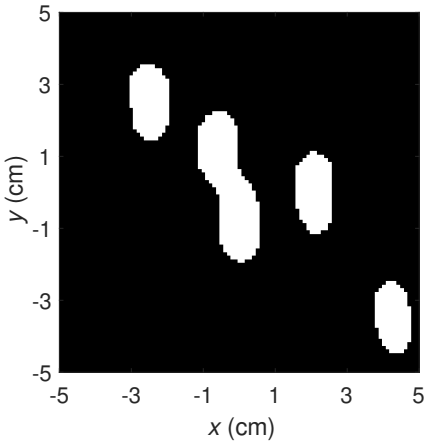


Figura 33: Aplicação direta do limiar de Otsu

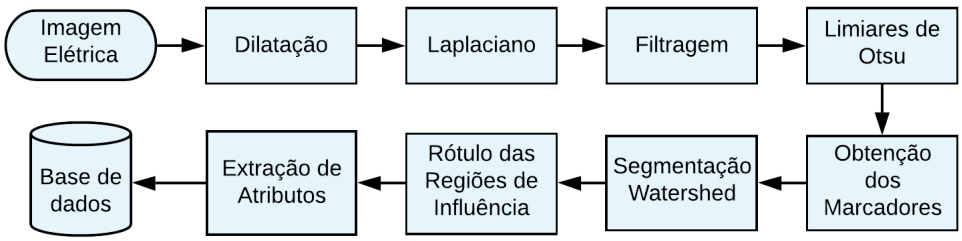


Figura 34: Fluxo do processamento de imagem

De acordo com a Figura 34, o primeiro estágio de processamento é a dilatação, que é usada para preservar os defeitos menores. O operador Laplaciano é usado para realçar as bordas das regiões de influência, porém, por ser um operador diferencial pode intensificar o ruído. O que faz necessário o emprego de uma etapa dedicada à filtragem contendo um filtro passa baixas e um filtro de mediana que preserva as bordas da imagem (DEMIROVIĆ, 2019). O princípio do filtro de mediana é substituir o valor de cada *pixel* pela mediana dos valores em dada vizinhança, em vez de usar a média correspondente

(TAN; JIANG, 2019). Na sequência, veja que o limiar de Otsu ainda é utilizado, entretanto, como passo intermediário. Na Figura 35 é mostrado o resultado neste ponto, note que já há maior definição na forma das regiões de influência quando feita a comparação com a Figura 33.

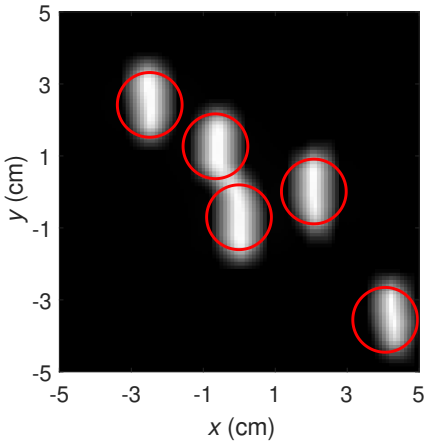


Figura 35: Limiar de Otsu como processo intermediário

No estágio seguinte, por meio da transformada de distância euclidiana, são encontradas áreas específicas que permitem a identificação dos centros das regiões de influência. A transformada de distância euclidiana calcula a distância de cada pixel interior à uma região fechada até a fronteira (segundo plano) (ANGAYARKANNI; DURAIRAJ, 2015). Ou seja, os *pixels* no centro de cada região apresentam maior distância à borda, assim os valores máximos fornecidos pela transformada euclidiana de distância com as respectivas localizações (marcadores) são utilizados. Na Figura 36 são apresentados os marcadores obtidos para o exemplo abordado, conforme indicações em vermelho.

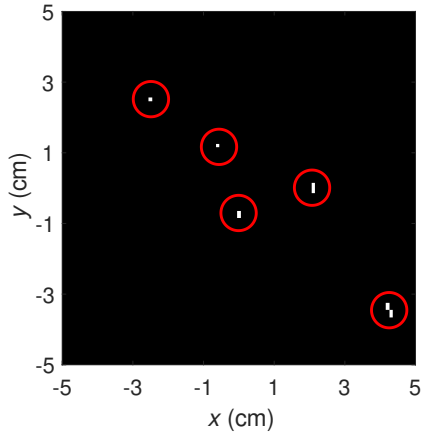
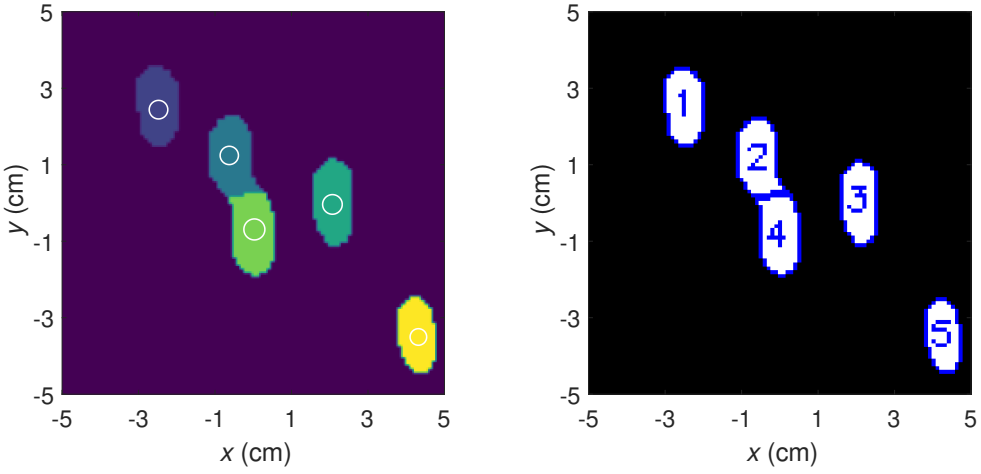


Figura 36: Marcadores obtidos

Seguindo a Figura 34, a última etapa de processamento é a segmentação *watershed*, um algoritmo que combina técnicas de região de crescimento e detecção de bordas, cujo título se refere em uma tradução literal à bacias hidrográficas. A definição associada à esse termo geográfico pode ser aplicada aplicada da seguinte forma: uma paisagem ou relevo topográfico que é inundado pela água, sendo as bacias hidrográficas as linhas de divisão dos domínios de atração de chuva que caem sobre a região (YOUNG, 1983). Uma abordagem alternativa é imaginar a paisagem imersa em um lago, com buracos perfurados em mínimos locais. As bacias hidrográficas se enchem de água a partir desses mínimos locais e, nos pontos em que a água proveniente de diferentes bacias se encontra, as barragens são construídas. Quando o nível da água atinge o pico mais alto da paisagem, o processo é interrompido. Como resultado, a paisagem é dividida em regiões ou bacias separadas por barragens, chamadas bacias hidrográficas (ROERDINK; MEIJSTER, 2000).

Seu funcionamento pode ser descrito como um aglutinador de pixels no entorno de regiões de valores mínimos da imagem, então, como os marcadores representam valores máximos, basta que eles sejam multiplicados por (-1) para que se obtenha os valores mínimos necessários. Como resultado, na segmentação *watershed* as fronteiras de agrupamentos adjacentes são precisamente dispostas ao longo da maior taxa de variação da imagem (SOILLE, 2003); na Figura 37 (a) é mostrada a saída da segmentação *watershed*, onde cada cor representa uma região de influência detectada e em branco estão destacadas as localizações dos alvéolos. E, por fim, na Figura 37 (b) tem-se cada região de influência devidamente rotulada.



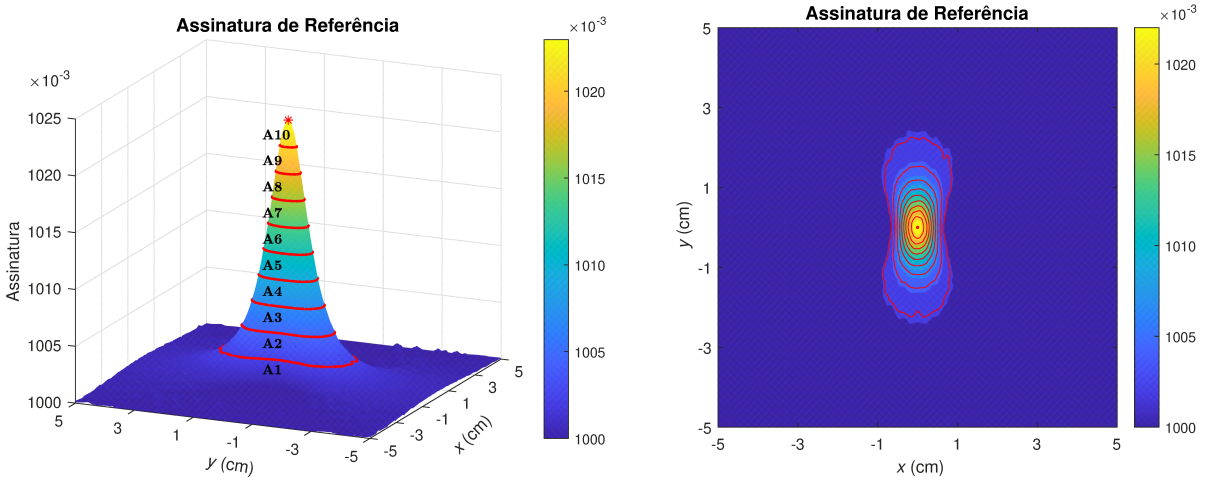
(a) resultado da segmentação *watershed* (b) imagem elétrica rotulada

Figura 37: Imagem elétrica processada

4.2.1 **Extração de Atributos**

Uma vez concluída a obtenção dos rótulos das regiões de influência, inicia-se a etapa de extração de atributos, para tal, foi utilizada a assinatura do alvéolo de referência apresentado na Figura 31 (b). Na Figura 38 (a) esta assinatura está representada em três dimensões com a indicação da localização de alguns atributos extraídos. Esta referência consiste em um dano com profundidade de 2,79 mm, que corresponde ao limiar da classe de maior severidade pela norma ASTM G46-84 (ASTM, 2018) e com centro em $(x, y) = (0, 0)$.

Parte da extração de características é feita em termos de porcentagem da assinatura produzida pelo alvéolo de referência, na faixa de 10% a 90%, utilizando passo de 10 pontos percentuais. Na Figura 38 (a) estes pontos de corte ($A1 \ldots A9$) estão indicados em vermelho. $A10$ indica a localização do valor máximo, que também é um atributo utilizado. Na Figura 38 (b) é mostrada a representação em duas dimensões da assinatura do alvéolo de referência, com a indicação dos pontos de extração de atributos, sendo $A1$ a linha vermelha mais afastada do centro.



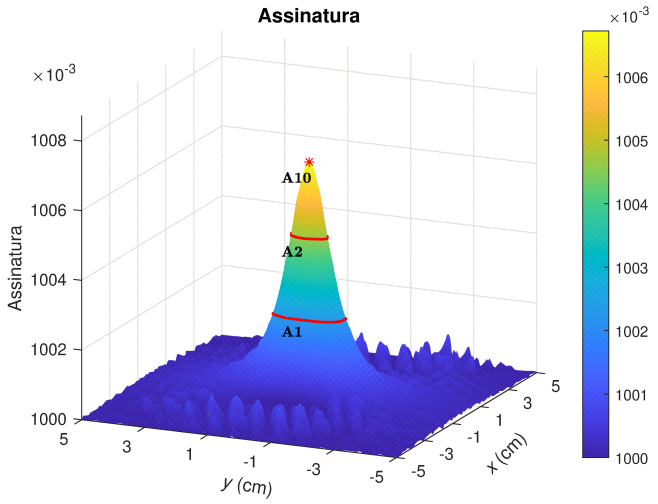
(a) representação tridimensional da assinatura de referência

(b) representação bidimensional da assinatura de referência

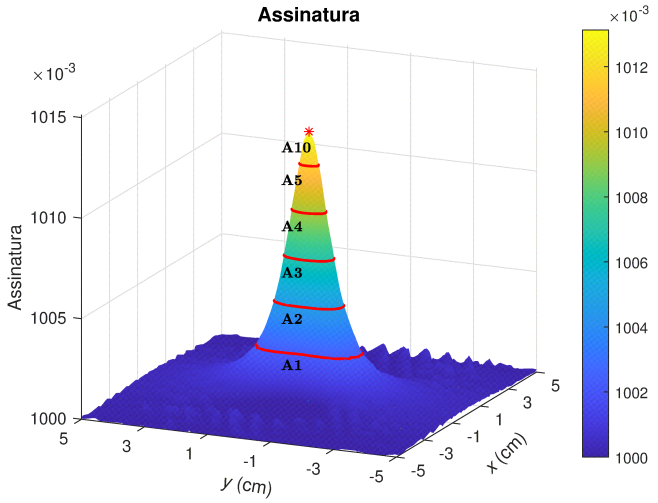
Figura 38: Ilustração de pontos de corte para extração de atributos do alvéolo de referência

Cada porção segmentada dá origem a uma Região de Interesse (RI), que consiste em uma caixa delimitadora de 32x32 *pixels* com centro coincidente com o da região segmentada; as dimensões da caixa delimitadora foram definidas de modo a comportar defeitos da maior severidade listada pela norma ASTM G46-84 (ASTM, 2018). Uma

forma de visualização do processo de extração de características pode ser verificada na Figura 39. A zona cujos limites são dados pela região segmentada é observada na imagem elétrica, de acordo com os pontos de corte indicados por $A1 \ldots A9$ são associados os atributos de área (Área1... Área9). Quanto menor o defeito, os atributos relacionados aos valores mais altos da assinatura são nulos; por outro lado, como $A10$ está ligado ao valor de pico, persiste nas diferentes situações. Adicionalmente, completando o conjunto de 11 características extraídas, foi obtido o raio da circunferência que delimita a região segmentada.



(a) representação tridimensional da assinatura para um alvéolo com profundidade de 1,86 mm



(b) representação tridimensional da assinatura para um alvéolo com profundidade de 2,31 mm

Figura 39: Comparação das assinaturas de alvéolos com diferentes profundidades

Todo o processamento de imagens, incluindo a extração de atributos, é feito utili-

zando as ferramentas *Opencv-Python* 4.3.0 (BRADSKI, 2000), *Scipy* 1.4.1 (VIRTANEN et al., 2020) e *Skimage* 0.17.2 (WALT et al., 2014).

4.3 Regressores e Classificadores

O conjunto de atributos obtidos pode ser apresentado ao algoritmos tradicionais de aprendizado de máquinas descritos nos Itens 3.1 - 3.5. Com exceção da Regressão Linear que foi utilizada com as configurações padrões, todos os demais algoritmos tiveram o ajuste de hiperparâmetros baseado na busca em grade com o método conhecido como *k-fold*, onde o conjunto de treinamento é divido em k pastas. Destas, k-1 são usadas para treinamento e uma para validação. O processo é repetido k vezes, de modo que em cada repetição os conjuntos de treinamento e validação sejam diferentes, o que produz k medidas de desempenho. Assim, o desempenho do modelo é a média das medidas obtidas em cada pasta (JAMES et al., 2014). Tanto para os regressores como os classificadores, as faixas de busca dos hiperparâmetros são mostradas no Apêndice A, na Tabela 17 e Tabela 18.

Além dos hiperparâmetros discutidos no Capítulo 3, as implementações da biblioteca *Sklearn* versão 0.23.2 (PEDREGOSA et al., 2011) que foi utilizada possuem outros ajustes, que foram mantidos como o padrão. Entretanto, para a Regressão Logística foi empregada a busca em grade para o número de iterações do algoritmo de otimização *n_iter* e a tolerância *tol* que atua como critério de parada.

4.4 Redes Neurais Convolucionais

Por se tratar de um modelo com extração automática de atributos, as redes neurais convolucionais recebem como entrada apenas as Regiões de Interesse, porém neste caso a assinatura é utilizada como representação da intensidade luminosa na região segmentada. Para a mesma região segmentada 2 da Figura 37 (b), na Figura 40 é mostrada o exemplo equivalente apresentado à rede convolucional.

Foram propostas três arquiteturas de redes neurais convolucionais (Conv 1, Conv 2 e Conv 3). A Conv 1, com menor complexidade, apresenta apenas uma camada convolucional com 32 filtros e duas camadas totalmente conectadas. Por sua vez, a Conv 2 é uma extensão da Conv 1 contando com duas camadas convolucionais e de agrupamento,

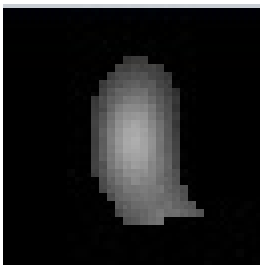


Figura 40: Exemplo de imagem apresentada à rede convolucional

seu modelo é mostrado na Figura 41. Em alternativa, a rede Conv 3 – representada na Figura 42 – foi inspirada nas camadas iniciais da arquitetura VGG (*Very Deep Convolutional Networks for Large-Scale Image Recognition*) (SIMONYAN; ZISSERMAN, 2014) que obteve a acurácia de 92,7% na tarefa de classificação, levando os desenvolvedores ao segundo lugar do *ImageNet Challenge* 2014. As camadas de saída dos modelos propostos tiveram o número de neurônios fixados em 4 (uma para cada classe) e com função de ativação *softmax*.

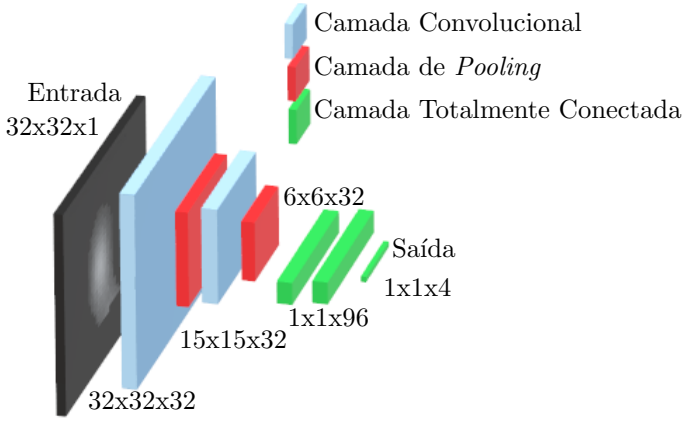


Figura 41: Arquitetura da rede Conv 2

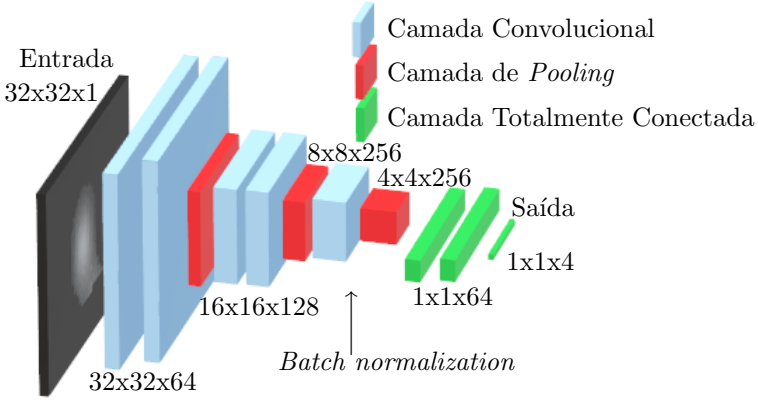


Figura 42: Arquitetura da rede Conv 3

Por conta do menor custo computacional requerido pela redes Conv 1 e Conv 2, a busca em grade de hiperparâmetros pôde ser feita considerando combinações de números de neurônios das camadas totalmente conectadas e de taxas de *dropout*, enquanto na Conv 3 a busca foi pela taxa de *dropout* mais adequada. Outra diferença entre os modelos se dá na utilização do *batch normalization*, nos dois mais simples esta técnica de regularização está presente após cada camada convolucional, já na Conv 3 é empregada apenas após a última camada de convolução. Os valores dos hiperparâmetros testados na busca em grade são mostrados na Tabela 19, no Apêndice A. Para a implementação das redes convolucionais foram empregadas as bibliotecas *Keras* 2.4.3 (CHOLLET et al., 2015) e *TensorFlow* 2.3.0 (ABADI et al., 2015), com execução em placa gráfica NVIDIA Tesla T4.

5 ESTUDO DE CASO

Conforme descrito no Item 4.1 e seguindo as classificações estabelecidas pela norma ASTM G46-94 (ASTM, 2018) foram geradas 3.300 simulações resultando em 16.500 alvéolos, esta etapa precisou de 76 horas para ser concluída. Na Tabela 3 está descrita a composição do banco de dados utilizado, dos alvéolos simulados, 13.545 regiões de influência foram identificadas pelo processamento de imagens descrito no Item 4.2. E destes, 90% foram utilizados para treinamento e os 10% restantes foram empregados no conjunto de teste com a finalidade de avaliar o desempenho dos algoritmos em amostras não apresentadas durante o treinamento. Também é apresentada a *taxa de acerto*, dada pela relação do número de regiões segmentadas e a quantidade de alvéolos simulados para cada classe. Na Figura 43 são mostradas as distribuições das áreas dos alvéolos simulados, bem como as correspondentes áreas às quais as regiões de influência foram identificadas.

Tabela 3: Descrição do banco de dados

Classificação ASTM G46-94	**Área** (mm)		**Profundidade Equivalente** (mm)		**Alvéolos Simulados**	**Regiões Segmentadas**	**Taxa de Acerto (%)**
	Mín	**Máx**	**Mín**	**Máx**			
B1	0,50	1,99	0,399	0,796	0	-	-
B2	2,00	7,99	0,798	1,595	4.238	2.455	57,93
B3	8,00	12,49	1,596	1,994	3.261	3.193	97,91
B4	12,50	24,49	1,995	2,792	6.001	5.501	91,67
B5	24,50	-	2,793	-	3.000	2.396	79,87

Nota: em testes prévios não foram detectadas regiões de influência para alvéolos da classe B1

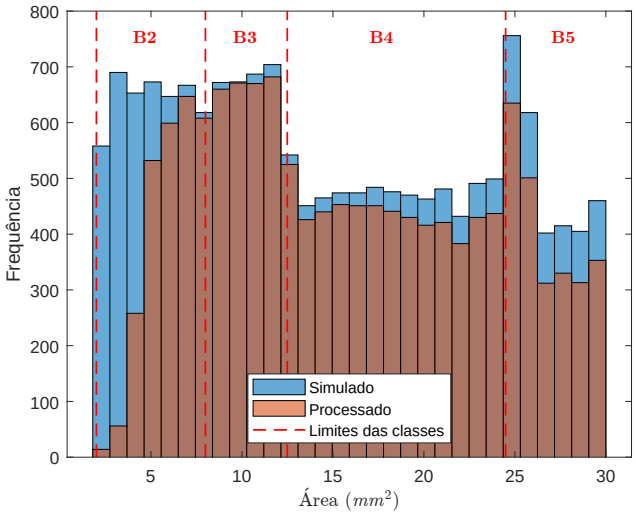


Figura 43: Distribuição dos alvéolos simulados e correspondentes regiões de influência obtidas

O ajuste dos hiperparâmetros mais relevantes de cada algoritmo é feito utilizando a busca em grade e a técnica *k-fold*, apresentada no Item 4.3; uma vez obtidos os hiperparâmetros mais adequados, os modelos são treinados usando todo o conjunto de treinamento. Na Tabela 4 são descritos os hiperparâmetros manipulados nos problemas de estimativa da profundidade e classificação da severidade dos alvéolos. Divido em três aspectos principais, este Capítulo reúne os resultados obtidos ao longo do trabalho.

Tabela 4: Descrição de hiperparâmetros

Algoritmo	**Hiperparâmetros**
Regressão Linear	N/A
Regressão Logística	*n_iter*: número de iterações para convergência *tol*: tolerância, critério de parada
Árvore de Decisão	p: profundidade *critério*: função ganho de informação*
k-*Nearest Neighbors*	k: número de vizinhos
Vetor de Suporte	C: regularização *kernel*: função de transformação de espaço
Gradient Boosting	η: taxa de aprendizado n: numero de estimadores
Random Forest	p: profundidade das árvores n: número de estimadores

* Válido para classificação da gravidade dos alvéolos. Na estimativa da profundidade dos alvéolos o critério adotado é o erro quadrático médio.

5.1 Estimativa da Profundidade dos Alvéolos

Nesta análise foram conduzidos experimentos considerando o efeito da normalização *Z-score*, que torna os dados com média nula e desvio padrão unitário, e, também a influência da seleção de atributos baseada na correlação entre as variáveis. Apoiado no texto de (SEGHIER et al., 2020) que apesar de tratar de corrosão externa, também lida com a estimativa de profundidade de alvéolos, foram adotadas como medida de desempenho o Erro Quadrático Médio (EQM), o Erro Absoluto Médio ($\overline{EA}$), Erro Absoluto Máximo (EA_{max}) e o Coeficiente de Determinação R^2, definidos nas Equações 29 - 32; assim o objetivo é obter os mínimos valores das métricas EQM e relacionadas ao erro absoluto, já para o Coeficiente de Determinação espera-se que atinja valores o mais próximo da unidade possível.

$$EQM = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{29}$$

$$\overline{EA} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{30}$$

$$EA_{max} = arg\,max\left(|y_i - \hat{y}_i|\Big|_{i=1}^{N}\right) \tag{31}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2} \tag{32}$$

Onde:

y_i – profundidade real da amostra i;

$\hat{y}_i$ – profundidade estimada para a amostra i;

$\overline{y}$ – profundidade real média;

N – número de amostras do conjunto de teste.

Para maior interpretação dos resultados obtidos, estas métricas podem se agrupadas formando a razão R^2/EQM baseado em (SEGHIER et al., 2020) e definindo o $Fator\,de\,erro = EA_{max} \times \overline{EA} \times 100$. Assim, na razão R^2/EQM deseja-se obter o máximo valor realizável, uma vez que objetiva-se maximizar o coeficiente de determinação e minimizar o erro quadrático médio. Por outro lado, o fator de erro deve ser minimizado. O erro de estimativa, ou simplesmente *erro* é dado conforme Equação 33.

$$e_i = y_i - \hat{y}_i \tag{33}$$

5.1.1 Experimentos sem Normalização

No primeiro cenário observado, foram utilizados os dados da forma como o processamento de imagens os disponibiliza, ou seja, todos os atributos não normalizados foram apresentados aos algoritmos, na Tabela 5 são mostrados os resultados obtidos. A partir disto, os dados podem ser interpretados graficamente, conforme mostrado na Figura 44. Utilizando a razão R^2/EQM e no fator de erro é evidenciada a diferença de desempenho do *Gradient Boosting* e *Random Forest* sobre os demais algoritmos, com vantagem para o último.

Na Figura 45 são apresentadas a dispersão do erro de estimativa e seu histograma para o modelo com maior desempenho — *Random Forest*; em 84,01% das previsões a diferença absoluta entre a profundidade real e estimada é igual ou inferior a 0,1 mm.

Tabela 5: Desempenho dos regressores sem normalização e sem seleção de atributos

Regressor	**Hiperparâmetros**	**Métricas**			
		EQM ($\times 10^{-3}$)	**R^2** ($\times 10^{-1}$)	**EA_{max}** ($\times 10^{-1} mm$)	**$\overline{EA}$** ($\times 10^{-2} mm$)
Regressão Linear	N/A	17,56	9,41	9,90	8,75
Árvore de Decisão	$p = 6$	11,88	9,60	8,50	6,82
k-*Nearest Neighbors*	$k = 3$	16,31	9,45	10,65	8,04
Vetor de Suporte	$C = 0{,}4$; $kernel =$ RBF	18,91	9,37	9,78	8,54
Gradient Boosting	$\eta = 0{,}4$; $n = 250$	8,93	9,70	9,16	5,87
Random Forest	$\mathbf{p = 10; n = 80}$	**8,77**	**9,71**	**8,80**	**5,83**

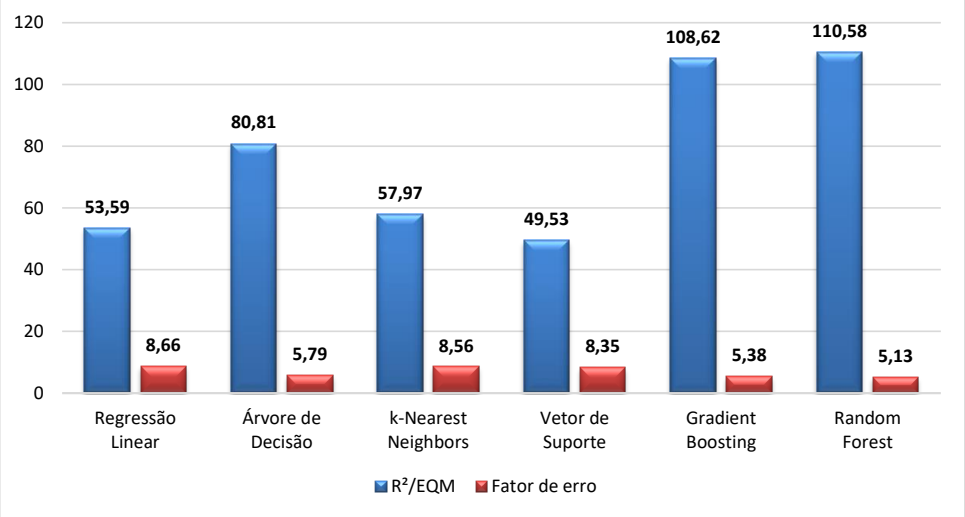
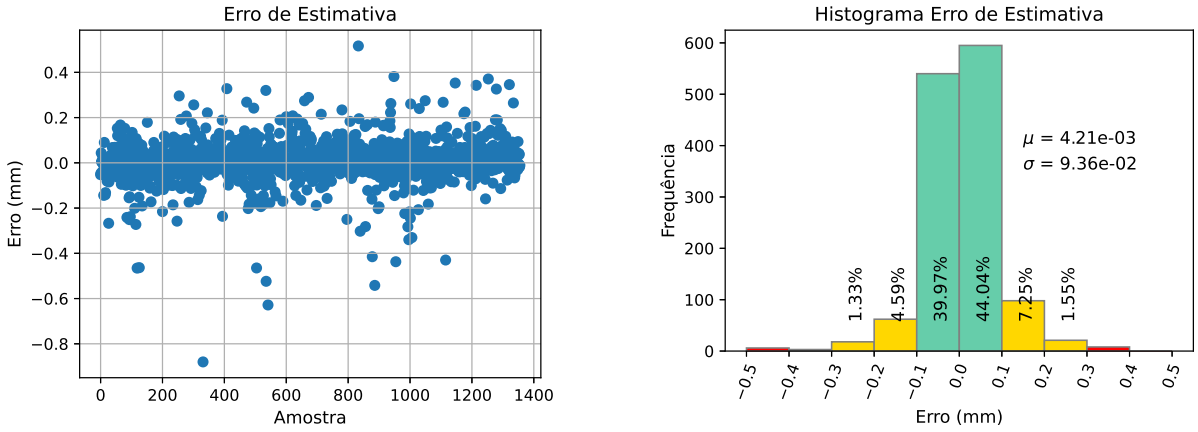


Figura 44: Resultados obtidos, sem normalização e sem seleção de atributos



(a) distribuição do erro de estimativa (b) histograma do erro

Figura 45: Análise do erro com *Random Forest* sem seleção de atributos

Com o intuito de observar o comportamento dos modelos tanto em termos de desempenho como de complexidade, é feita a seleção de atributos com base na correlação linear. O limiar adotado (0,92) permite preservar o raio da circunferência que delimita a região segmentada, pois este exprime alta correlação linear com a profundidade real (coeficiente de 0,79). Na sequência, na Tabela 6 são apresentados os resultados para a situação com aplicação da seleção de atributos; devido ao grande número variáveis, a matriz de correlação linear é presentada no Apêndice A, Figura 66. Neste caso foram excluídos os atributos mais linearmente correlacionados do conjunto (Área2, Área3 e Área5); os dois primeiros apresentam entre si coeficiente de correlação igual a 0,98 e, respectivamente de 0,99 e 0,95 com a Área1. Esta, por sua vez, está mais ligada à profundidade real (coeficiente de correlação de 0,65) que os demais apresentando o índice inferior a 0,60. Já a variável Área5, é fortemente ligada à Área4 (0,94) e fracamente correlacionada com a profundidade dos alvéolos (0,01).

Tabela 6: Desempenho dos regressores sem normalização e com seleção de atributos

Regressor	**Hiperparâmetros**	**Métricas**			
		EQM ($\times 10^{-3}$)	$\mathbf{R^2}$ ($\times 10^{-1}$)	$\mathbf{EA_{max}}$ ($\times 10^{-1} mm$)	$\overline{\mathbf{EA}}$ ($\times 10^{-2} mm$)
Regressão Linear	N/A	17,65	9,41	9,87	8,80
Árvore de Decisão	$p = 6$	11,80	9,60	8,50	6,80
k-*Nearest Neighbors*	$k = 3$	16,42	9,45	11,50	8,10
Vetor de Suporte	$C = 0{,}4$; $kernel =$ RBF	18,42	9,38	9,74	8,50
Gradient Boosting	$\boldsymbol{\eta = 0{,}4;\ n = 150}$	**8,62**	**9,71**	**9,23**	**5,80**
Random Forest	$\boldsymbol{p = 10;\ n = 80}$	**8,77**	**9,71**	**8,92**	**5,83**

Por meio da análise da Figura 46 não é possível definir o melhor modelo nesta situação, uma vez que o *Gradient Boosting* apresentou maior valor da razão R^2/EQM, porém com maior fator de erro quando comparado com os resultados do *Random Forest*. Entretanto, observando o coeficiente de determinação idêntico nos dois regressores na Tabela 6, pode-se afirmar que na realidade os seus resultados são equivalentes. Isto também é verificado na Figura 47, onde 84,16% das previsões feitas pelo *Gradient Boosting* apresentam erro de até 0,1 mm, enquanto com *Random Forest* foi obtido 84,37%. Por outro lado, a seleção de atributos proporciona a redução número de estimadores no *Gradient Boosting*, de 250 para 150, na comparação com o caso sem seleção.

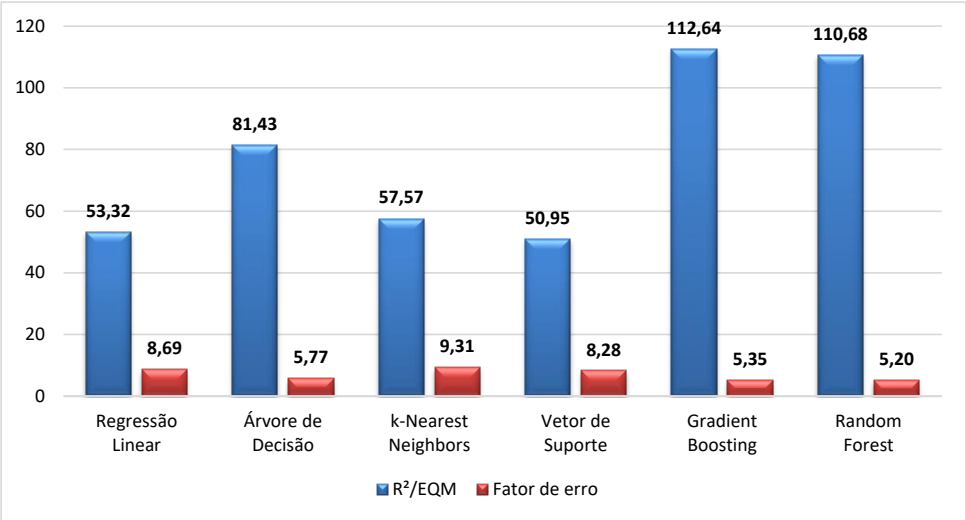
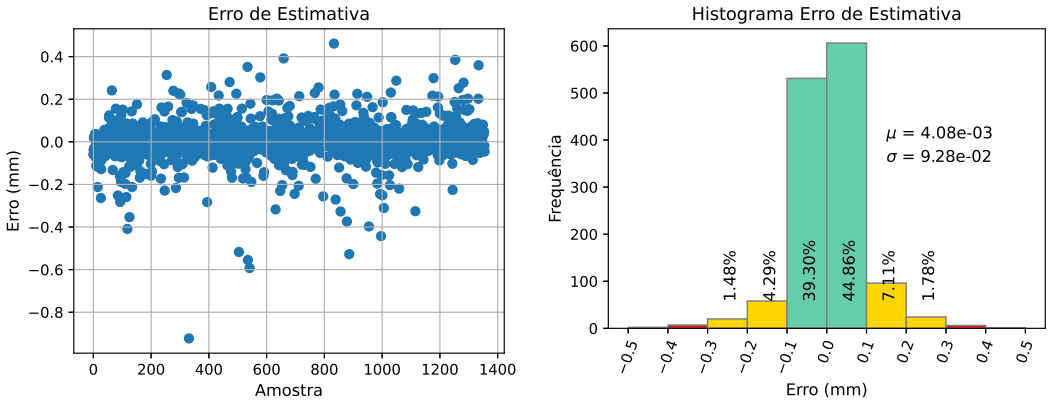
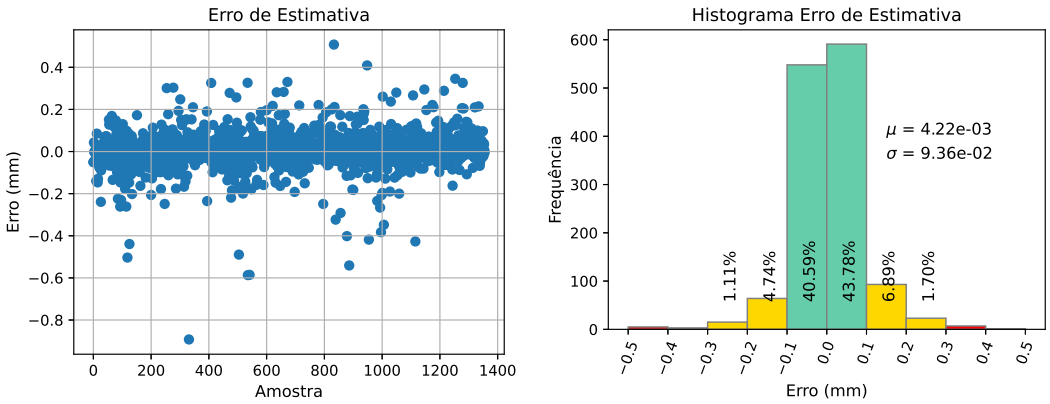


Figura 46: Resultados obtidos, sem normalização com seleção de atributos



(a) *Gradient Boosting*



(b) *Random Forest*

Figura 47: Análise do erro para *Gradient Boosting* e *Random Forest*

5.1.2 Experimentos com Normalização

Neste estudo foi aplicada a normalização *Z-score* e observada a influência da seleção de atributos; sendo primeiramente mostrados os resultados utilizando todos os atributos disponíveis, como pode ser observado na Tabela 7 e na Figura 48. A normalização, de forma geral, trouxe incremento de desempenho para os algoritmos analisados, isto é mais visível no erro quadrático médio principalmente com o k-*Nearest Neighbors* que apresentou redução de 39,91%. Porém, mais uma vez, o maior desempenho foi conseguido utilizando *Random Forest*, atingindo EQM = 8,52 e fator de erro igual a 5,06.

Tabela 7: Desempenho dos regressores com normalização sem seleção de atributos

Regressor	**Hiperparâmetros**	**Métricas**			
		EQM ($\times 10^{-3}$)	$\mathbf{R^2}$ ($\times 10^{-1}$)	$\mathbf{EA_{max}}$ ($\times 10^{-1} mm$)	$\overline{\mathbf{EA}}$ ($\times 10^{-2} mm$)
Regressão Linear	N/A	17,56	9,41	10,02	8,75
Árvore de Decisão	$p = 6$	12,20	9,59	8,50	6,84
k-*Nearest Neighbors*	$k = 7$	9,80	9,67	8,63	5,94
Vetor de Suporte	$C = 0{,}1$; $kernel =$ linear	17,58	9,41	10,80	8,77
Gradient Boosting	$\eta = 0{,}3$; $n = 100$	8,76	9,71	9,13	5,93
Random Forest	$\mathbf{p = 10}$; $\mathbf{n = 200}$	8,52	**9,71**	**8,80**	**5,75**

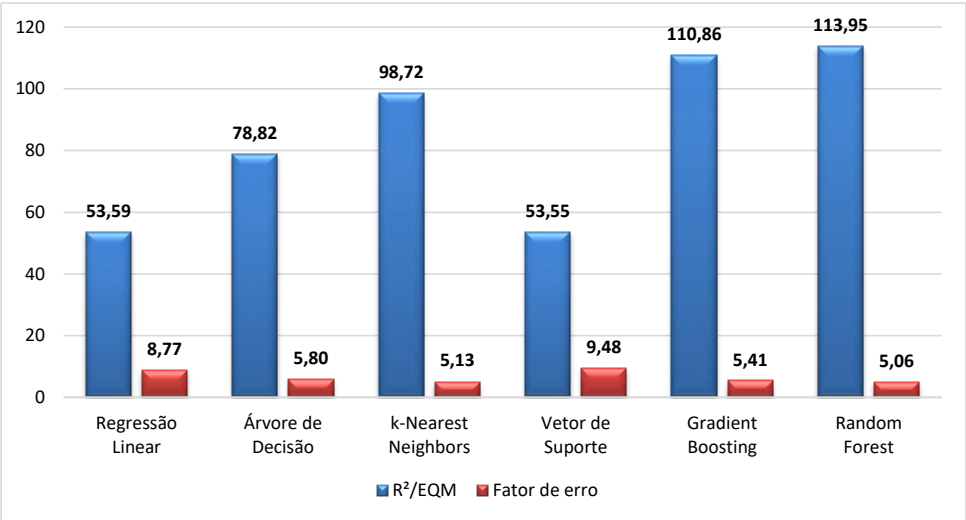
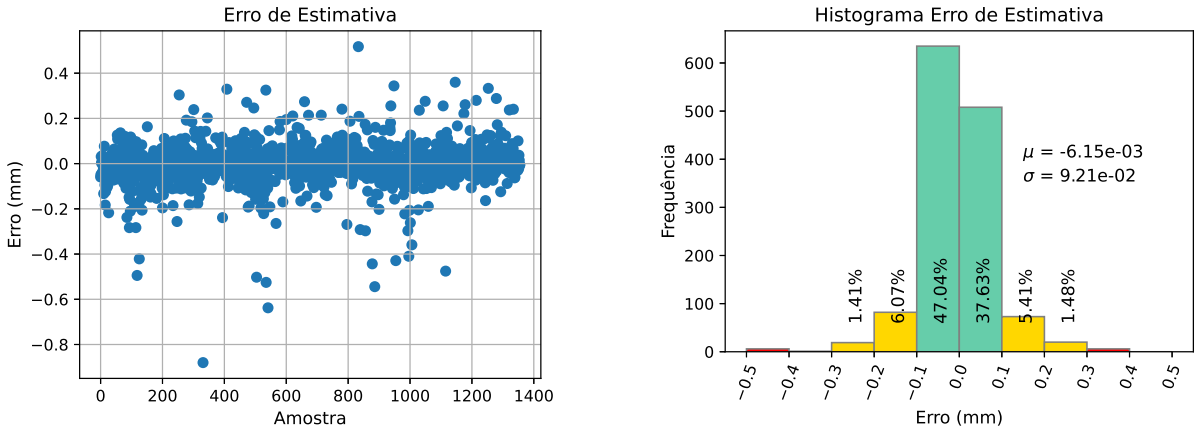


Figura 48: Resultados obtidos, com normalização sem seleção de atributos

Na Figura 49 são mostrados os principais resultados proporcionados pelo algoritmo *Random Forest*, onde 84,67% dos valores preditos indicam variação absoluta entre as profundidades estimadas e as reais inferior ou igual a 0,1 mm.



(a) distribuição do erro de estimativa

(b) histograma do erro

Figura 49: Análise do erro com *Random Forest* sem seleção de atributos

Por último, foi realizado o ensaio utilizando seleção de atributos com base na correlação linear, sendo retirados três deles. Os resultados desta etapa estão mostrados na Tabela 8. Pela análise da Figura 50 nota-se que houve aumento da razão R^2/EQM para a Árvore de Decisão, k-*Nearest Neighbors* e *Gradient Boosting* e queda para a Regressão Linear, Vetor de Suporte e *Random Forest*. Ao passo que o fator de erro apresenta leve variação em todos os algoritmos, com queda apenas para o Vetor de Suporte na comparação com o experimento sem seleção de atributos.

Tabela 8: Desempenho dos regressores com normalização com seleção de atributos

Regressor	**Hiperparâmetros**	**Métricas**			
		EQM ($\times 10^{-3}$)	**R^2** ($\times 10^{-1}$)	**EA_{max}** ($\times 10^{-1} mm$)	**$\overline{EA}$** ($\times 10^{-2} mm$)
Regressão Linear	N/A	17,65	9,41	9,98	8,80
Árvore de Decisão	$p = 6$	12,13	9,59	8,50	6,83
k-*Nearest Neighbors*	$k = 7$	9,68	9,68	8,79	5,96
Vetor de Suporte	$C = 0{,}1$; *kernel* = linear	17,64	9,41	10,59	8,83
Gradient Boosting	$\eta = 0{,}3$; n = 100	8,73	9,71	9,12	5,96
Random Forest	$\mathbf{p = 10}$; $\mathbf{n = 200}$	**8,59**	**9,71**	**8,95**	**5,77**

Mantendo a tendência vista nos demais experimentos, como pode ser observado na Tabela 9, os modelos baseados em *Gradient Boosting* e *Random Forest* apresentam os dois melhores desempenhos, e, novamente o regressor *Random Forest* apresenta resultados superiores. Na Figura 51 estão mostrados valores obtidos acerca do erro de estimativa, 84,81% dos erros cometidos possuem magnitudes de até 0,1 mm.

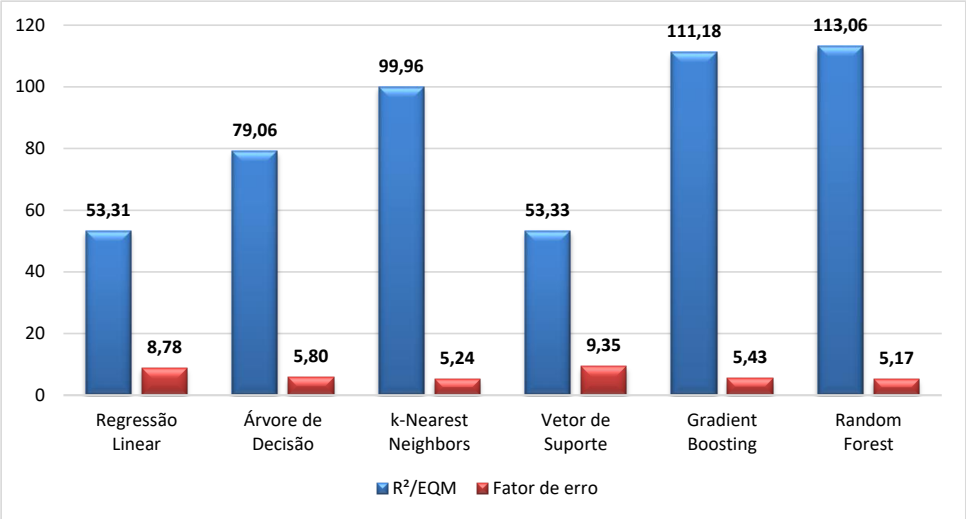
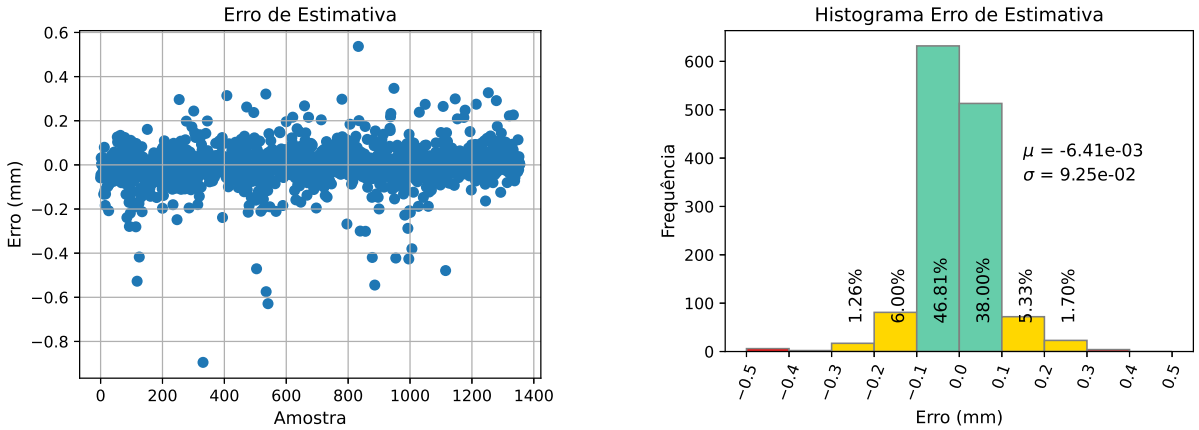


Figura 50: Resultados obtidos, com normalização com seleção de atributos



(a) distribuição do erro de estimativa (b) histograma do erro

Figura 51: Análise do erro com *Random Forest* com seleção de atributos

Tabela 9: Desempenho dos regressores baseados em comitês de CARTs

Regressor	**Experimento**	**Métricas**			
		EQM ($\times 10^{-3}$)	**R^2** ($\times 10^{-1}$)	**EA_{max}** ($\times 10^{-1} mm$)	**$\overline{EA}$** ($\times 10^{-2} mm$)
Gradient Boosting	sem normalização, sem seleção	8,93	9,70	9,16	5,87
Random Forest	sem normalização, sem seleção	8,77	9,71	8,80	5,83
Gradient Boosting	sem normalização, com seleção	9,68	9,68	8,79	5,96
Random Forest	sem normalização, com seleção	8,77	9,71	8,92	5,83
Gradient Boosting	com normalização, sem seleção	8,76	9,71	9,13	5,93
Random Forest	com normalização, sem seleção	8,52	9,71	8,80	5,75

5.2 **Classificação da Gravidade dos Alvéolos**

Também adotou-se a estratégia de verificar os impactos causados pela normalização Z-*score* e pela seleção de atributos no desempenho dos classificadores tradicionais. Como métricas de desempenho foram adotadas a *acurácia*, que está relacionada com o número de acertos de classificação em todo o conjunto de teste e a *recall* que fornece informações de classificação por classe, definidas nas Equações 34 e 35. Além destas, também foi empregado o que denominou-se como *Erro Crítico*.

$$Acurácia = \frac{VP}{N} \times 100 \tag{34}$$

$$Recall\ Bi = \left.\frac{VP_{Bi}}{n_{Bi}}\right|_{i=2}^{5} \times 100 \tag{35}$$

Onde:

VP – verdadeiros positivos (acertos de classificação total);

N – número de amostras do conjunto de teste;

VP_{Bi} – verdadeiros positivos da classe Bi;

n_{Bi} – número de amostras da classe Bi no conjunto de teste.

Ilustrativamente para a definição do Erro Crítico, na Figura 52 é mostrada uma matriz de confusão genérica relacionando as classes reais com as preditas por um classificador. Perceba que os acertos (Verdadeiros Positivos) estão na diagonal principal, portanto os valores nas demais posições representam erros de classificação. Por exemplo, quando um algoritmo gera uma previsão da classe B2 porém a classe real é B3, do ponto de vista da segurança e operação da estrutura onde o alvéolo está localizado, é um erro mais grave que se a predição fosse da classe B3 e a classe real B2. Assim, o Erro Crítico fica definido como a soma dos erros de classificação abaixo da diagonal da matriz de confusão gerada pelo classificador em questão; para a configuração mostrada na Figura 52 o número de Erros Críticos observados é dado por $29 + 1 + 31 + 39 = 100$.

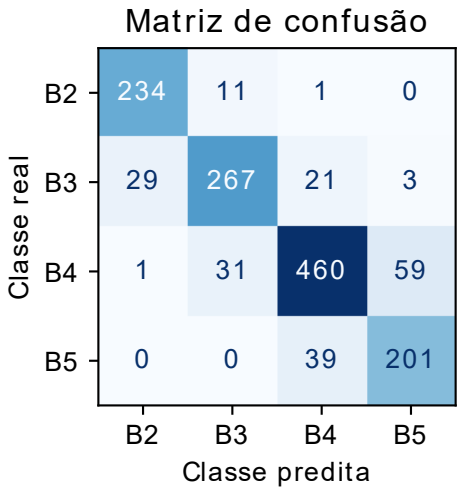


Figura 52: Ilustração para definição do Erro Crítico

5.2.1 Experimentos sem Normalização

Primeiramente, foram avaliados os desempenhos dos classificadores apresentando todos os atributos e sem normalização. Os resultados obtidos estão mostrados na Tabela 10, na Figura 53 e na Figura 54, onde se observa que os modelos baseados em árvores de decisão foram os que apresentaram as maiores acurácias. Individualmente, a classe B2 apresentou os maiores valores de recall em todos os classificadores analisados. O *Gradient Boosting*, proporcionou as mais altas recall média e acurácia, com o menor número de ocorrências de erros críticos.

Tabela 10: Desempenho dos classificadores sem normalização sem seleção de atributos

Classificador	**Hiperpa-râmetros**	**Recall (%)**					**Acurácia (%)**	**Erro Crítico**
		Média	**B2**	**B3**	**B4**	**B5**		
Regressão Logística	$n_iter = 1600$; $tol = 10^{-5}$	86,62	94,31	79,69	87,48	85,00	86,44	102
Árvore de Decisão	$p = 10$; $critério =$ Gini	90,64	95,12	84,69	89,84	92,92	90,13	72
k-*Nearest Neighbors*	$k = 5$	85,39	93,90	83,44	84,21	80,00	85,04	93
Vetor de Suporte	$C = 3{,}0$; $kernel =$ RBF	86,52	95,12	92,50	82,21	86,25	85,34	99
Gradient Boosting	$\boldsymbol{\eta = 0{,}1}$; $\boldsymbol{n = 300}$	**91,81**	**94,31**	**87,19**	**93,65**	**92,08**	**91,97**	**61**
Random Forest	$p = 10$; $n = 600$	91,23	94,72	86,25	93,10	90,83	91,38	65

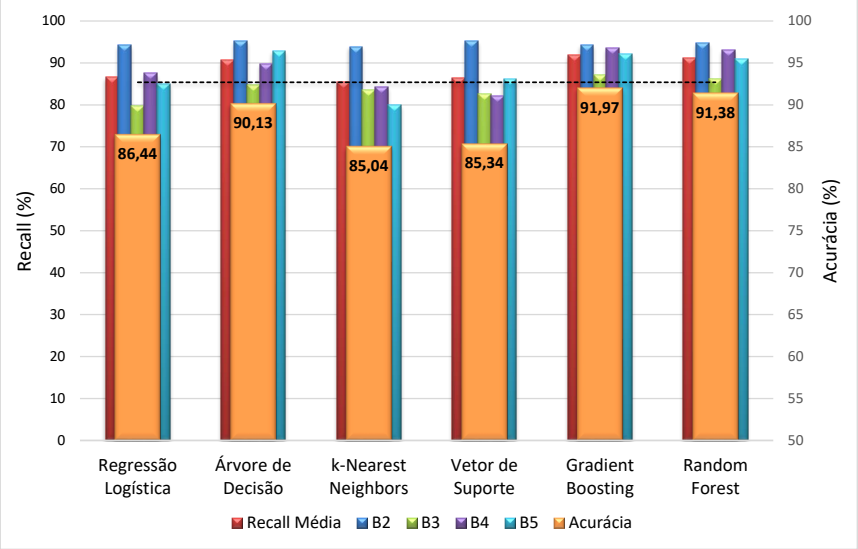


Figura 53: Recall e acurácia, sem normalização sem seleção de atributos

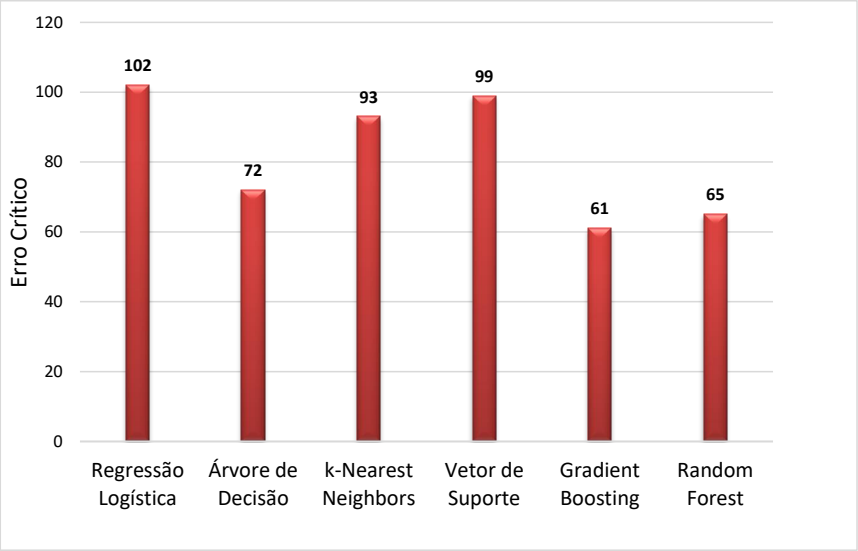


Figura 54: Erro crítico, sem normalização sem seleção de atributos

Na sequência, realizando a exclusão dos atributos mais linearmente correlacionados do conjunto de dados (também com limiar de 0,92), os classificadores foram submetidos à nova análise. Os resultados encontrados são apresentados na Tabela 11, na Figura 55 e na Figura 56. As diferenças mais significativas observadas em relação ao estudo sem seleção de atributos foram o aumento da recall na classe B3 da Árvore de Decisão (de 84,69% para 90,00%), contribuindo para o aumento na acurácia de 90,13% para 91,30%. Considerando o número de ocorrências de erros críticos, houve redução de 72 para 64, inclusive superando o desempenho do *Random Forest*.

Tabela 11: Desempenho dos classificadores sem normalização com seleção de atributos

Classificador	Hiperparâmetros	Recall (%)					Acurácia (%)	Erro Crítico
		Média	B2	B3	B4	B5		
Regressão Logística	$n_iter = 1600$; $tol = 10^{-5}$	85,60	94,72	78,44	86,75	82,50	85,48	107
Árvore de Decisão	$p = 10$; $critério =$ Gini	91,47	92,68	90,00	91,11	92,08	91,30	64
k-*Nearest Neighbors*	$k = 5$	85,24	93,09	83,75	84,94	79,17	85,11	98
Vetor de Suporte	$C = 3{,}0$; $kernel =$ RBF	86,45	95,12	83,44	83,48	83,75	85,63	100
Gradient Boosting	$\boldsymbol{\eta = 0{,}05}$; $\boldsymbol{n = 1000}$	**91,58**	**93,90**	**87,19**	**92,74**	**92,50**	**91,60**	**60**
Random Forest	$p = 10$; $n = 100$	90,97	94,72	85,63	93,10	90,42	91,16	67

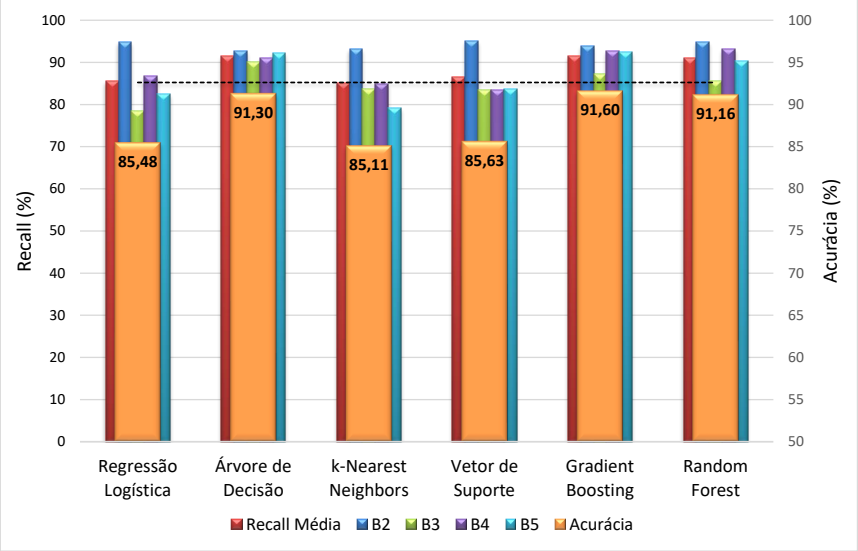


Figura 55: Recall e acurácia, sem normalização com seleção de atributos

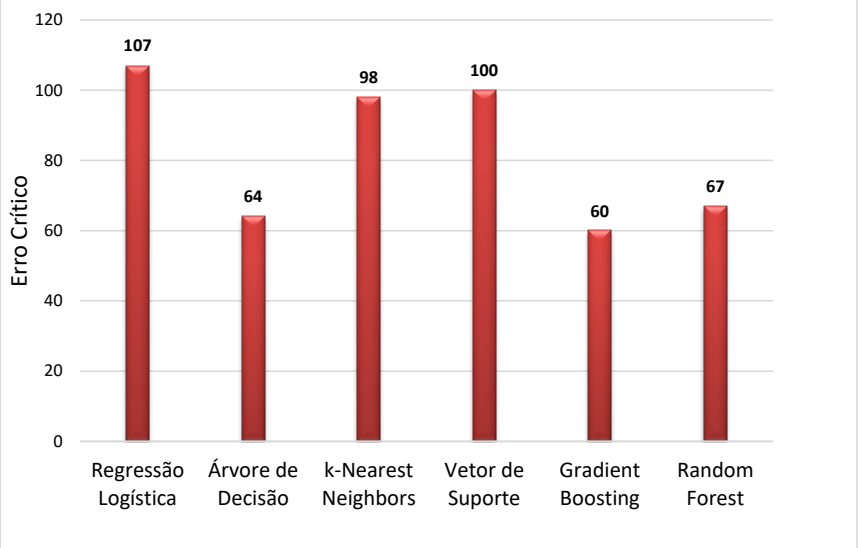


Figura 56: Erro crítico, sem normalização com seleção de atributos

5.2.2 Experimentos com Normalização

Os efeitos da normalização que torna cada atributo com média nula e desvio padrão unitário foram de maior intensidade nos classificadores Regressão Logística, k-*Nearest Neighbors* e Vetor de Suporte, principalmente os dois últimos que tiveram a acurácia incrementada de 86,44% para 88,43% e de 85,04% para 89,98% respectivamente. Quanto ao erro crítico, observa-se queda em todos os algoritmos, com destaque novamente para Regressão Logística, k-*Nearest Neighbors* e Vetor de Suporte que registrou a maior diferença em relação ao caso sem normalização, com 41 ocorrências a menos. Os resultados completos estão mostrados na Tabela 12, na Figura 57 e na Figura 58.

Tabela 12: Desempenho dos classificadores com normalização sem seleção de atributos

Classificador	**Hiperparâmetros**	**Recall (%)**					**Acurácia (%)**	**Erro Crítico**
		Média	**B2**	**B3**	**B4**	**B5**		
Regressão Logística	$n_iter = 400$; $tol = 10^{-5}$	88,55	94,72	83,13	89,29	87,08	88,43	79
Árvore de Decisão	$p = 10$; $critério =$ Gini	90,46	92,28	85,31	89,66	94,58	89,98	62
k-*Nearest Neighbors*	$k = 3$	90,38	94,72	82,50	90,56	93,75	89,98	60
Vetor de Suporte	$C = 3{,}0$; $kernel =$ RBF	91,25	94,72	84,06	91,65	94,58	90,94	58
Gradient Boosting	$\eta = 0{,}3$; $n = 200$	91,37	94,31	85,94	92,74	92,50	91,38	53
Random Forest	$\mathbf{p = 10}$; $\mathbf{n = 1000}$	**91,40**	**94,31**	**85,31**	**93,47**	**92,50**	**91,53**	**54**

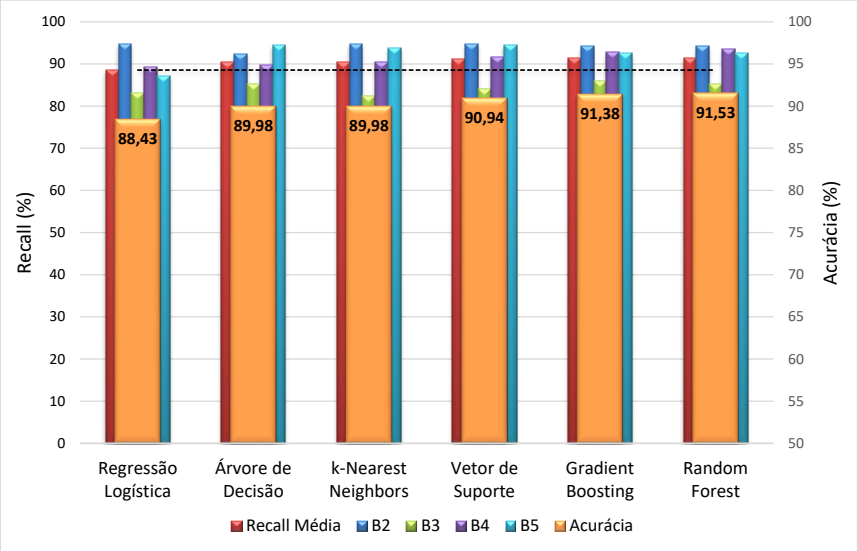


Figura 57: Recall e acurácia, com normalização sem seleção de atributos

Note que o desempenho do classificador Vetor de Suporte supera o da Árvore de Decisão, levando em conta o erro crítico, acurácia e recall média. Já as maiores acurácia

e recall média foram obtidos com *Random Forest*.

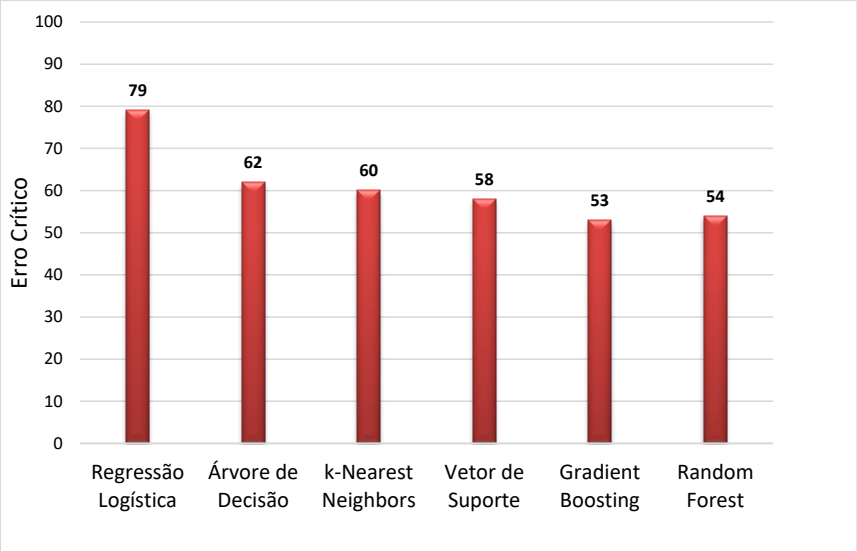


Figura 58: Erro crítico, com normalização sem seleção de atributos

Para finalizar este conjunto experimentos, foi aplicada a seleção de atributos juntamente com a normalização Z-*score*, os resultados são mostrados na Tabela 13, Figura 59 e na Figura 60. Neste caso, o *Gradient Boosting* produziu os melhores resultados de recall média e acurácia. Também observa-se o menor erro crítico dentre os ensaios realizados. Seguindo a tendência dos demais testes estudados, a classe B3 apresentou índices de acerto mais baixos que as demais.

Tabela 13: Desempenho dos classificadores com normalização com seleção de atributos

Classificador	**Hiperparâmetros**	**Recall (%)**					**Acurácia (%)**	**Erro Crítico**
		Média	**B2**	**B3**	**B4**	**B5**		
Regressão Logística	$n_iter = 200$; $tol = 10^{-5}$	88,66	94,72	82,50	89,11	88,33	88,43	77
Árvore de Decisão	$p = 10$; *critério* = Gini	90,49	92,28	84,69	91,65	93,33	90,42	61
k-*Nearest Neighbors*	$k = 3$	89,88	93,90	81,25	90,20	94,17	89,46	62
Vetor de Suporte	$C = 3{,}0$; *kernel* = RBF	91,12	94,72	83,75	91,83	94,17	90,86	59
Gradient Boosting	$\boldsymbol{\eta = 0{,}05}$; $\boldsymbol{n = 1000}$	**91,61**	**93,90**	**85,94**	**93,28**	**93,33**	**91,67**	**52**
Random Forest	$p = 10$; $n = 800$	91,49	94,31	85,63	93,10	92,92	91,53	53

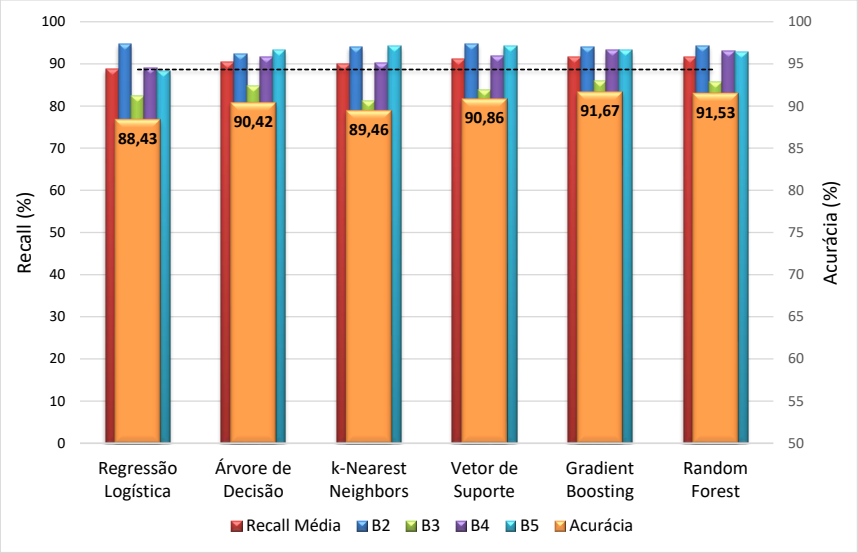


Figura 59: Recall e acurácia, com normalização com seleção de atributos

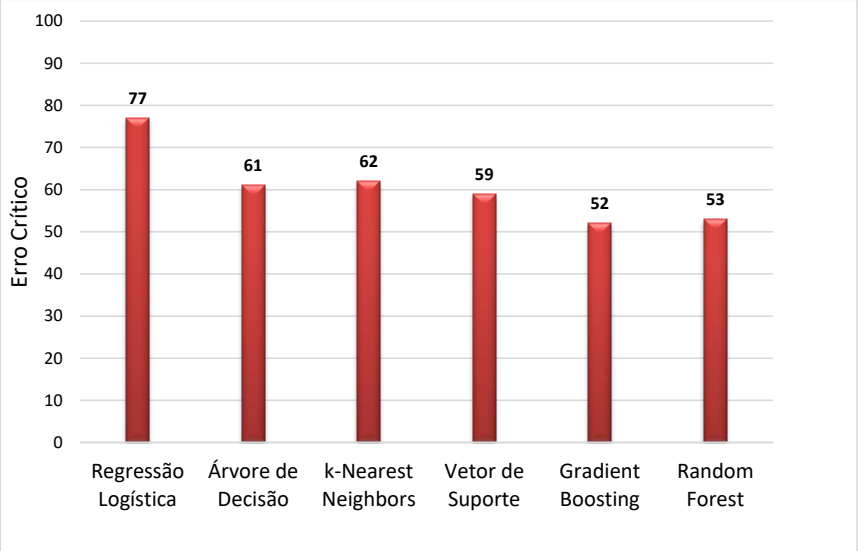


Figura 60: Erro crítico, com normalização com seleção de atributos

5.2.3 Experimentos com Redes Neurais Convolucionais

Assim como no caso dos classificadores tradicionais, foi realizada a busca de hiperparâmetros, para tal 10% do conjunto de treinamento foi dedicado para a validação. Devido às características estocásticas das redes neurais convolucionais, cada combinação de hiperparâmetros foi testada 20 vezes. Assim, a medida de desempenho nesta etapa é dada pela acurácia média obtida para cada combinação no conjunto de validação. Por conta do custo computacional envolvido, a busca de hiperparâmetros concentrou-se na taxa de *dropout* e no número de neurônios das camadas totalmente conectadas; para evi-

tar *overfitting* foi utilizada a estratégia da parada antecipada. A Tabela 14 e Figura 61 mostram as acurácias médias obtidas em cada arquitetura.

Tabela 14: Modelos com maior acurácia média na validação

Arquitetura	**Acurácia Média (%)**	**Taxa de *Dropout***	**N° de Neurônios**
Conv 1	93,04±0,397	0,5	96
Conv 2	93,68±0,685	0,5	96
Conv 3	94,51±0,498	0,5	64

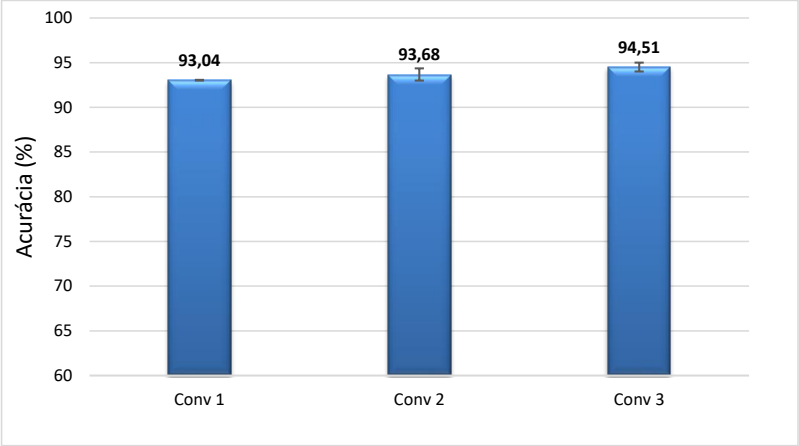


Figura 61: Acurácias médias na validação

Concentrando a atenção na arquitetura Conv 3, que apresentou maior acurácia média na fase de ajuste de hiperparâmetros, agora seu desempenho foi avaliado no conjunto de teste (referente aos mesmos exemplos apresentados aos classificadores tradicionais). Na Tabela 15 estão expressos os resultados obtidos neste estágio.

Tabela 15: Avaliação no teste para a rede Conv 3

Arq.	**Recall (%)**					**Acurácia (%)**	**Erro Crítico**
	Média	**B2**	**B3**	**B4**	**B5**		
Conv 3	94,20	97,36±0,662	91,30±1,312	95,03±1,035	94,10±1,104	94,41±0,322	40,6±7,220

Na sequência, dentre os 20 modelos obtidos para Conv 3, o modelo com maior desempenho individual no conjunto de teste foi selecionado. Estes resultados são apresentados na Tabela 16.

Tabela 16: Modelo com maior desempenho individual

Modelo	**Recall (%)**					**Acurácia (%)**	**Erro Crítico**
	Média	**B2**	**B3**	**B4**	**B5**		
Conv 3	94,96	96,75	91,56	95,28	96,25	94,84	32

Dentre os experimentos realizados com os classificadores tradicionais, no geral, o caso de maior desempenho considerando a quantidade de erros críticos foi no cenário com normalização e com seleção de atributos, onde somando os erros críticos de todos os algoritmos atinge-se o valor de 364 ocorrências. Nas outras possibilidades observadas foram registrados 492 erros críticos (sem normalização sem seleção), 496 (sem normalização com seleção) e 366 com normalização sem seleção. Assim, na Figura 62 e na Figura 63 são comparados os resultados obtidos no contexto com normalização e seleção de atributos com os proporcionados pelo melhor modelo convolucional.

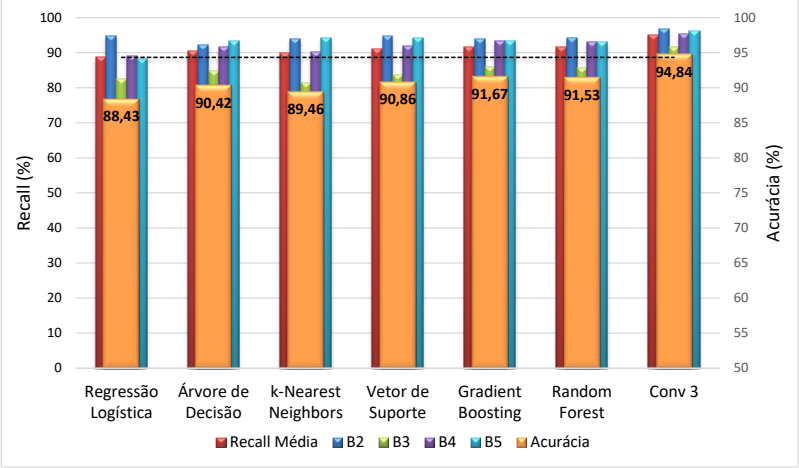


Figura 62: Recall e acurácia para o experimento com normalização e seleção de atributos e comparação com a melhor rede Conv 3

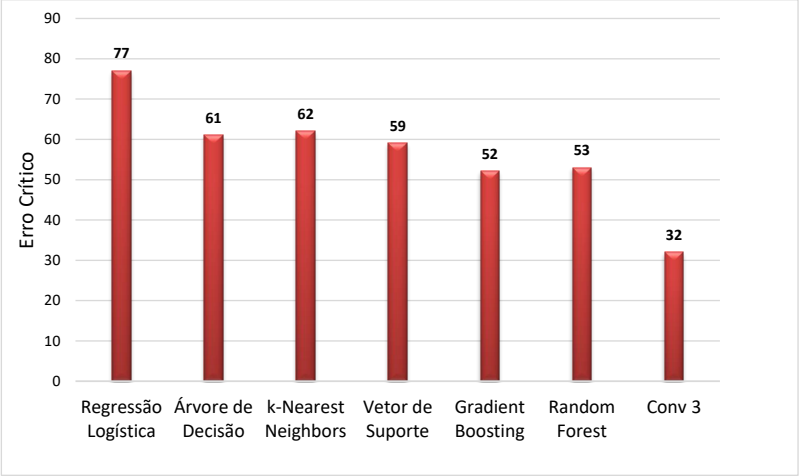


Figura 63: Erro crítico para o experimento com normalização e seleção de atributos e comparação com a melhor rede Conv 3

Na Figura 64 e Figura 65 são feitas comparações com os classificadores tradicionais com maior acurácia obtidos. Observa-se o desempenho superior proporcionado pela rede Conv 3 frente aos classificadores tradicionais.

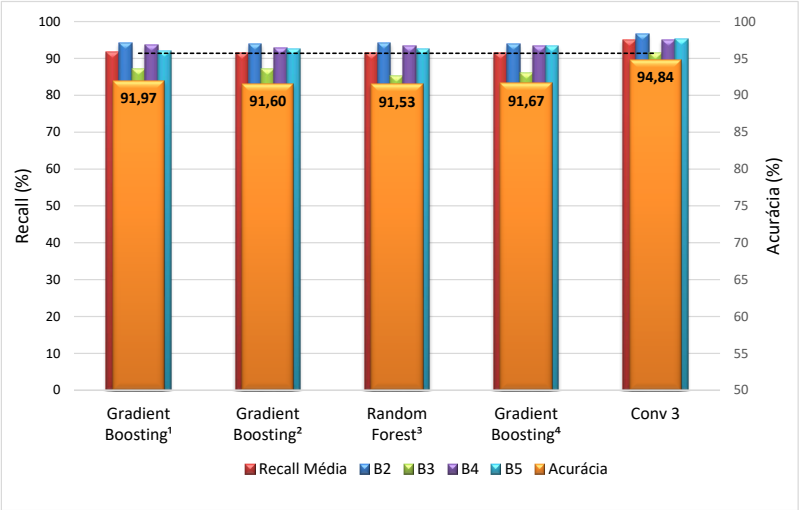


Figura 64: Recall e erro crítico, algoritmos tradicionais *versus* Conv 3
Nota: [1] sem normalização sem seleção
[2] sem normalização com seleção
[3] com normalização sem seleção
[4] com normalização com seleção

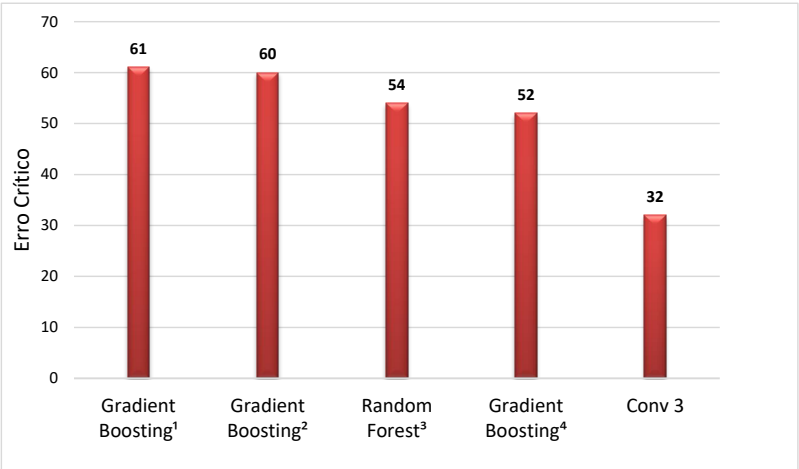


Figura 65: Erro crítico, algoritmos tradicionais *versus* Conv 3
Nota: [1] sem normalização sem seleção
[2] sem normalização com seleção
[3] com normalização sem seleção
[4] com normalização com seleção

CONCLUSÃO

Diante dos resultados alcançados, este trabalho mostra que é possível utilizar as técnicas de *Potential Drop* e modelagem em Elementos Finitos para obtenção de imagens elétricas, que refletem a distribuição do gradiente de potencial elétrico. Em conjunto com estratégias de processamento de imagens, há a geração de exemplos que podem ser aplicados a algoritmos de aprendizado de máquina nas tarefas de estimativa da profundidade e de classificação da severidade de danos causados por corrosão interna em placas de aço inox AISI-304.

As diferentes abordagens permitem observar a sensibilidade dos regressores e classificadores à aplicação de normalização e seleção de atributos. Na primeira comparação, considerando a estimativa da profundidade dos alvéolos com e sem normalização, observa-se que a normalização impactou fortemente no desempenho do k-*Nearest Neighbors*, que apresentou redução aproximada de 26% no erro absoluto médio e de 40% no erro quadrático médio; entretanto não o suficiente para superar os comitês baseados em árvore de decisão (*Gradient Boosting* e *Random Forest*).

Quanto à avaliação da influência da seleção de atributos, vale destacar que no caso sem normalização, o *Gradient Boosting* que apesar de apresentar redução de apenas 3,47% no erro quadrático médio, alcançou significativa redução no número de estimadores (de 250 para 150), reduzindo a complexidade do modelo.

Apesar de a Árvore de Decisão ter se apresentado como um regressor de desempenho regular, com o menor valor de erro máximo (0,85 mm) e erro absoluto médio em patamar mediano, elevando o fator de erro, ao aplicá-la em modelos do tipo comitê o desempenho final é aumentado significativamente. Tomando como exemplo o estudo com normalização e sem seleção de atributos, onde o *Random Forest* proporcionou a mais alta razão R^2/EQM (113,95) e o mais baixo fator de erro (5,06), o erro absoluto médio experimentado pela *Random Forest* (0,0575 mm) é 15,94% menor que a Árvore de Decisão separadamente proporciona (0,0684 mm).

Outro fato observado no efeito da normalização sobre a estimativa das profundidades dos danos diz respeito à distribuição do erro de estimativa. Para os melhores modelos nos casos sem normalização a média deste erro tende a ser positiva, enquanto com a aplicação na normalização a tendência é negativa. Isto indica que na primeira situação

ocorre com maior frequência a subestimação das profundidades, ao passo que na outra há superestimação.

Na classificação da gravidade dos alvéolos, a normalização sem seleção de atributos trouxe maiores benefícios ao k-*Nearest Neighbors* e Vetor de Suporte que tiveram a recall média incrementada de 85,39% e 86,52% para 90,38% e 91,25% respectivamente. Outro efeito observado com o uso da normalização é a redução do erro crítico em todos os classificadores analisados, mesmo naqueles que não apresentaram significativa diferença de acurácia; o *Gradient Boosting* por exemplo teve queda de acurácia de 91,97% para 91,38%, porém melhor resultado na ocorrência de erros críticos, 8 a menos que no caso sem normalização. Os algoritmos *Gradient Boosting* e *Random Forest* dominaram esta etapa do estudo, a menor frequência de erro crítico (52) foi obtida com *Gradient Boosting* no experimento com normalização e com seleção de atributos, já a mais alta acurácia (91,97%) e recall média (91,81%) foram encontradas no experimento sem normalização e sem seleção de atributos, também com o *Gradient Boosting*.

A seleção de atributos favoreceu a classificação com Árvore de Decisão, que no caso sem normalização teve acurácia aumentada de 90,13% para 91,30% e alcançou a maior recall dentre os classificadores tradicionais na classe B3, atingindo 90,00%. Ao longo dos experimentos esta classe apresentou os menores índices acerto. Pela natureza básica das árvores de decisão, a normalização exerce pouca influência sobre seus resultados, conforme pôde ser constatado nos casos de regressão e classificação.

Entretanto, a abordagem com redes neurais convolucionais proporcionou importante aumento de desempenho em todos os aspectos considerados. O modelo baseado na arquitetura Conv3 produziu acurácia de 94,84%, recall média de 94,96% e a ocorrência de erros críticos foi de 32. Estes resultados indicam o incremento de 3,03% na acurácia, 3,32% na recall média e redução de 38,46% no erro crítico em relação aos melhores desempenhos alcançados com os classificadores tradicionais. Inclusive na classe B3 os resultados obtidos são mais altos, alcançando recall de 91,56%.

Como continuidade deste trabalho, a aplicabilidade da rede convolucional U-Net (RONNEBERGER; FISCHER; BROX, 2015) pode ser investigada para melhorar a etapa de segmentação. Essa rede poderia ser treinada para segmentar diretamente a área danificada em vez das regiões de influência. É especialmente importante porque pode melhorar a detecção da classe B2, visando encontrar corrosão interna em seus estágios mais iniciais.

Assim como o uso de redes neurais convolucionais possibilitou o aumento da capacidade de classificação, espera-se que a aplicação de redes neurais para regressão possam trazer benefícios para a estimativa da profundidade dos alvéolos.

No sentido físico/geométrico do problema de corrosão interna, uma direção promissora é investigar a possibilidade de detectar a mudança na densidade de corrente medindo o campo magnético. Isso permitiria empregar este método quando não houver acesso ao exterior da peça metálica a ser testada. Além disso, uma transição natural é a adoção de geometrias cilíndricas, mais próximas da realidade de tubulações e dutos.

REFERÊNCIAS

ABADI, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Disponível em: ⟨https://www.tensorflow.org/⟩.

AL-SHAMARI, A. R. et al. Pipeline integrity management through internal corrosion monitoring. *NACE Corrosion*, San Antonio, Texas, 2014.

AMARAL, J. L. M. D. et al. Análise não destrutiva de estruturas de aço utilizando a técnica do campo magnético de fuga e redes neurais. *VI SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE*, Bauru, 9 2003.

ANGAYARKANNI, N.; DURAIRAJ, K. Euclidean distance transform (edt) algorithm applied to binary image for finding breast cancer. *Biomedical and Pharmacology Journal*, v. 8, p. 407–411, 06 2015. DOI 10.13005/bpj/628.

ASTM. *G46-94*: Standard guide for examination and evaluation of pitting corrosion, standard. [S.l.], 2018. G46-94 (Reapproved 2018).

ATHA, D.; JAHANSHAHI, M. Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Structural Health Monitoring: An International Journal*, p. 147592171773705, 11 2017. DOI 10.1177/1475921717737051.

AWAD, M.; KHANNA, R. Support vector regression. In: _____. *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Berkeley, CA: Apress, 2015. p. 67–80. ISBN 978-1-4302-5990-9. Disponível em: ⟨https://doi.org/10.1007/978-1-4302-5990-9_4⟩. Acesso em: 27 set. 2020.

BRADSKI, G. The opencv library. *Dr. Dobb's Journal of Software Tools*, p. 122–125, 2000.

BREIMAN, L. Random forests. *Machine Learning*, Kluwer Academic Publishers, USA, v. 45, n. 1, 2001. ISSN 0885-6125. Disponível em: ⟨https://link.springer.com/article/10.1023/A:1010933404324⟩. Acesso em: 04 out. 2020.

BUDUMA, N. *Fundamentals of Deep Learning : Designing Next-Generation Machine Intelligence Algorithms*. 1. ed. [S.l.]: Sebastopol: O'Reilly, 2017. ISBN 978-1491925584.

CHATTERJEE, S. *Different Kinds of Convolutional Filters*. 2017. Disponível em: ⟨https://www.saama.com/blog/different-kinds-convolutional-filters/⟩. Acesso em: 27 jun. 2020.

CHOLLET, F. et al. *Keras*. 2015. ⟨https://keras.io⟩.

COMSOL AC/DC Module User's Guide. Stockholm, Sweden: [s.n.], 2018. COMSOL Multiphysics, COMSOL AB. Disponível em: ⟨www.comsol.com⟩.

CORCORAN, J. et al. A quasi-dc potential drop measurement system for material testing. *IEEE Transactions on Instrumentation and Measurement*, v. 69, n. 4, p. 1313–1326, 2020. DOI 10.1109/TIM.2019.2908509. Disponível em: ⟨https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8680671⟩. Acesso em: 30 jun. 2020.

DEMIROVIĆ, D. An Implementation of the Mean Shift Algorithm. *Image Processing On Line*, v. 9, p. 251–268, 2019. ⟨https://doi.org/10.5201/ipol.2019.255⟩.

DOREMUS, L. et al. Calibration of the potential drop method for monitoring small crack growth from surface anomalies - crack front marking technique and finite element simulations. *International Journal of Fatigue*, v. 70, p. 178 – 185, 2015. ISSN 0142-1123. Disponível em: ⟨http://www.sciencedirect.com/science/article/pii/S014211231400231X⟩.

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification (2nd Edition)*. USA: Wiley-Interscience, 2000. ISBN 0471056693.

FRIEDMAN, J. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, v. 29, 11 2000.

GAN, F. et al. Investigation of pitting corrosion monitoring using field signature method. *Journal of the International Measurement Confederation*, Elsevier B.V., v. 82, p. 46–54, 3 2016. Disponível em: ⟨https://doi.org/10.1016/j.measurement.2015.12.040⟩.

GANDOSSI, L. et al. The potential drop method for monitoring crack growth in real components subjected to combined fatigue and creep conditions: Application of fe techniques for deriving calibration curves. *International Journal of Pressure Vessels and Piping*, v. 78, p. 881–891, 11 2001. DOI 10.1016/S0308-0161(01)00103-X.

GENTIL, V. *Corrosão*. 4. ed. [S.l.]: LTC - Livros Técnicos e Científicos Editora, 2003.

GIRSHICK, R. et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2013. Disponível em: ⟨http://arxiv.org/abs/1311.2524v5⟩. Acesso em: 24 jul. 2020.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning*. 2. ed. [S.l.]: Springer-Verlag New York, 2009. ISBN 978-0-387-84857-0.

HAYT, W. H. J.; BUCK, J. A. *Eletromagnetismo*. 8. ed. Porto Alegre: AMGH, 2013. ISBN 978-8580551549.

HE, K. et al. *Deep Residual Learning for Image Recognition*. 2015. Disponível em: ⟨https://arxiv.org/abs/1512.03385v1⟩. Acesso em: 24 jul. 2020.

HEGAZY, A. A. *Thermal Joint Conductance of Conforming Rough Surfaces: Effect of Surface Microhardness Variation*. Tese (Doutorado) — University of Waterloo, 1985.

HOANG, N.-D.; DUC, T. Image processing based detection of pipe corrosion using texture analysis and metaheuristic-optimized machine learning approach. *Computational Intelligence and Neuroscience*, In Press, p. 13, 06 2019. DOI 10.1155/2019/8097213.

IOFFE, S.; SZEGEDY, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. Disponível em: ⟨https://arxiv.org/pdf/1502.03167.pdf⟩. Acesso em: 11 set. 2020.

JAFARZADEH, S.; CHEN, Z.; BOBARU, F. Computational modeling of pitting corrosion. *Corrosion Reviews*, De Gruyter, Berlin, Boston, v. 37, n. 5, p. 419 – 439, 2019. DOI: https://doi.org/10.1515/corrrev-2019-0049. Disponível em: ⟨https://www.degruyter.com/view/journals/corrrev/37/5/article-p419.xml⟩. Acesso em: 25 ago. 2020.

JAMES, G. et al. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014. ISBN 1461471370. Disponível em: ⟨http://faculty.marshall.usc.edu/gareth-james/ISL/ISLR%20Seventh%20Printing.pdf⟩. Acesso em: 23 jul. 2020.

KETKAR, N. *Deep Learning with Python A Hands-on Introduction*. 1. ed. [S.l.]: APRESS, 2017. ISBN 978-1484227657.

KHAN, A. et al. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, Springer Science and Business Media LLC, Apr 2020. ISSN 1573-7462. DOI 10.1007/s10462-020-09825-6. Disponível em: ⟨https://arxiv.org/ftp/arxiv/papers/1901/1901.06032.pdf⟩. Acesso em: 24 jul. 2020.

KOCH, G. et al. International measures of prevention, application and economics of corrosion technologies study (impact). *Technical report*, NACE international, Houston, 2016. Disponível em: ⟨http://impact.nace.org/documents/Nace-International-Report.pdf⟩. Acesso em: 28 jun. 2020.

KOTU, V.; DESHPANDE, B. Chapter 4 - classification. In: KOTU, V.; DESHPANDE, B. (Ed.). *Predictive Analytics and Data Mining*. Boston: Morgan Kaufmann, 2015. p. 63 – 163. ISBN 978-0-12-801460-8. Disponível em: ⟨http://www.sciencedirect.com/science/article/pii/B9780128014608000045⟩. Acesso em: 04 out. 2020.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. Curran Associates, Inc., p. 1097–1105, 2012. Disponível em: ⟨http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf⟩.

LANTZ, B. *Machine Learning with R*. 2. ed. Birmingham B3 2PB, UK: Packt Publishing, 2015. ISBN 978-1-78439-390-8.

LeCun, Y. et al. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, v. 1, n. 4, p. 541–551, 1989. DOI 10.1162/neco.1989.1.4.541.

Lecun, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998. DOI 10.1109/5.726791.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. 2014. ArXiv:1411.4038 [cs.CV]. Disponível em: ⟨https://arxiv.org/abs/1411.4038v1⟩. Acesso em: 25 out. 2020.

MAZUROWSKI, M. A. et al. *Deep learning in radiology: an overview of the concepts and a survey of the state of the art*. 2018. Disponível em: ⟨https://arxiv.org/pdf/1802.08717v1.pdf⟩. Acesso em: 24 jul. 2020.

MITCHELL, M.; LINK, R.; LJUSTELL, P. The effect of large scale plastic deformation on fatigue crack length measurement with the potential drop method. *Journal of Testing and Evaluation*, v. 39, p. 985–1002, 01 2011. DOI 10.1520/JTE103532.

NIELSEN, M. A. *Neural Networks and Deep Learning*. Determination Press, 2015. Disponível em: ⟨http://neuralnetworksanddeeplearning.com/⟩. Acesso em: 23 jul. 2020.

NITG390. Techniques for monitoring corrosion and related parameters in field applications. *NACE International,*, Houston, Texas, Publication 3T199, 2013.

NONDESTRUCTIVE Testing Handbook: Electric Current Test Principles. 1. ed. Columbia: American Society for Nondestructive Testing, 1959. v. 9.

Otsu, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 9, n. 1, p. 62–66, 1979. DOI 10.1109/TSMC.1979.4310076.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, v. 12, n. 85, p. 2825–2830, 2011. Disponível em: ⟨http://jmlr.org/papers/v12/pedregosa11a.html⟩.

PINHEIRO, G. et al. Obtenção da imagem de defeitos em materiais por mapeamento elétrico. ANAIS DO 14° SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, Campinas, p. 2681–2686, 01 2019. DOI 10.17648/sbai-2019-111545.

REDDY, J. N. *An Introduction to the Finite Element Method*. 3. ed. New York: McGraw-Hill Education, 2006. ISBN 9781259861901. Disponível em: ⟨https://www.accessengineeringlibrary.com/content/book/9780072466850/toc-chapter/chapter1/section/section1⟩. Acesso em: 01 set. 2020.

RITCHIE, R.; J., B. K. On the calibration of the electrical potential technique for monitoring crack growth using finite element methods. *International Journal of Fracture*, n. 15, p. 47–55, 1979.

ROBERGE, P. R. *Corrosion Inspections and Monitoring*. Kingston/Ontario: John Wiley & Sons, 2007. ISBN 978-0470099766.

ROERDINK, J. B. T. M.; MEIJSTER, A. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundam. Inform.*, v. 41, p. 187–228, 2000.

RONNEBERGER, O.; FISCHER, P.; BROX, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. Disponível em: ⟨https://arxiv.org/abs/1505.04597⟩.

RYU, K. et al. Pipe thinning model development for direct current potential drop data with machine learning approach. *Nuclear Engineering and Technology*, v. 52, n. 4, p. 784 – 790, 2020. ISSN 1738-5733. DOI 10.1016/j.net.2019.10.004. Disponível em: ⟨http://www.sciencedirect.com/science/article/pii/S1738573319301135⟩.

SAHA, S. *A Comprehensive Guide to Convolutional Neural Networks*. 2018. Disponível em: ⟨https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53⟩. Acesso em: 27 jun. 2020.

SCHWEITZER, A. *Fundamentals of corrosion - Mechanisms, Causes and Preventative Methods*. [S.l.]: CRC Press, 2010. ISBN 978-1420067705.

SEGHIER, M. E. A. B. et al. Prediction of maximum pitting corrosion depth in oil and gas pipelines. *Engineering Failure Analysis*, v. 112, p. 104505, 2020. ISSN 1350-6307. DOI: https://doi.org/10.1016/j.engfailanal.2020.104505. Disponível em: ⟨http://www.sciencedirect.com/science/article/pii/S1350630719318746⟩. Acesso em: 25 ago. 2020.

SHI, Y. et al. Theory and application of magnetic flux leakage pipeline detection. *Sensors*, v. 15, p. 31036–31055, 12 2015. DOI 10.3390/s151229845.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, v. 6, n. 60, 2019. DOI https://doi.org/10.1186/s40537-019-0197-0. Disponível em: ⟨https://link.springer.com/content/pdf/10.1186/s40537-019-0197-0.pdf⟩. Acesso em: 10 set. 2020.

SILVA, V. et al. Evaluation of non-intrusive systems for internal corrosion monitoring. *REVISTA IPT TECNOLOGIA E INOVAÇÃO*, v. 1, 01 2017. DOI 10.34033/2526-5830-v1n6-5.

SIMONETTI, F. et al. Ultrasonic computerized tomography for continuous monitoring of corrosion and erosion damage in pipelines. In: . [S.l.: s.n.], 2015.

SIMONYAN, K.; ZISSERMAN, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. 1409.1556 p. ArXiv:1409.1556v6.

SOILLE, P. *Morphological Image Analysis: Principles and Applications*. 2. ed. Berlin, Heidelberg: Springer-Verlag, 2003. ISBN 3540429883.

SPOSITO, G. *Advances in Potential Drop Techniques for Non-Destructive Testing*. PhD Thesis — Imperial College London, Londres, 2009.

SPOSITO, G.; CAWLEY, P.; NAGY, P. Potential drop mapping for the monitoring of corrosion or erosion. *JNDT&E International*, p. 394–402, March 2010. DOI 10.1016/j.ndteint.2010.03.005.

SPOSITO, G. et al. A review of non-destructive techniques for the detection of creep damage in power plant steels. *NDT & E International*, v. 43, p. 555–567, 10 2010. DOI 10.1016/j.ndteint.2010.05.012.

SRIVASTAVA, N. et al. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, v. 15, n. 56, p. 1929–1958, 2014. Disponível em: ⟨http://jmlr.org/papers/v15/srivastava14a.html⟩. Acesso em: 10 set. 2020.

STANFORD. *CS231n Convolutional Neural Networks for Visual Recognition*. 2019. Stanford Vision and Learning Lab - Stanford University. Disponível em: ⟨https://cs231n.github.io/convolutional-networks/⟩. Acesso em: 23 jul. 2020.

SZEGEDY, C. et al. *Going Deeper with Convolutions*. 2014. Disponível em: ⟨https://arxiv.org/abs/1409.4842⟩. Acesso em: 24 jul. 2020.

TAN, L.; JIANG, J. Chapter 13 - image processing basics. In: TAN, L.; JIANG, J. (Ed.). *Digital Signal Processing (Third Edition)*. Third edition. Academic Press, 2019. p. 649 – 726. ISBN 978-0-12-815071-9. Disponível em: ⟨http://www.sciencedirect.com/science/article/pii/B9780128150719000130⟩.

TARNOWSKI, K. M. *Measuring Crack Initiation and Growth in the Presence of Large Strains using the Potential Drop Technique*. Tese (Doutorado) — Imperial College London, Londres, 2016.

UDPA, S. S.; MOORE, P. O. *Nondestructive Testing Handbook:Electromagnetic Testing*. 3. ed. Columbia: American Society for Nondestructive Testing, 2004. v. 5. ISBN 1-57117-046-4.

VALUEVA, M. et al. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*, v. 177, p. 232 – 243, 2020. ISSN 0378-4754.

VIRTANEN, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, v. 17, p. 261–272, 2020.

Vriesman, D. et al. Texture cnn for thermoelectric metal pipe image classification. In: *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. [S.l.: s.n.], 2019. p. 569–574. DOI 10.1109/ICTAI.2019.00085.

WALT, S. van der et al. scikit-image: image processing in python. *PeerJ*, v. 2, p. e453, jun 2014. ISSN 2167-8359. Disponível em: ⟨https://doi.org/10.7717/peerj.453⟩.

YANG, L. *Techniques for Corrosion Monitoring*. Houston, Texas: Woodhead Publishing, 2008. ISBN 978-1845691875.

YOUNG, I. *Image Analysis and Mathematical Morphology*. USA: Academic Press, Inc., 1983. ISBN 0126372403.

YOVANOVICH, M. Four decades of research on thermal contact, gap, and joint resistance in microelectronics. *IEEE TRANSACTIONS ON COMPONENTS AND PACKAGING TECHNOLOGIES*, v. 28, june 2005. DOI 10.1109/TCAPT.2005.848483. Disponível em: ⟨https://ieeexplore.ieee.org/document/1432927⟩.

YOVANOVICH, M.; MAROTTA, E. Thermal spreading and contact resistance. Heat Transfer Handbook, A. Bejan and A.D. Kraus, John Wiley & Sons, 2003.

ZEILER, M. D.; FERGUS, R. *Visualizing and Understanding Convolutional Networks*. 2013. 1311.2901 p. ArXiv:1311.2901v3.

ZHOU, Z.-H. *Ensemble Methods: Foundations and Algorithms*. 1st. ed. [S.l.]: Chapman & Hall/CRC, 2012. ISBN 1439830037.

APÊNDICE A

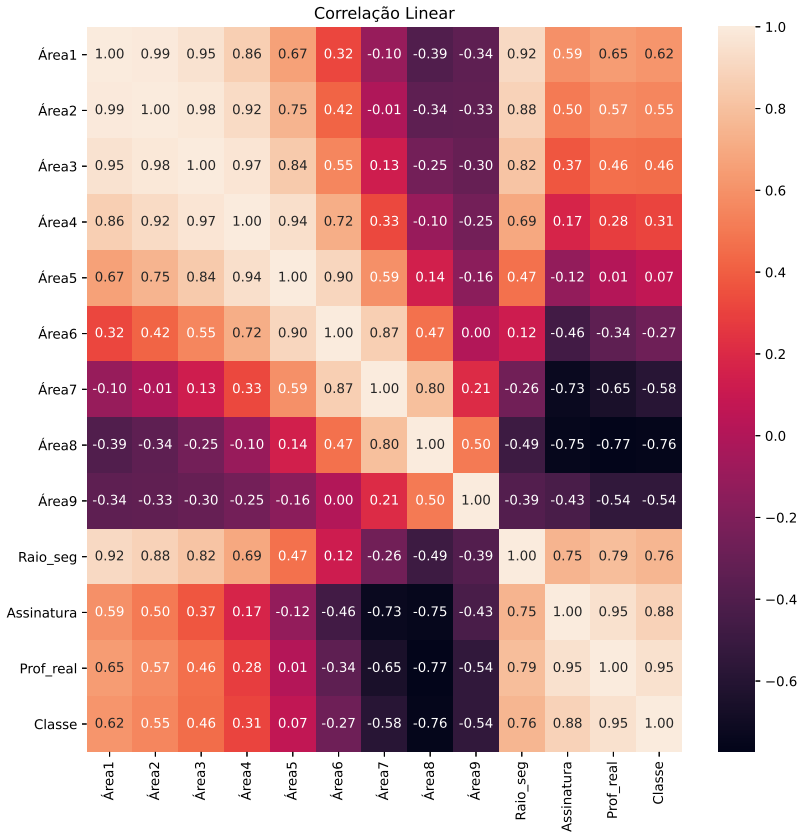


Figura 66: Correlação linear dos atributos extraídos

Tabela 17: Valores de hiperparâmetros utilizados na busca em grade para os regressores

Regressor	**Hiperparâmetros**
Regressão Linear	N/A
Árvore de Decisão	p = [2, 4, 6, 8, 10]
k-*Nearest Neighbors*	k = [1, 3, 5, 7, 9, 11, 13, 15]
Vetor de Suporte	C = [0,1; 0,3; 0,5; 0,7; 0,9; 1,0; 1,3; 1,5; 1,7; 2,0; 2,2; 2,4; 2,6; 2,8; 3,0]; *kernel* = [Linear, RBF]
Gradient Boosting	η = [0,001; 0,01; 0,05; 0,1; 0,3; 0,4]; n = [50, 100, 150, 200, 250, 300, 500, 1000]
Random Forest	p = [2, 4, 6, 8, 10]; n = [80, 100, 200, 400, 600, 800, 1000]

Tabela 18: Valores de hiperparâmetros utilizados na busca em grade para os classificadores

Classificador	**Hiperparâmetros**
Regressão Logística	n_iter = [200, 400, 800, 1600]; tol = [1e-5, 1e-6]
Árvore de Decisão	p = [2, 4, 6, 8, 10]; *critério* = [Gini, Entropia]
k-*Nearest Neighbors*	k = [1, 3, 5, 7, 9, 11, 13, 15]
Vetor de Suporte	C = [0,1; 0,3; 0,5; 0,7; 0,9; 1,0; 1,3; 1,5; 1,7; 2,0; 2,2; 2,4; 2,6; 2,8; 3,0]; *kernel* = [Linear, RBF]
Gradient Boosting	η = [0,001; 0,01; 0,05; 0,1; 0,3; 0,4]; n = [50, 100, 150, 200, 250, 300, 500, 1000]
Random Forest	p = [2, 4, 6, 8, 10]; n = [80, 100, 200, 400, 600, 800, 1000]

Tabela 19: Valores de hiperparâmetros utilizados na busca em grade para as redes convolucionais

Arquitetura	**Hiperparâmetros**
Conv 1	*Nº de Neurônios* = [32, 64, 96]; *batch_size* = 128; *patience* = 30; *dropout* [0,2; 0,4; 0,5]
Conv 2	*Nº de Neurônios* = [32, 64, 96, 128]; *batch_size* = 128; *patience* = 35; *dropout* [0,2; 0,4; 0,5]
Conv 3	*Nº de Neurônios* = 64; *batch_size* = 256; *patience* = 40; *dropout* [0,2; 0,4; 0,5]

batch_size: número de exemplos de treinamento usados em uma iteração;
patience: número de épocas que controla a parada antecipada.