UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

JOSÉ PAULO GONÇALVES DE OLIVEIRA

APRENDIZADO DE MÁQUINA NA INDÚSTRIA 4.0: detecção de anomalias em sistemas embarcados e classificação de substâncias

Recife
2022

JOSÉ PAULO GONÇALVES DE OLIVEIRA

APRENDIZADO DE MÁQUINA NA INDÚSTRIA 4.0: detecção de anomalias em sistemas embarcados e classificação de substâncias

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Doutor em Engenharia Elétrica.

Área de concentração: Processamento de Energia.

Orientador: Prof. Dr. Carmelo José Albanez Bastos Filho.

Coorientador: Prof. Dr. Sergio Campello Oliveira.

Recife

2022

Catalogação na fonte:
Bibliotecária Sandra Maria Neri Santiago, CRB-4 / 1267

O48a Oliveira, José Paulo Gonçalves de.
Aprendizado de máquina na indústria 4.0: detecção de anomalias em sistemas embarcados e classificação de substâncias / José Paulo Gonçalves de Oliveira. – 2022.
161 f.: il., figs., tabs., abrev. e siglas.

Orientador: Prof. Dr. Carmelo José Albanez Bastos Filho.
Coorientador: Prof. Dr. Sergio Campello Oliveira.
Tese (Doutorado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Elétrica. Recife, 2022.
Inclui referências.

1. Engenharia elétrica. 2. Testes automatizados. 3. Indústria 4.0. 4. Detecção de anomalias. 5. Classificação de substâncias. 6. Aprendizado de máquina. 7. Autoencoder. I. Bastos Filho, Carmelo José Albanez (Orientador). II. Oliveira, Sergio Campello (Coorientador). III. Título.

UFPE

621.3 CDD (22. ed.) BCTG/2022-408

JOSÉ PAULO GONÇALVES DE OLIVEIRA

APRENDIZADO DE MÁQUINA NA INDÚSTRIA 4.0: detecção de anomalias em sistemas embarcados e classificação de substâncias

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, como requisito parcial para a obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Processamento de Energia.

Aprovada em: 18/11/2022.

BANCA EXAMINADORA

_____

Prof. Dr. Carmelo José Albanez Bastos Filho (Orientador)
Universidade de Pernambuco

_____

Prof. Dr. Abel Guilhermino da Silva Filho (Examinador Externo)
Universidade Federal de Pernambuco

_____

Prof. Dr. Francisco Madeiro Bernardino Júnior (Examinador Externo)
Universidade de Pernambuco

_____

Prof. Dr. Guilherme de Alencar Barreto (Examinador Externo)
Universidade Federal do Ceará

_____

Prof. José Alfredo Ferreira Costa (Examinador Externo)
Universidade Federal do Rio Grande do Norte

Este trabalho é dedicado àqueles que mudam o mundo por meio da ciência e pela busca do conhecimento. E à minha família.

AGRADECIMENTOS

Muitas pessoas e instituições contribuíram de forma técnica e não técnica para este trabalho com seu apoio, críticas construtivas, sugestões e orientações.

Em primeiro lugar, gostaria de agradecer ao meu amigo e orientador Prof. Carmelo Bastos pela oportunidade de trabalhar ao seu lado em um tema tão interessante, bem como por seus amplos conhecimento e experiência na área.

Gostaria de expressar minha sincera gratidão ao meu amigo e coorientador Prof. Sérgio Campello, que me incentivou a escolher o tema e viabilizou financeiramente todo o projeto, além das inúmeras sugestões a respeito da escrita desta tese.

Sou profundamente grato aos colegas do Ecomp-POLI/UPE pelo apoio durante o período da pesquisa.

Agradeço também a todos os professores que participaram da minha formação ao longo da graduação e da pós-graduação.

A FACEPE também merece agradecimento pelo apoio financeiro ao projeto, que viabilizou a aquisição dos equipamentos necessários às montagens experimentais.

Acima de tudo, agradeço à minha esposa Dani e a meus filhos Bia e Mateus, que agraciam minha vida com seu apoio, compreensão e amor.

And you run, and you run to catch up with the sun but it's sinking
Racing around to come up behind you again
The sun is the same in a relative way but you're older
Shorter of breath and one day closer to death.
(E você corre, e corre para alcançar o sol, mas ele está se pondo
Arrodeando a Terra para aparecer atrás de você novamente
O sol é o mesmo, de uma forma relativa, mas você está mais velho
Com falta de ar e um dia mais perto da morte.) (WATERS, 1973).

RESUMO

O controle de qualidade é um aspecto crítico, especialmente no contexto da indústria 4.0. Além de ser uma necessidade para atender aos pré-requisitos funcionais de um determinado produto, a qualidade está intimamente relacionada à segurança, à proteção, e a questões econômicas. Neste trabalho, abordamos dois aspectos específicos de certificação da qualidade no contexto moderno da indústria: detecção de anomalias em sistema eletrônicos embarcados e classificação de substâncias químicas ou biológicas. As soluções abordadas são baseadas em modelos de aprendizado de máquina. Na indústria 4.0, formas tradicionais de teste baseadas em inspeções manuais tornaram-se desatualizadas e ineficientes. Além disso, existe uma demanda por produtos com alto índice de personalização. Isso exige alto grau de flexibilidade nos processos de concepção, de projeto e de testes. São necessárias soluções eficazes e flexíveis que não utilizem pontos de contato físico com o produto testado. Nosso estudo apresenta soluções de teste automatizados e não invasivos. Em relação aos testes de substâncias, tradicionalmente utilizam-se técnicas de espectroscopia, realizadas com espectrômetros. Apesar de ser uma técnica bastante madura, sua limitação são o custo e a complexidade do equipamento. Nós propomos uma alternativa simples, porém eficiente para realizar testes sem o uso de espectrômetros. Para validação, projetamos e construímos protótipos para realização de experimentos. Anomalias são detectadas pela análise de sinais temporais capturados do circuito de forma não invasiva. Os sinais são convertidos em imagens espectrográficas que são analisadas por um modelo de aprendizado de máquina. Para classificação de substâncias, apresentamos uma Prova de Conceito utilizando um sistema de transmissão e recepção ópticas. O sinal transmitido possui forma de onda especialmente projetada para obtenção de máximo desempenho. O sinal detectado é convertido em imagem espectrográfica que é usada por um modelo de aprendizado de máquina que realiza a classificação. Para vários cenários experimentais de validação, a taxa de acerto de detecção de anomalia e de classificação de substâncias chega a 100%. Adicionalmente, apresentamos uma técnica para aumentar o desempenho por meio da transformação dos dados utilizados para treinamento e validação do modelo. A eficácia da técnica é

comprovada experimentalmente tanto para detecção de anomalias quanto para classificação de substâncias.

Palavras-chave: testes automatizados; indústria 4.0; detecção de anomalias; classificação de substâncias; aprendizado de máquina; autoencoder.

ABSTRACT

Quality control is a critical aspect, especially in the context of Industry 4.0. In addition to being a necessity to meet the functional prerequisites of a given product, quality is closely related to safety, protection, and economic issues. In this work, we address two specific aspects of quality certification in the modern context of industry: detection of anomalies in embedded electronic systems and classification of chemical or biological substances. The addressed solutions are based on machine learning models. In Industry 4.0, traditional forms of testing based on manual inspections have become outdated and inefficient. In addition, there is a demand for products with a high level of customization. This requires a high degree of flexibility in the conception, design and testing processes. Effective and flexible solutions that do not use physical contact points with the tested product are required. Our study presents automated and non-invasive testing solutions. With regard to classification of substances, traditionally spectroscopy techniques are employed, which is performed with spectrographs. Despite being a very mature technique, its limitations are the cost and complexity of the equipment. We propose a simple but efficient alternative to perform tests without the use of spectrographs. For validation, we design and build prototypes to carry out experiments. Anomalies are detected by analyzing temporal signals captured from the circuit in a non-invasive way. The signals are converted into spectrographic images that are analyzed by the machine learning model. For classification of substances, we present a Proof of Concept using an optical transmission and reception system. The transmitted signal has a specially designed waveform for maximum performance. The detected signal is converted into a spectrographic image that is used by the machine learning model that performs the classification. For several experimental validation scenarios, the accuracy reaches 100%. Additionally, we present a technique to increase performance by transforming the data used for model training and validation. The effectiveness of the technique is experimentally demonstrated both for detecting anomalies and classifying substances.

Keywords: automated testing; industry 4.0; anomaly detection; classification of substances; machine learning; autoencoder.

LISTA DE ABREVIATURAS E SIGLAS

ADC	*Analog to digital converter* (conversor analógico-digital)
ATSR	*Arc-tangent square root* (raiz quadrada arco-tangente)
AUC	*Area under the curve* (área sob a curva)
Divergência-KL	Divergência de Kullback-Leibler
DUT	*Device under test* (sistema a ser testado)
Err	Taxa de erro (de classificação ou detecção de anomalias)
Err_{min}	Taxa de erro mínima
$Errr_{rec}$	Erro de reconstrução do autoencoder
FM	*Frequency modulation* (modulação em frequência)
FN	*False negatives* (número de previsões negativas erradas)
f_o	Frequência da portadora da modulação FM
FP	*False positives* (número de previsões de positivas erradas)
FPR	*False positives rate* (taxa de falsos positivos)
f_{Range}	Faixa de frequência do espectrograma
K_p	Constante da modulação FM
LED	*Light emitting diode* (diodo emissor de luz)
MSE	*Mean squared error* (erro médio quadrático)
Nível DC	Valor médio do sinal
NQ	Norma quadrática média entre dois vetores
PCI	Placa de circuito impresso
Q	Parâmetro de desempenho
RAM	*Random access memory* (memória interna semicondutora)
RGB	Três cores: *red* (vermelho); *Ggreen* (verde); *blue* (azul)
ROC	*Receiver operating characteristic* (características operacionais do receptor)
ROI	Region of interest (região de interesse)
SIFT	*Scale-invariant feature transform* (transformação de característica invariável a escala)
SSD	*Solid state device* (memória externa semicondutora)
TN	*True negatives* (número de previsões negativas corretas)
TNR	*True negative rate* (taxa de verdadeiros negativos ou especificidade)

TP	*True positives* (número de previsões positivas corretas)
TPR	*True positive rate* (taxa de verdadeiros positivos ou sensibilidade)
β	Parâmetro de intensidade não linear da ATSR
ΔT	Duração da janela de aquisição
γ	Parâmetro de ajuste do limiar de decisão
μ	Limiar de decisão
μ_o	Limiar de decisão ótimo (quando Err = Err_{min})
μ_{OK}	Média do erro de reconstrução da classe OK
σ_{OK}	Desvio padrão do erro de reconstrução da classe OK

SUMÁRIO

1 INTRODUÇÃO

O conceito Manufatura Avançada se refere à quarta revolução industrial (Devezas, 2017; Xu, 2018), que é caracterizada pela associação dos sistemas de automação aos sistemas ciber físicos ou CPS (*Cyber-Physical Systems*) (Lu, 2017; Jamaludin, 2018). Essa associação torna o processo produtivo conectado e integrado. A infraestrutura que viabiliza essa nova abordagem é composta por sensores, redes de comunicação e sistemas de armazenamento e análise de dados. Os principais emblemas da também chamada Indústria 4.0 são a personalização da produção, a redução de custos e de energia e a flexibilização da logística.

Diante dessa nova realidade, muitas empresas têm demandado estudos e pesquisas visando ao desenvolvimento de produtos e processos aplicados à manufatura. Tais demandas envolvem testes inteligentes, aferição de qualidade, sensoriamento, conectividade, segurança de dados, análise de dados (*big data*) e produção personalizada. No quesito teste e aferição de qualidade, a indústria busca maneiras rápidas e eficientes para alcançar alto grau de competitividade de seus produtos.

Nesse contexto, a busca por soluções inovadoras que viabilizem a produção de forma eficiente é cada vez importante. Dessa forma, alinhado às necessidades (Indústria) e competências (Universidades), este projeto visa à elaboração de soluções aplicáveis e adequadas à Manufatura Avançada, por meio da implementação de sistemas de aferição da qualidade não invasivos, flexíveis e eficientes.

1.1 O PROBLEMA

Do ponto de vista dos produtores, dado o cenário da quarta revolução industrial, existe uma demanda por processos de teste/qualidade não invasivos, flexíveis, eficazes (precisão, acurácia), e eficientes (baixo custo e baixa complexidade de implantação).

Neste trabalho especificamente, abordamos o problema relacionado à aferição da qualidade de dois casos de aplicação na indústria: sistemas computacionais embarcados (Capítulos 4 e 5); e substâncias químicas ou biológicas (Capítulo 6).

Neste texto, utilizamos o termo DUT (*Device Under Test*) para nos referirmos ao sistema computacional testado e o termo substância pode denotar tanto as químicas quanto as orgânicas.

Ferramentas e métodos de teste de sistemas embarcados convencionais possuem limitações. Em geral são dedicados a um modelo específico e, quando oferecem adaptação, é de forma muito complexa e limitada. Por isso, é muito comum ter que desenvolver um novo sistema de testes para cada modelo distinto de sistema embarcado produzido. Além disso, muitas vezes as soluções são invasivas, ou seja, é preciso realizar conexões físicas dedicadas entre o produto testado e o sistema de teste. No contexto dinâmico da Indústria 4.0, isso se torna inviável.

Os testes de substâncias químicas e/ou biológicas na indústria podem ser executados por meio de análise espectrográfica (Pavia et al., 2014). Nesses casos, o uso de espectrômetros é essencial. A técnica é bastante madura, portanto, sua eficácia e sua eficiência são reconhecidas. A limitação, no caso, são o custo e a complexidade do equipamento.

Como, então, desenvolver soluções capazes de testar o funcionamento correto de sistemas embarcados e de classificar substâncias da forma menos invasiva possível? Ao implementar tal solução, como fazer de forma mais eficiente? E como fazer de forma mais precisa? Em se encontrando respostas, é possível reutilizar tal solução em outras aplicações industriais? Na Seção 1.2, propomos respostas para essas questões e introduzimos os conceitos que as embasam.

1.2 SOLUÇÕES PROPOSTAS

Considerando as questões mencionadas na Seção 1.1 e os dois casos industriais abordados – testes de sistemas computacionais embarcados e classificação de substâncias – este trabalho propõe as respectivas soluções baseadas em um modelo de aprendizado de máquina (autoencoder) que utiliza imagens como dados de entrada. Essas imagens são, por sua vez, construídas a partir de espectrogramas (Cohen, 1989) de sinais temporais. Esses sinais são adquiridos diretamente do sistema embarcado ou por meio de interação da luz com a substância analisada. Nas Seções 1.2.1 a 1.2.3, a seguir, apresentamos um resumo de cada abordagem pesquisada.

Além disso, com o intuito de diferenciar quantitativamente aqueles casos em que a precisão obtida é de 100%, desenvolvemos uma métrica de análise dos resultados baseada não apenas na precisão, como também na robustez oferecida pelo modelo de aprendizado de máquina (Seção 3.7.4).

1.2.1 Detecção de anomalias em sistemas embarcados

O sistema de detecção de anomalias para sistemas embarcados proposto valida seu funcionamento adequado por meio da detecção de anomalias tanto de *firmware* (ou *software*), quanto de *hardware*. Duas abordagens distintas são avaliadas:

- Assinatura elétrica: dados são obtidos pela aquisição de corrente elétrica consumida pelo sistema embarcado testado (Capítulo 4);
- Assinatura termográfica: dados são obtidos pela aquisição termográfica da placa de circuito impresso onde o sistema embarcado é implementado (Capítulo 5).

Em ambos os casos, o sinal adquirido (elétrico ou termográfico) é convertido em espectrograma. Os espectrogramas gerados, por sua vez, são convertidos em imagens coloridas, que são usadas como entrada de um modelo de aprendizado de máquina, responsável pela detecção de anomalias.

1.2.2 Classificação de substâncias

No caso do teste de substâncias, propomos um sistema de classificação que pode ser usado para aferir a qualidade de um produto (indústria química ou alimentícia, por exemplo) ou detectar algum tipo de substância (detecção de patógenos, por exemplo). Apresentamos uma Prova de Conceito (PoC – *Proof of Concept*) de uma técnica alternativa que utiliza aprendizado de máquina para detectar se a amostra testada corresponde a um tipo específico. Também é chamado de classificador de uma classe (Tsai, 2021). Trata-se, portanto, de uma forma alternativa à análise espectrográfica tradicional (Capítulo 6). Em vez de empregar o equipamento espectrômetro, utilizamos um sistema de transmissão e recepção de sinal óptico. O sinal transmitido possui formato (forma de onda temporal) pré-determinado. O sinal detectado é convertido em espectrograma, a partir do qual uma imagem colorida é gerada. Essa imagem é usada por um modelo de aprendizado de máquina que realiza a classificação.

1.2.3 Aperfeiçoamento usando transformação

Além dos sistemas de classificação e teste em si, também propomos melhorias para uso dos modelos de aprendizado de máquina. Tais melhorias são baseadas em pré-processamento dos sinais adquiridos (Capítulo 7). Em um cenário real, a necessidade de melhoria pode decorrer da quantidade escassa ou da qualidade limitada dos dados. O projeto e desenvolvimento de sistemas baseados em aprendizado de máquina dependem fortemente da disponibilidade de dados para treinar o modelo. Em um ambiente real, contudo, nem sempre é possível contar com tal disponibilidade. Seja devido à característica da própria aplicação (sistema a ser testado) ou por restrição de tempo (necessário para coletar dados, treinar o modelo e realizar a implantação). Nesses casos, desenvolver uma solução com restrição de dados oferece limitações, sobretudo relacionadas à acurácia. Por isso, estudamos técnicas que visam contornar a limitação imposta pela falta de dados adequados para treinamento.

As técnicas de pré-processamento de dados investigadas neste trabalho usam as seguintes transformações: modulação de frequência (Faruque, 2017); subtração da tendência linear no domínio do tempo (Detrend) (Moore & Sanadhya, 2009); e ATSR (*Arc-Tangent Square Root*) (Oo, 2010). Selecionamos essas técnicas a partir de um grande número de possibilidades após a realização de testes preliminares.

1.3 ARRANJOS EXPERIMENTAIS

Com o intuito de validar nossas propostas, realizamos aquisição de dados experimentais. Para a detecção de anomalias, um sistema embarcado foi projetado e construído. A partir desse sistema, adquirimos os sinais de corrente e termográficos. Para o teste de substâncias, montamos um sistema de transmissão e recepção usando LEDs (*Light Emitting Diodes* – Diodos Emissores de Luz) e Fotodetectores para aquisição dos sinais ópticos.

Os resultados experimentais validam as abordagens propostas. Tanto o sistema de detecção de anomalias quanto o sistema de classificação de substâncias obtiveram até 100% de precisão em ao menos uma configuração testada.

Em relação às técnicas de aperfeiçoamento, todas as abordagens (detecção de anomalias em sistemas embarcados por assinatura elétrica ou termográfica; e

classificação de substâncias) apresentam substancial melhoria (Seção 7.2). Os resultados indicam que, mesmo em um cenário onde os dados disponíveis são limitados, é possível alcançar um desempenho aceitável utilizando alguma das técnicas propostas de pré-processamento dos dados.

Com base nos resultados obtidos experimentalmente, confirmamos que as soluções propostas podem ser empregadas em ambientes da indústria 4.0, proporcionando aumento da qualidade e da produtividade. Sobretudo, levando-se em conta o fato de que se trata de uma solução de baixa complexidade computacional (modelo de aprendizado de máquina), flexível (adaptável a novos produtos ou substâncias) e de simples implantação. Pois, inicialmente, é necessário apenas a aquisição de dados para treinamento e validação. Ainda que haja limitação na aquisição de dados para treinamento, as técnicas de aperfeiçoamento propostas podem ser empregadas para atingir o desempenho desejado.

1.4 CONTRIBUIÇÕES DA TESE

Em resumo, as contribuições deste trabalho são:

- Desenvolvimento e validação de abordagem para detecção de anomalias (de *software* e de *hardware*) de forma não invasiva de sistemas embarcados por meio da assinatura elétrica (Capítulo 4) e aprendizado de máquina;
- Desenvolvimento e validação de abordagem para detecção de anomalias (de *software* e de *hardware*) de forma não invasiva de sistemas embarcados por meio da assinatura termográfica (Capítulo 5) e aprendizado de máquina;
- Classificação de substâncias químicas e biológicas (Capítulo 6) usando sinais ópticos modulados em frequência e aprendizado de máquina;
- Método para definição do limiar de decisão na ausência de dados da classe anômala (Seção 4.3.3);
- Técnica de aperfeiçoamento do classificador/detector de anomalias por meio de transformação dos dados (Capítulo 7);
- Análise da relação entre a dissimilaridade estatística dos dados pertencentes a classes distintas e o desempenho do modelo de

aprendizado de máquina na detecção de anomalia ou classificação de substâncias (Capítulo 7);

- Definição de uma métrica mais abrangente que a taxa de erro para avaliação de desempenho do modelo de aprendizado de máquina utilizado para detecção de anomalias ou classificação (Seção 3.7.4).

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo, apresentamos uma revisão das publicações mais relevantes ao nosso trabalho. A revisão seguiu o roteiro pesquisa, seleção e leitura de artigos e projetos relacionados aos temas: demandas e soluções da indústria 4.0; aprendizado de máquina; detecção de anomalias em sistemas embarcados; aplicação de aprendizado de máquina e análise espectrográfica na classificação de substâncias; aperfeiçoamento de modelos de aprendizado de máquina.

O objetivo foi identificar técnicas presentes na literatura e adquirir conhecimento, provendo direcionamento adequado à nossa pesquisa. A grande quantidade de trabalhos relacionados aos temas detecção de anomalias/testes de sistemas embarcados e classificação de substâncias demonstram a maturidade científica, demanda por novas soluções e relevância do assunto.

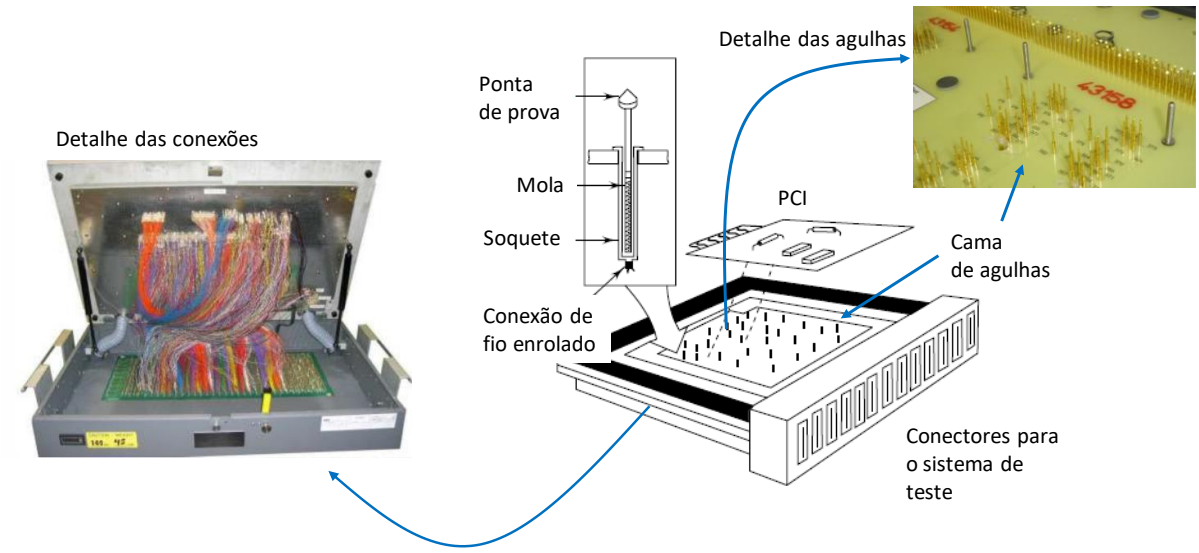
2.1 TESTES DE SISTEMAS EMBARCADOS

Sistemas embarcados são compostos de elementos de *hardware* e de *software*, portanto, os defeitos apresentados podem ser de natureza física ou lógica. Ambos devem ser detectados e, se possível e viável, corrigidos antes da etapa final de produção. O teste independente de *software* e de *hardware* é uma abordagem bastante comum. Entretanto, isso torna o processo mais dispendioso por se tratar de uma solução em duas etapas. Por outro lado, executar testes de *hardware* e *software* de maneira unificada nem sempre é tecnicamente viável, sobretudo quando a complexidade do sistema testado é alta (a complexidade pode ser definida, por exemplo, por meio do número de funcionalidades e interfaces de entrada/saída). Uma abordagem muito utilizada em testes de circuitos eletrônicos é a geração de padrões de sinais que servem como entrada para o DUT (Esser, 2007; Grout, 2006) para emulação de defeitos. Contudo, esses padrões são mais difíceis de se implementar para testar o *software* (Marwedel, 2021). Portanto, a maioria dos trabalhos publicados apresentam soluções para testes independentes de *hardware* e *software*.

Em relação aos testes de *hardware*, os tipos mais comuns de defeitos presentes em sistemas embarcados são montagem de PCIs (Placas de Circuito Impresso), componentes defeituosos e mau funcionamento do produto final (Khandpur, 2006).

Uma infinidade de métodos de teste de *hardware* pode ser empregada dependendo das restrições financeiras e barreiras técnicas. Das técnicas convencionais de testes da PCI, a mais tradicional é baseada em uma cama de agulhas ou *fixture* (Lu, 1997), onde a placa a ser testada repousa, permitindo o contato elétrico. Por meio desses contatos são enviados e recebidos os sinais de entrada e saída do teste. O comportamento do *hardware* e do *software* pode ser verificado de forma unificada nesse procedimento. A principal desvantagem dessa técnica é a falta de flexibilidade do sistema de teste. Há uma dependência física (disposição das agulhas) em relação ao DUT. Qualquer modificação do produto a ser testado demanda um novo projeto para a *fixture*. A Figura 1 ilustra um caso prático desse tipo de sistema, onde é possível observar como a construção é feita de forma artesanal, pois as ligações entre pinos e conectores devem ser feitas manualmente para cada DUT.

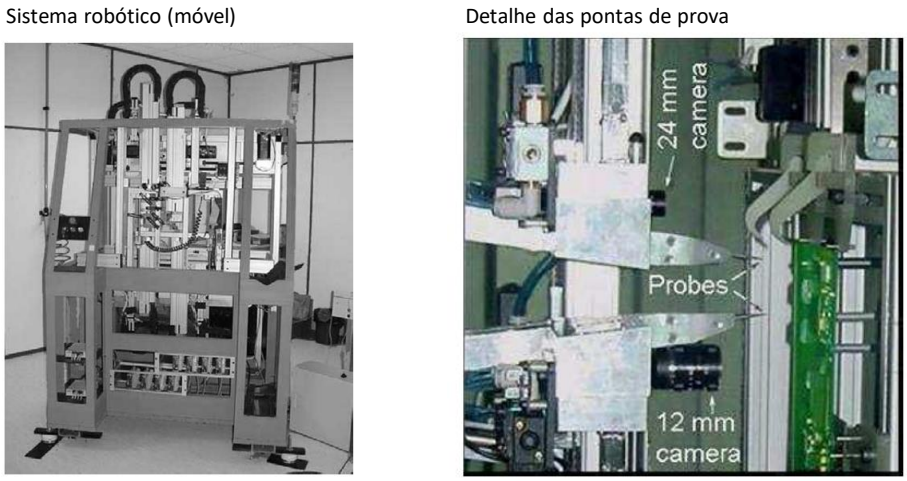
Figura 1 – Exemplo de sistema de testes com cama de agulhas (*fixture*). No detalhe à esquerda, é possível observar o trabalho artesanal realizado durante a montagem



Fonte: O Autor (2022).

Uma abordagem semelhante utiliza pontas de prova flutuantes (Gómez et al., 2007), que são acopladas a um sistema robótico móvel no plano x-y. O DUT, que repousa no plano x-y, pode ter qualquer ponto acessado por meio dos contatos móveis para envio ou recepção de sinais de teste. (ver ilustração na Figura 2).

Figura 2 – Sistema móvel de testes de PCIs



Fonte: Gómez *et al*. (2007).

O principal diferencial da solução proposta em Gómez et al. (2007) foi o desenvolvimento de uma interface visual para calcular deslocamento da placa testada, o que evita problemas de desalinhamento. Contudo, mesmo com a possibilidade de adaptação, sistemas como da Figura 2 são muito complexos e dispendiosos. Além de ter um grau de adaptabilidade restrito. Por exemplo, as dimensões do DUT são limitadas ao alcance físico do sistema robótico.

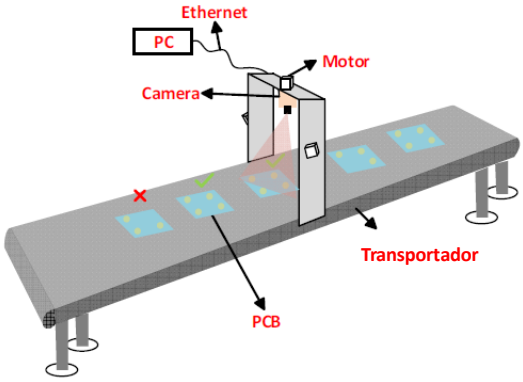
Uma forma alternativa, menos restrita, de teste de PCIs é alcançada por meio de processamento digital de imagens (PDI). As imagens da placa são capturadas por meio de câmeras e enviadas a um sistema computacional, onde algoritmos dedicados verificam a conformidade da PCI. Várias técnicas baseadas em PDI foram propostas. A maioria, entretanto, resolve apenas problemas pontuais do sistema de testes. No trabalho de Baygin et al. (2017), por exemplo, o arranjo da Figura 3 foi utilizado para adquirir imagens com o objetivo de definir a posição exata dos furos da placa e detectar possíveis furos ausentes.

Já o trabalho de Guo (2011) descreve uma solução para aperfeiçoar a imagem capturada da placa antes de ser processada, de modo a aumentar a confiabilidade dos testes realizados por meio de PDI.

Para PCIs com múltiplas camadas, a restrição de acesso às estruturas internas por meio de câmeras convencionais exige outras técnicas. Por exemplo, para detecção de erros nas camadas internas de forma não invasiva, é possível utilizar imagens de raio X. Assim, é possível verificar a anatomia da PCI sem ter que

separar as camadas, o que seria inviável. Em (Chuang et al., 2010), imagens de raio X foram aplicadas para detectar desalinhamento das conexões internas e externas em placas multicamadas.

Figura 3 – Configuração experimental para teste de PCIs por meio de PDI

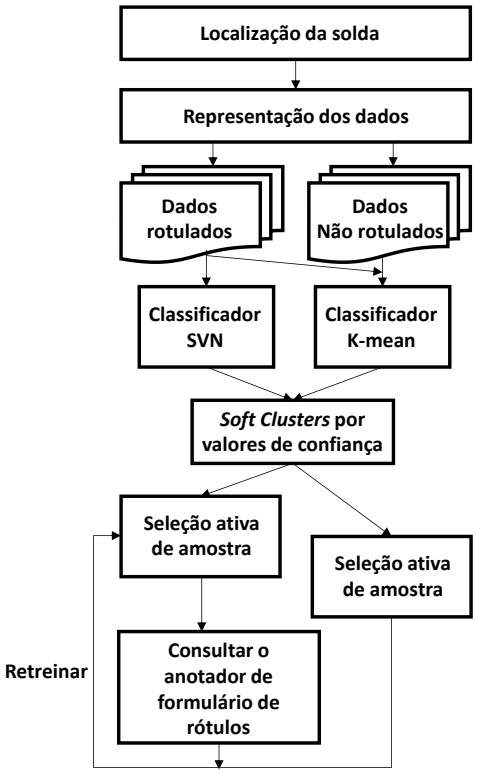


Fonte: Baygin *et al*. (2017).

De um modo geral, técnicas automáticas de teste de PCI por meio de imagens são denominadas AOI (*Automated/Automatic Optical Inspection*). Em AOI, existem duas abordagens fundamentais por meio das quais os testes são realizados: método referencial e método não-referencial (Taha, 2014). No primeiro caso, é feita uma comparação entre a imagem de teste e uma imagem de referência (*golden image*) (Tatibana, 1997). Nesse caso, o processamento exige mais tempo e é preciso haver alinhamento correto entre as imagens comparadas. O método não-referencial, por outro lado, não requer imagens de referência. Os testes são realizados por meio da observação de regras de projeto. Características da imagem de teste são comparadas com valores padrões. Apesar de ser mais simples e rápida, esta técnica é menos eficaz em detectar defeitos quando comparada com o método referencial. Uma forma de combinar as vantagens dos dois métodos descritos anteriormente, é usar uma abordagem híbrida (Cai, 2012).

Recentemente, AOI foi empregada em conjunto com técnicas de aprendizado de máquina para testes de PCIs. Em (Richter, 2017) a técnica de aprendizado por transferência (Yosinski et al., 2014) foi utilizada em uma rede neural pré-treinada para detectar erros em PCIs sem precisar utilizar imagens de referência. O trabalho de Dai et al. (2020) propôs uma arquitetura de rede neural artificial mista (ver Figura 4) capaz de realizar classificação e *clustering* (análise de agrupamento de dados) para detectar defeitos de soldagem em PCIs, obtendo uma precisão de até 95%.

Figura 4 – Arquitetura da rede neural artificial proposta em Dai et al. (2020)

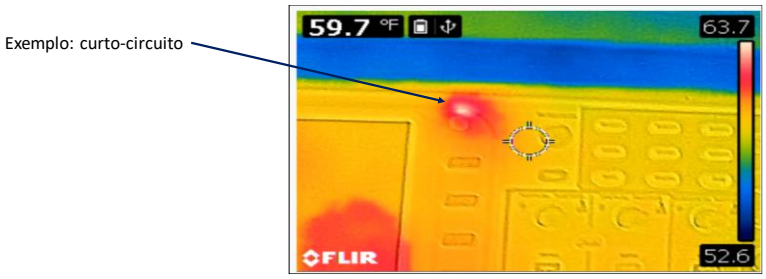


Fonte: Adaptado de Dai *et al*. (2020).

O trabalho de Silva (2019) também utiliza transferência de aprendizado e técnicas AOI combinados com ampliação de dados para construir redes neurais aplicadas a teste de PCIs. Ampliação de dados ou *data augmentation* (Shorten, 2019) é imprescindível quando há poucos dados disponíveis para o treinamento da rede neural. Seus resultados apresentam precisão de até 89%. Em (Acciani, 2006), foi proposto um sistema misto que utiliza rede neural, extração de características, AOI e análise *wavelet* para classificação de soldas em circuitos integrados. Os resultados apresentados alcançaram taxa de reconhecimento acima de 94%.

Outro método de teste não invasivo de circuitos eletrônicos utiliza câmeras infravermelhas para obtenção de imagens térmicas. Tradicionalmente, imagens térmicas têm sido utilizadas em inspeções não automatizadas, como no exemplo ilustrativo da Figura 5, em que a imagem é analisada visualmente por um especialista para encontrar indicativos de defeitos, como o curto-circuito na região de alta temperatura da placa.

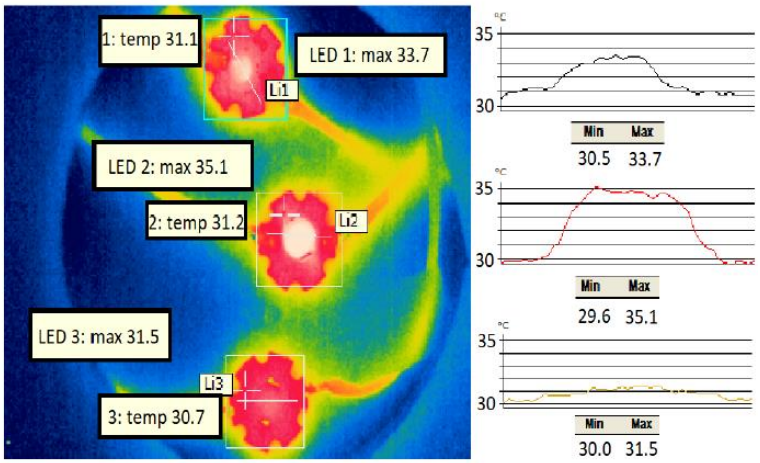
Figura 5 – Detecção visual (não automática) de um curto-circuito em uma região da PCI por meio de análise termográfica



Fonte: O Autor (2022).

Todavia, trabalhos recentes aplicaram a técnica termográfica em conjunto com processamento de imagens para automatizar o processo de teste. Por exemplo, o trabalho descrito em (Mamchur et al., 2020) utilizou termografia passiva para detectar automaticamente defeitos de soldagem por meio da caracterização térmica dos pontos de solda. A imagem da Figura 6 ilustra os resultados obtidos. A técnica permite definir a localização e o tipo do defeito detectado.

Figura 6 – Exemplo de imagem termográfica utilizada para detecção de defeitos em PCIs. À esquerda: imagens térmicas de três LEDs montados em uma PCI. À direita: tipos de soldagem. De cima para baixo: solda reativa; solda defeituosa; solda tradicional

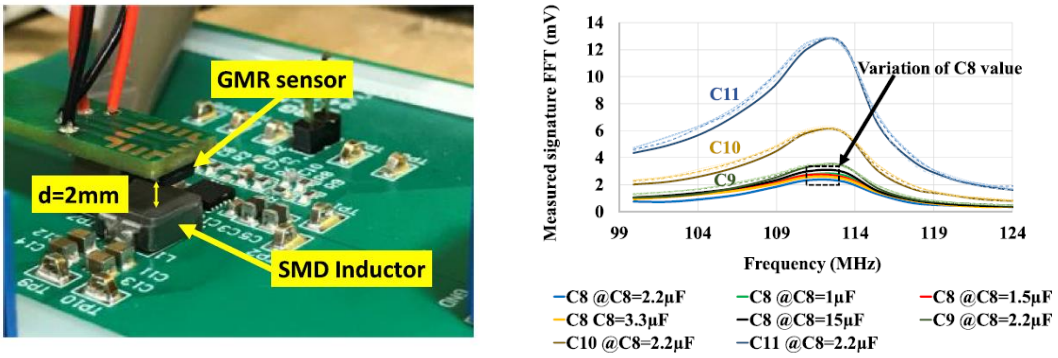


Fonte: Mamchur *et al*. (2020).

Dadas as características das PCIs e os desafios apresentados, realizar testes confiáveis e abrangentes (envolvendo aspectos de *hardware* e de *software*) de forma não invasiva ainda demanda técnicas e algoritmos inovadores. Alguns trabalhos publicados propuseram soluções que, entretanto, ainda apresentam limitações para aplicações práticas. Por exemplo, testes de PCIs por meio da assinatura da emissão eletromagnéticas (a curtas distâncias) de certos componentes eletrônicos foram propostos em (Alaoui et al., 2018; El Belghiti Alaoui

et al., 2018). Os resultados foram obtidos por meio de simulação Monte Carlo em modelos SPICE de circuitos analógicos e por meio de medições experimentais (Figura 7-a). O objetivo do sistema de teste é detectar variações do campo eletromagnético emitido causadas por componentes defeituosos. Entretanto, conforme declaração dos próprios autores “apenas componentes que possuem amplitude de emissão eletromagnética significativamente alta na região de campo próximo são testáveis”. Isso pode ser observado na Figura 7-b, que mostra a variação de valores medidos do campo eletromagnético capturado para vários capacitores do circuito. A variação do campo é decorrente da alteração (forçada, no caso do experimento) dos valores nominais dos componentes.

Figura 7 – Medição da assinatura de emissão eletromagnética (b) capturada por um sensor de campo próximo (a) de capacitores em um circuito eletrônico analógico



(a) (b)

Fonte: El Belghiti Alaoui *et al*. (2018).

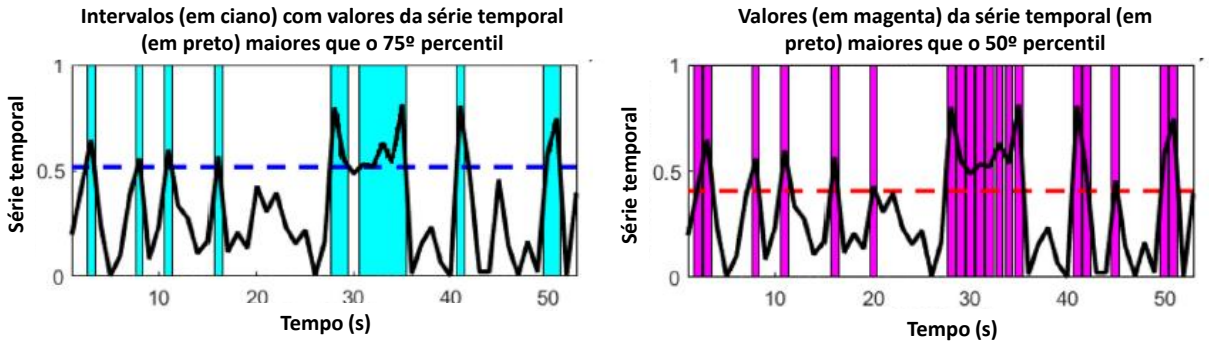
Nas curvas do gráfico é possível perceber que a amplitude da variação é bem limitada. Portanto, basicamente capacitores de alta potência apresentaram resultados experimentais aceitáveis. Essa restrição limita a aplicabilidade da solução em circuitos reais.

Outra possibilidade de implementar testes não invasivos é por meio de autoencoders, em que imagens obtidas dos sistemas livres de anomalia são usadas para treinar a rede. Essa solução é denominada de classificador de classe única, pois, o objetivo da rede é determinar se o DUT pertence à classe livre de defeito ou não. Ou seja, o resultado do teste é do tipo aprovado ou reprovado. Dadas as características e a forma de funcionamento do autoencoder, se uma imagem não pertencente à classe utilizada no seu treinamento for apresentada à sua entrada, a imagem reconstruída é muito diferente da imagem original (mais detalhes nas

Seções 3.1.5 e 3.8). Isso significa que o erro medido como a diferença entre as duas imagens pode ser usado para separar a imagem de entrada em duas classes. A aplicação de autoencoders treinados com imagens de placas sem anomalia foi estudada no trabalho de Mujeeb et al. (2019). O sistema proposto foi utilizado para detectar defeitos de solda e defeitos mecânicos (arranhões) em PCIs. Os resultados foram apresentados em termos de Taxa de Falsos Positivos e Taxa de Falsos Negativos. Os valores obtidos foram comparados com um algoritmo de comparação de imagens (SIFT - *Scale-Invariant Feature Transform*) (Lowe, 2004). A restrição da solução apresentada nesse trabalho é a dificuldade de se obter muitas imagens diferentes de placas sem defeito, o que limita o processo de treinamento do autoencoder.

Na linha de pesquisa semelhante ao nosso estudo, várias propostas de aplicação de autoencoders para detecção de anomalia foram publicadas nos últimos anos. Vários desses estudos estão analisados no trabalho de revisão (Chalapathy, 2019). Mais recentemente, a aplicação de autoencoders em detecção de anomalias em produtos industrializados (especificamente, máquinas de lavar) foi descrita em (Alfeo et al., 2020). Nesse trabalho, autoencoders são usados para reconstruir dados obtidos do comportamento de séries temporais que representam sinais de vibração e consumo de energia. Exemplos dessas séries estão ilustradas na Figura 8, onde a amplitude é comparada com as características estatísticas (percentil 75% e 50% do valor máximo) do sinal.

Figura 8 – Extração de características da série temporal

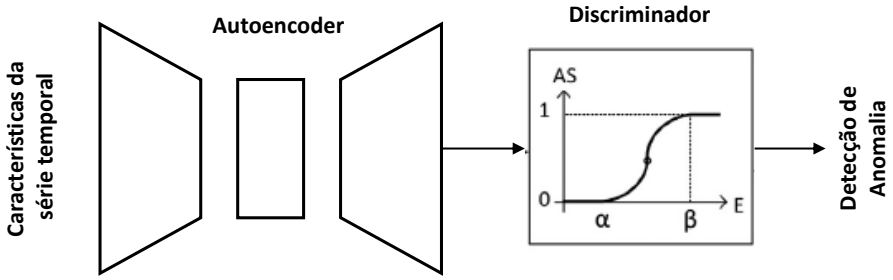


Fonte: Adaptado de Alfeo *et al*. (2020).

As características da série temporal são enviadas à entrada do autoencoder treinado com dados de produtos sem anomalia. A saída, que corresponde ao dado reconstruído é enviado a um discriminador sigmoidal que transforma o erro médio

quadrático da reconstrução em um valor entre 0 e 1 para indicar anomalia (1) ou normalidade (0) de operação do sistema testado. A estrutura da proposta está ilustrada na Figura 9.

Figura 9 – Estrutura de autoencoder com saída conectada a um discriminador sigmoidal



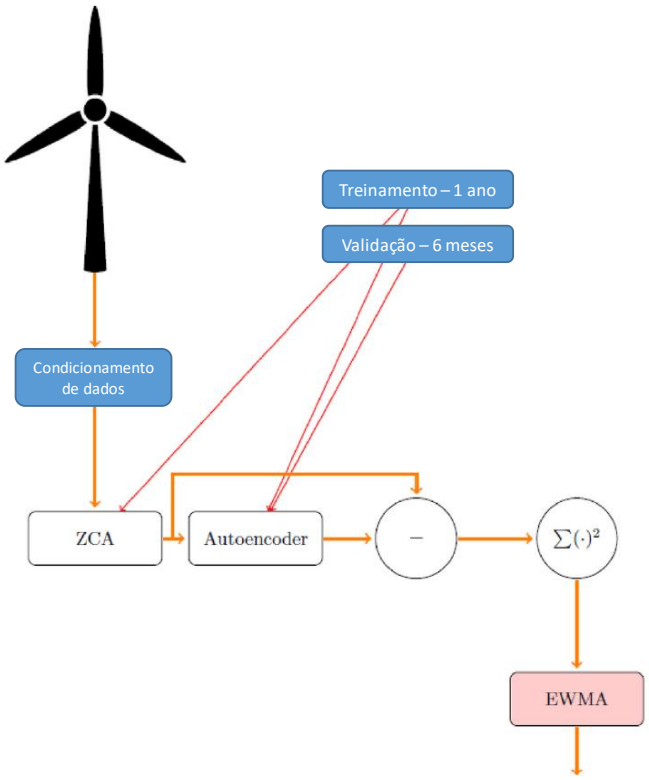
Fonte: Adaptado de Alfeo *et al*. (2020).

O desempenho da rede proposta foi comparado com resultados obtidos aplicando Floresta de Isolamento (Liu, 2008) para o comportamento da energia consumida e da vibração de máquinas de lavar.

Em outra publicação relacionada ao nosso tema de pesquisa, um sistema de prevenção de anomalias em sistemas de geração de energia eólica foi implementado também utilizando autoencoders (Renström et al, 2020). Neste caso, os sinais monitorados são provenientes do sistema SCADA (*Supervisory Control And Data Acquisition*, Boyer, 1999). Ao todo, 33 sinais foram utilizados, entre eles, velocidade e direção de vento, velocidade de giro do rotor e temperatura do óleo da engrenagem. O fluxograma da metodologia apresentada no trabalho está ilustrado na Figura 10.

Primeiramente, os dados coletados da turbina são condicionados e depois pré-processados usando o método de análise ZCA (*on Zero-Phase Component Analysis*). Em seguida, os dados são enviados ao autoencoder treinado, que os tenta reconstruir em sua saída. O erro de reconstrução é filtrado usando EWMA – *Exponentially Weighted Moving Average* (Hunter, 1986). Essa filtragem é necessária devido ao longo período de tempo de aquisição dos sinais envolvidos. O treinamento utiliza dados obtidos ao longo de um ano, e a validação utiliza dados coletados por seis meses, o que torna a implantação do sistema bem restritiva.

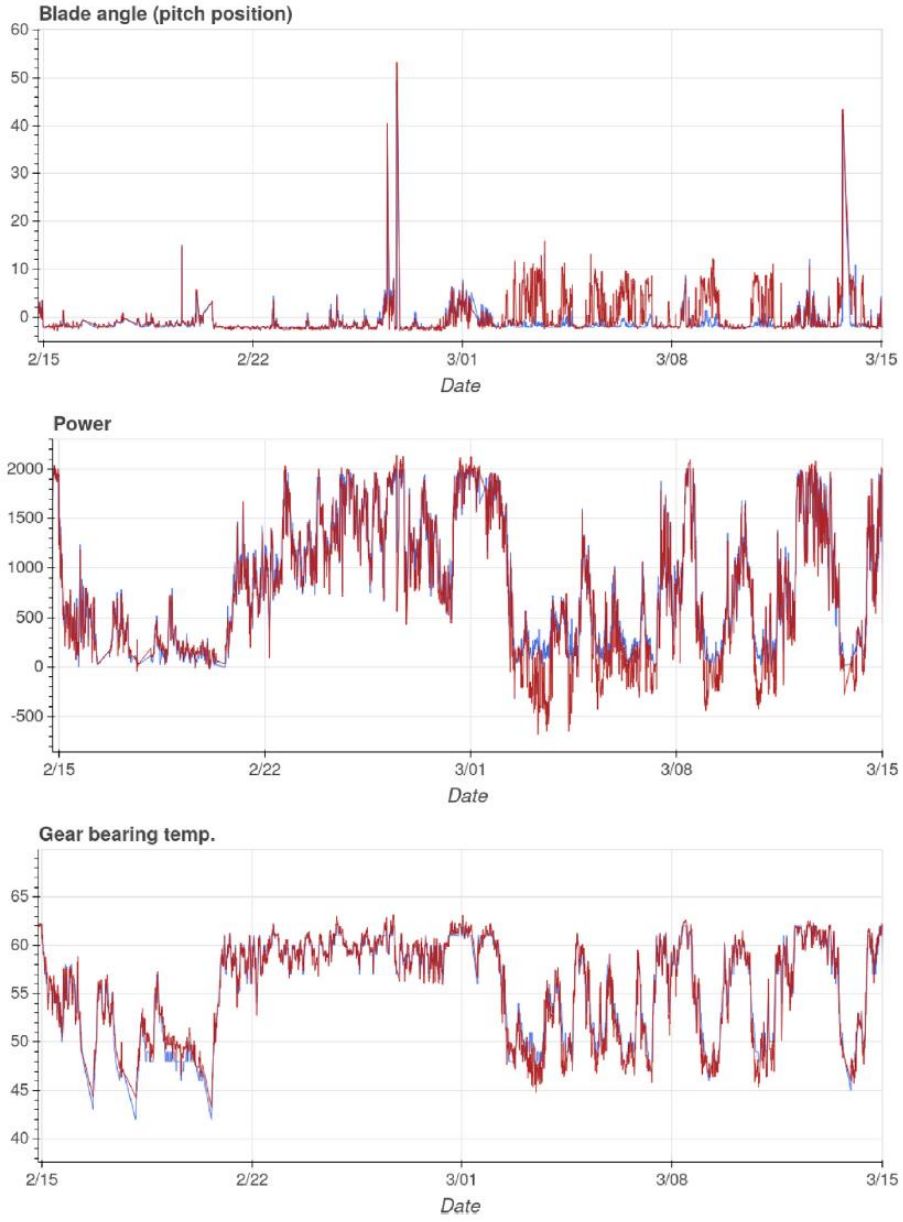
Figura 10 – Sistema de monitoramento de turbinas eólicas a partir de dados do sistema SCADA



Fonte: Adaptado de Renström *et al*. (2020)

Para ilustrar como funciona o sistema proposto, a Figura 11 mostra os sinais originais (azul) e suas reconstruções (vermelho) durante um período em que uma anomalia no sistema de rotação das hélices foi detectada. Os sinais diretamente relacionados ao sistema de rotação (*Power* e *Blade angle*, na figura) apresentam discrepância entre original e reconstrução. Ao mesmo tempo, para o sinal não relacionado à rotação (*Gear bearing temp.*), a diferença é quase inexistente. Segundo os autores, isso indica que o sistema de detecção de anomalia também pode determinar onde (em que parte do sistema da turbina) a anomalia irá ocorrer.

Figura 11 – Sinais originais e sua reconstrução para um período em que houve detecção de anomalia no sistema de rotação do conjunto de hélices

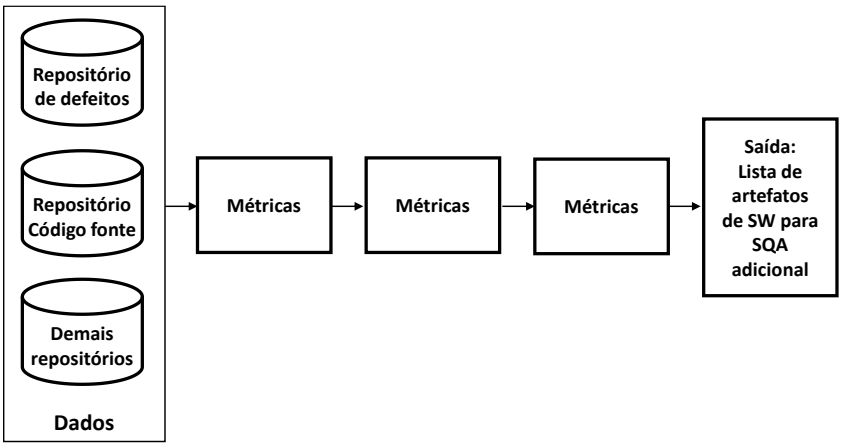


Fonte: Renström *et al*. (2020).

As soluções apresentadas anteriormente são basicamente dedicadas aos testes estruturais (de *hardware*). Em relação ao *software*, muitas técnicas de testes automatizados foram desenvolvidas e têm sido sistematicamente utilizadas (Kamei, 2016). Entretanto, geralmente as peculiaridades do *software* de sistemas embarcados não são consideradas nesses estudos. As soluções propostas buscam, por meio de análise baseada em métricas (Moser, 2008) de *software* de aplicação geral (não embarcada), prever a ocorrência defeitos (bugs). A Figura 12 mostra uma visão geral do processo de previsão de anomalias do *software*. O modelo de previsão é construído com base em informações de histórico e definição de

métricas. Os principais objetivos da predição são: classificar a propensão ao erro do *software* testado; prever o número de defeitos que podem ocorrer; estimar a probabilidade de ocorrência de um erro de *software*. A aplicação direta desses modelos em testes de sistemas embarcados, onde anomalias apresentadas durante a execução precisam ser detectadas, pode não apresentar resultados satisfatórios.

Figura 12 – Visão geral do procedimento de detecção de anomalias em *software*



Fonte: Adaptado de Moser (2008).

Com o aumento da complexidade dos sistemas embarcados, sobretudo a partir dos anos 1990, técnicas de projeto concorrente de *hardware* e de *software* – denominadas *hardware/Software co-design* (Kra, 1993) – passaram a ser estudadas e aplicadas com o objetivo de verificar a conformidade dos sistemas embarcados de forma unificada (Mukherjee et al., 2017). Mais recentemente, estudos aplicados a sistemas embarcados apresentaram soluções para previsão de anomalias em *software* embarcados (Zong, 2018; Singh, 2017; Manjula, 2019). As soluções propostas nesses trabalhos utilizam modelos de aprendizado de máquina ou redes neurais artificiais e métricas de *software*. Contudo, assim como o caso do *software* de alto nível, o objetivo é prever ocorrência de anomalias de *software* em um sistema embarcado e não as detectar.

2.2 CLASSIFICAÇÃO DE SUBSTÂNCIAS

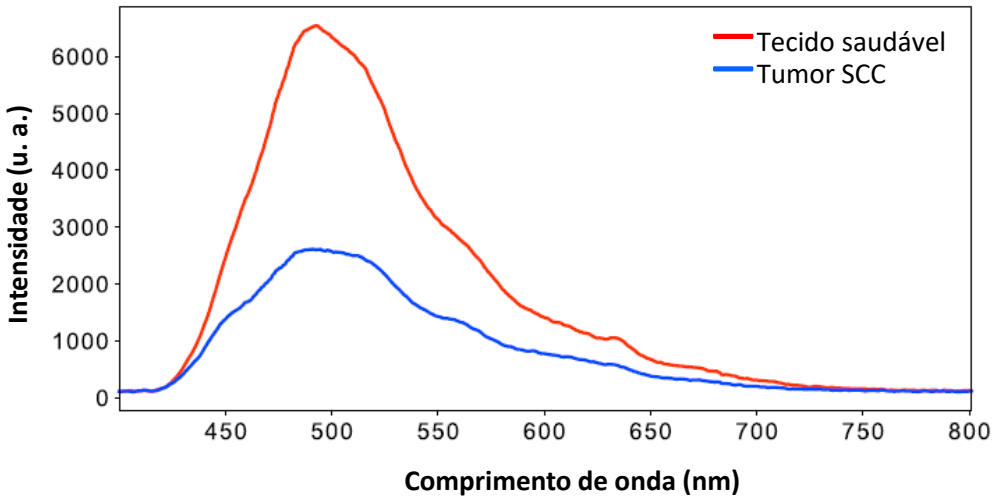
A detecção de substâncias químicas ou biológicas é um problema comum nas indústrias modernas (Plevridis, 1995). O método amplamente utilizado para realizar tais procedimentos é a espectroscopia (Berman, 2011), que utiliza um instrumento denominado espectrômetro. Apesar da maturidade da técnica, os espectrômetros

são bastante caros e muitas vezes complexos e compostos de partes delicadas. Além disso, para testes remotos, o tamanho do próprio equipamento é um fator limitante (Xie et al., 2015).

De qualquer forma, pesquisas têm sido realizadas em que técnicas de aprendizado de máquina juntamente com espectroscopia são aplicadas nas mais diversas áreas. O trabalho de Tsakanikas et al. (2020), por exemplo, teve por objetivo desenvolver processo de aprendizado baseado em regressão PLS (*Partial Least Squares Regression*) e classificação SVM (*Suport Vector Machine*), para automação da categorização de alimentos usando espectroscopia FTIR (Fourier *Transform Infrared Spectroscopy*). O sistema exibiu alta eficiência na classificação de 7 tipos diferentes de alimentos crus.

Um exemplo de aplicação na área de saúde é o trabalho de Nogueira et al. (2021), onde uma rede neural foi projetada e usada para a diferenciação de pele saudável e com Tumor SCC (*Squamous Cell Carcinoma*) em camundongos, a partir de conjuntos de dados de espectroscopia de fluorescência. A Figura 13 traz uma ilustração do artigo, mostrando espectros (neste caso, de fluorescência) para duas classes de amostra: saudável e não-saudável. O modelo apresentou sensibilidade de 85,5%, especificidade de 92,86% e uma acurácia de 90%.

Figura 13 – Espectros de fluorescência típicos de pele saudável (*healthy tissue*) e com câncer (SCC tumor)

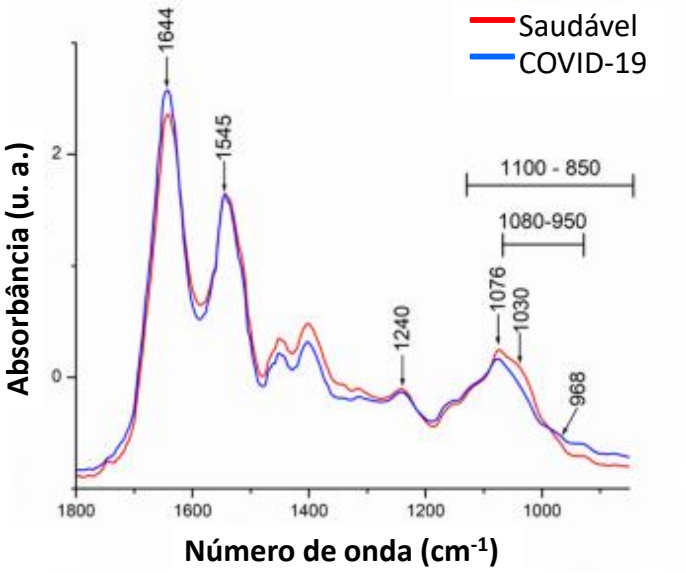


Fonte: Adaptado de Nogueira *et al*. (2021).

A publicação de Martinez-Cuazitl et al. (2021) traz uma pesquisa interessante em que os espectros de absorção no infravermelho da Figura 14 foram usados para

distinguir amostras de saliva saudável de contaminadas pelo vírus do COVID-19. Para a classificação das amostras, um modelo de regressão linear multivariada (MLRM) foi utilizado.

Figura 14 – Espectros brutos de FTIR de grupos saudáveis (n=1209) e COVID-19 (n=255)



Fonte: Martinez-Cuazitl *et al*. (2021).

Em face da quantidade e da qualidade das publicações relacionando espectroscopia de infravermelho e modelos de aprendizado de máquina, conclui-se que a demanda por soluções inovadoras e o potencial de aplicação é bastante vasto. Para finalizar, indicamos exemplos adicionais de trabalhos com possível aplicação industrial, que podem se beneficiar da abordagem proposta em nosso trabalho: indústria química (Sidhik, 2015), indústria alimentícia (Rajalakshmi et al., 2017); indústria da segurança (reconhecimento de drogas) (Dong et al., 2015); saúde pública (status de multiparidade dos mosquitos selvagens) (Milali et al., 2020); ramo farmacêutico (Tian et al., 2010).

2.3 APERFEIÇOAMENTO DE MODELOS DE APRENDIZADO DE MÁQUINA

Especificamente no campo da detecção de anomalias em sistemas embarcados, o aperfeiçoamento de modelos de aprendizado de máquina por meio da transformação de dados não é pesquisado com muita frequência. No entanto, técnicas similares têm sido aplicadas em outras áreas.

Em (Abedin et al., 2020), por exemplo, os autores propõem aplicar transformações às características extraídas dos dados para melhorar a previsão de

inadimplência fiscal. O trabalho descrito em (Suthar et al., 2017) aplicou a transformação de estiramento de fase para melhorar a detecção de bordas em imagens com deficiência visual. O estudo descrito em (du Pin Calmon et al., 2018) tem um objetivo empolgante: mitigar a discriminação algorítmica (Calders, 2013). Ele afirma que os conjuntos de dados transformados levaram a classificações mais justas quando comparados aos conjuntos de dados originais. Em relação a testes de *software*, os autores de (Zhang, 2017) investigaram o efeito de múltiplas transformações em um sistema de previsão de defeitos de *software*. O objetivo da transformação é aumentar a normalidade dos dados de métricas de *software* (Bishara, 2015).

Mais recentemente, o trabalho de Xu et al. (2021) alega usar uma metodologia inovadora de pré-processamento de dados que transforma os conjuntos de dados de entrada de modo que contenha amostras mais equilibradas em termos de tipo e tamanho, bem como remove valores atípicos (*outliers*) que afetariam o viés de detecção. A metodologia é demonstrada com um autoencoder aplicado em detecção de anomalias de rede de comunicação. O método proposto atingiu uma acurácia de 90.61%, superando outros modelos previamente publicados.

No trabalho de Sana et al. (2022), ainda em fase de apreciação (*preprint*), foi realizado um estudo em modelo de predição da rotatividade de clientes de empresas de telecomunicações. Os autores utilizaram métodos de transformação de dados – *Log, Rank, Box-cox, Z-score*, Discretização e *Weight-of-evidence* (WOE) – para melhorar o desempenho de predição. Vários modelos de aprendizado de máquina foram considerados nesse estudo. Os resultados experimentais indicam que, “na maioria dos casos”, os métodos de transformação de dados melhoram a qualidade dos dados e melhoram o desempenho.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, apresentamos uma visão geral das ferramentas mais relevantes usadas ao longo de nossa pesquisa. Inicialmente, os conceitos de aprendizado de máquina são revisitados.

Dos modelos de aprendizado de máquina, destacamos os autoencoders, que são os mais utilizados em nosso trabalho. Como usamos o mesmo modelo de autoencoder para detecção de anomalias e classificação de substâncias, apresentamos neste capítulo seu desenvolvimento realizado na linguagem Python (Francois, 2017).

Em seguida, discutimos os conceitos necessários à implementação das soluções propostas, como espectrogramas, termografia e espectroscopia. Além disso, as transformadas propostas para o aperfeiçoamento do modelo são apresentadas. Também definimos as métricas utilizadas para comparação e validação dos resultados obtidos.

3.1 APRENDIZADO DE MÁQUINA

Máquinas não são inteligentes por natureza. Em princípio, são desenvolvidas para desempenhar funções específicas, e.g., transporte de pessoas e transmissão de informações. A principal vantagem das máquinas é que elas desempenham tais funções de maneira mais rápida e precisa que os seres humanos (Francois, 2017). O aspecto capital que diferencia a máquina de uma pessoa é a capacidade, ou falta dela, de executar trabalhos utilizando inteligência. O cérebro humano recebe um conjunto de informações ou estímulos do ambiente a partir dos 5 sentidos. Tais informações são levadas ao cérebro por meio de algumas estruturas do sistema nervoso. Chegando ao cérebro, a informação é organizada, comparada a outras informações já armazenadas na memória e então interpretada. Em seguida, o cérebro toma uma decisão e envia os sinais de controle necessários para que o organismo responda de maneira adequada aos estímulos do ambiente. Informações deste processo podem ser armazenadas na memória e utilizadas em situações futuras, configurando o aprendizado pela experiência. As máquinas não são capazes de lidar com informações de maneira inteligente, pois elas não interpretam informações ou aprendem com a experiência.

Daí surge o aprendizado de máquina, uma área da ciência que objetiva tornar as máquinas capazes de analisar situações e tomar decisões de modo semelhante aos seres humanos (Fong, 2018). De acordo com Mitchell (1997), “o aprendizado de máquina é um produto natural da interseção entre a ciência da computação e a estatística”. Enquanto a primeira área de estudo raciocina sobre como as máquinas podem ser construídas para resolver problemas, a estatística trabalha sobre o que pode ser inferido, com determinado nível de confiabilidade, a partir de um conjunto disponível de informações.

Em linhas gerais, o aprendizado de máquina é um ramo da inteligência artificial (Hunt, 2014) cujo objetivo é, com o uso de algoritmos, tornar as máquinas capazes de desempenhar certas funções, como o reconhecimento de objetos em imagens, de maneira hábil. Esta área de conhecimento geralmente inclui questões relacionadas ao desenvolvimento ou seleção de arquiteturas e algoritmos que sejam capazes de capturar, organizar, armazenar, combinar e recuperar informações de modo eficiente e efetivo. O aprendizado de máquina visa projetar modelos que recebam informações do ambiente e forneçam determinadas respostas.

3.1.1 Conceitos fundamentais

I. Dados:

Todas as técnicas de aprendizado de máquina são direcionadas a dados. O treinamento dos sistemas é realizado com conjuntos de dados, que podem ter diversas origens, e.g., medições de sensores ou levantamentos estatísticos. Quanto maior a quantidade de dados em um conjunto, maior a quantidade de informações que se pode extrair dele (Haykin, 2004).

II. Modelos:

O modelo é a estrutura que propicia uma caracterização matemática para o processo de aprendizado de máquina (Haykin, 2004).

III. Treinamento:

Sistemas inteligentes necessitam passar por um processo de treinamento para que as relações entre suas entradas e suas saídas sejam construídas de maneira adequada (Haykin, 2004). Esse processo define o próprio aprendizado e pode ser

feito de forma supervisionada, não-supervisionada ou ainda de forma semi-supervisionada.

a. Aprendizado supervisionado:

O objetivo deste tipo de aprendizado é inferir funções ou mapeamentos a partir de conjuntos de dados rotulados (Cunningham, 2008). Rótulos associam os dados de entrada a uma resposta esperada, sendo atribuídos por um agente externo, o supervisor. Os algoritmos que pertencem ao grupo de técnicas de aprendizado supervisionado são os de regressão e de classificação. Problemas que envolvem esta forma de aprendizado são mais frequentes em aplicações reais.

b. Aprendizado não-supervisionado

Nesta categoria de aprendizado de máquina, nota-se a ausência de um supervisor ou de informações rotuladas (Celebi, 2016). O objetivo é encontrar estruturas ocultas dentro do conjunto de dados. Não existe uma resposta correta, mas um conjunto de respostas possíveis. *Clustering* (análise de agrupamento de dados) é um exemplo de técnica pertencente a esta categoria. Algoritmos de aprendizado não-supervisionado são utilizados frequentemente no pré-processamento dos dados, extraindo características que são utilizadas como informações de entrada para os algoritmos supervisionados.

Clustering constitui uma das famílias de algoritmos de aprendizado de máquina não supervisionado mais conhecidas (Cios et al., 2007). Ele tem como objetivo encontrar estruturas ocultas em um conjunto de dados não rotulados, os *clusters*. *Clusters* constituem um conjunto de objetos que apresentam semelhanças entre si, ao mesmo tempo em que diferem de objetos pertencentes a outros *clusters*. Não existe um resultado correto para o processo de *clustering*. A adequabilidade do resultado depende do problema. Os principais requerimentos para um algoritmo de *clustering* envolvem escalabilidade, capacidade de lidar com diferentes tipos de atributos, de encontrar *clusters* em formas arbitrárias, de lidar com ruídos e *outliers*, com altas dimensionalidades ou com a usabilidade.

c. Aprendizado semi-supervisionado

Neste tipo de aprendizado há uma mistura de informações com e sem rótulo. A combinação é utilizada na construção de modelos para a classificação dos dados

(Zhu, 2005). É um modelo de aprendizado adequado para situações em que há escassez de dados rotulados e abundância de informações sem rótulo.

IV. Validação e avaliação (teste) do modelo:

Não é suficiente que o modelo se adeque perfeitamente ao conjunto de dados utilizado no treinamento do modelo. Também é necessário avaliar o desempenho em um conjunto de testes. Este tipo de ação evita um fenômeno conhecido como *overfitting* (Chollet, 2021), que ocorre quando o desempenho da função é alto apenas com os dados utilizados no treinamento, sendo baixo com aqueles que não fizeram parte deste processo (mais detalhes na Seção 3.1.3). O fenômeno está relacionado à capacidade de generalização do sistema.

Hold-out (Dwork et al., 2015) e validação-cruzada (Francois, 2017) são duas técnicas utilizadas para avaliar a capacidade de generalização do modelo obtido. No *hold-out*, a maior parte dos dados disponíveis é utilizada no treinamento do modelo, sendo o restante utilizado para testá-lo após a conclusão do treinamento. A divisão dos dados entre grupos de treino e de teste deve ser feita de maneira aleatória, de modo a evitar resultados tendenciosos. Na validação-cruzada, o conjunto de dados é inicialmente dividido em grupos. Na abordagem básica, chamada validação-cruzada de n conjuntos, o conjunto de treinamento é dividido em n conjuntos menores. O seguinte procedimento é seguido para cada um dos n conjuntos:

- um modelo é treinado usando os conjuntos resultantes como dados de treinamento;
- o modelo resultante é validado na parte restante dos dados (ou seja, é usado como um conjunto de teste para calcular uma medida de desempenho, como precisão).

Ou seja, cada grupo é tratado como um conjunto de teste, sendo os demais grupos utilizados no processo de treinamento, e ocorre um revezamento dos grupos que são utilizados para treino e teste. O desempenho é avaliado de acordo com os resultados dos conjuntos de teste, sendo o desempenho geral do modelo a média dos desempenhos obtidos em cada um dos conjuntos individuais de teste.

Em nossos experimentos, o autoencoder, que é o modelo mais utilizado, é treinado usando a validação cruzada. Nas descrições, para cada caso estudado, o

tamanho do conjunto (treinamento, validação e teste) é explicitamente especificado para tornar a escrita menos dúbia.

3.1.2 Extração de características:

Em sistemas de classificação, características são variáveis ou atributos de entrada, e.g., na diagnose de doenças, sintomas podem ser vistos como características (Zebari et al., 2020). A extração de características é uma técnica bastante utilizada em aplicações que envolvem aprendizado de máquina. O processo consiste em obter uma série de medidas a partir de um dado conjunto de objetos. O uso da técnica tem por objetivo resumir a informação disponível e facilitar a separação das classes. Com isso, também é possível diminuir complexidade do classificador, eliminar ruído e, na maioria dos casos, melhorar a precisão do processo classificatório.

A redução de custo computacional e o aumento da precisão de classificação são os principais objetivos dos processos de extração e seleção de características. Os Conceitos Básicos relativos à extração de características são:

I. Vetor de Características

É o vetor obtido por meio do processo de extração de características (Baudat, 2003). Esse vetor apresenta as informações resumidas de um objeto em particular. A sua dimensionalidade é inferior à do conjunto de informações original. O vetor de características é associado a uma determinada classe por classificadores que utilizam diferentes critérios de classificação.

II. Padrão:

O padrão é a descrição estrutural ou quantitativa de um objeto, sendo comumente organizado na forma de um vetor de características (Han, 2022). As características de um determinado padrão podem ser valores discretos ou contínuos, sendo apenas necessário que elas sejam capazes de descrever um determinado objeto, diferenciando-se ao máximo daquelas que descrevem os demais objetos.

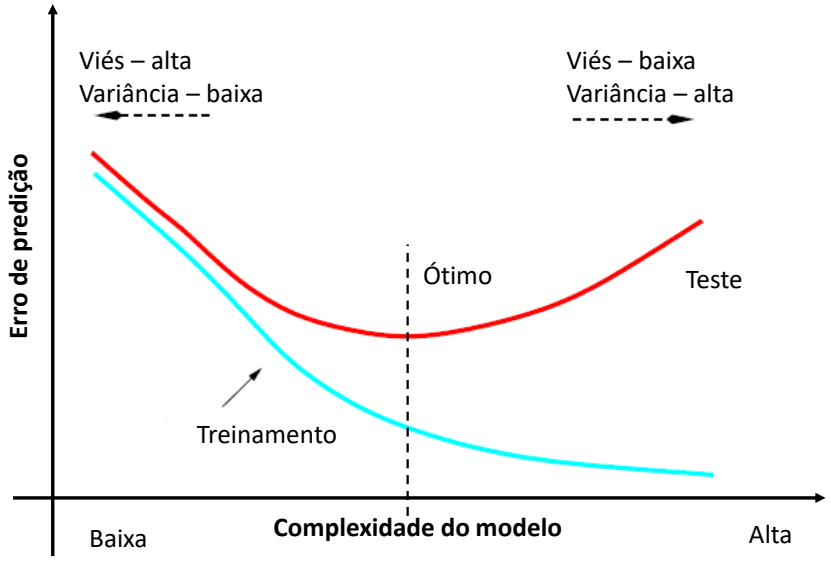
III. Classe:

É um conjunto de padrões que apresentam propriedades em comum (Han, 2022). As classes são comumente representadas por distribuições de probabilidade. Isso se deve à diversidade dos objetos, que apresentam diferenças mesmo quando pertencem a uma mesma classe. Como o processo de classificação visa ao reconhecimento destes padrões, é necessário que as distribuições de probabilidade sejam conjuntos separáveis e não vazios (ver detalhes práticos na Seção 0).

3.1.3 Dilema viés-variância

Existe uma relação entre a complexidade do modelo de aprendizado de máquina e sua capacidade de apresentar bom desempenho quando processa dados que não foram utilizados no treinamento. É o que se costuma chamar dilema viés-variância. A Figura 15 resume de maneira ilustrada esse conceito. O esquema mostra o típico erro de predição em função da complexidade do modelo para dois conjuntos de dados: de treinamento e de teste. Observe como o viés sempre diminui com a complexidade do modelo. Contudo, a variância, ou seja, a flutuação no desempenho devido ao tamanho finito do conjunto de treinamento, aumenta. Em consequência, o modelo pode apresentar o problema chamado de *overfitting* (Chollet, 2021), o que leva a uma queda no seu desempenho preditivo.

Figura 15 – Visão geral da estrutura de um autoencoder típico



Fonte: Adaptado de Mehta *et al*. (2019).

Quando o modelo apresenta bom desempenho com os dados de teste, porém não consegue generalizar o resultado para dados novos apresentados à sua entrada, o modelo não deve ser considerado para aplicações práticas.

3.1.4 Ferramentas computacionais

Para o desenvolvimento de nossa pesquisa, utilizamos várias ferramentas computacionais, seja para realizar a aquisição dos dados ou para executar treinamento e teste dos modelos construídos. A seguir, listamos os principais sistemas computacionais e suas características, que servem como referência para contextualizar a complexidade de cada modelo, e g., o tempo de treinamento do autoencoder.

Para aquisição dos dados de corrente elétrica e termográficos, usamos um computador pessoal portátil equipado com um processador Core i7, 1TB de memória externa (SSD - *Solid State Device*) e 16 GB de memória interna (RAM). Para treinamento dos modelos, utilizamos um computador portátil equipado com GPU Nvidia Quadro P600 com 4 GB de GDDR5, processador Intel® Core™ i7-8750H de 6 núcleos e memória externa SSD de 1 TB, e memória interna de 16 GB. O uso de GPU juntamente com a biblioteca Cuda (Parvat, 2017) é importante para reduzir o tempo de treinamento, sobretudo nos casos em que muitos dados são usados nesse processo.

Para análise e processamento dos dados, várias linguagens e bibliotecas foram utilizadas. O desenvolvimento dos modelos utilizados na análise preliminar, como árvores de decisão e *random forest* foi executado na linguagem R. O autoencoder, por sua vez, foi implementado usando Tensorflow (Pang, 2020), Keras (Francois, 2017) e Cuda, que estão disponíveis para a linguagem Python (Francois, 2017).

Para construção de gráficos e curvas usados na análise dos resultados, assim como para o processamento dos dados obtidos após os testes dos modelos, foi utilizado majoritariamente o *software Matlab* (Pietruszka, 2012).

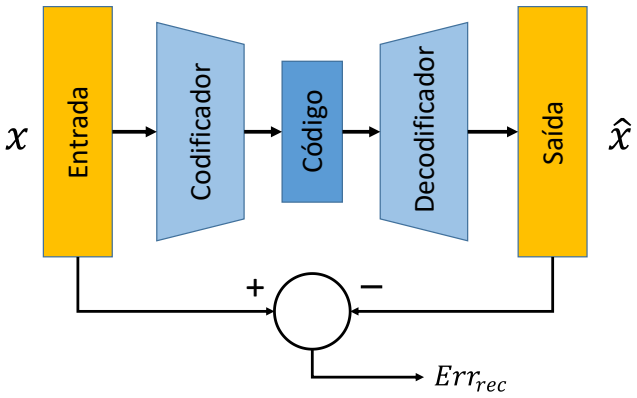
3.1.5 Autoencoders

Autoencoders (ou auto codificadores) podem ser vistos como redes neurais artificiais treinadas para aprender uma representação dos dados de sua entrada e,

baseando-se nessa representação, reconstruir o dado de entrada na sua saída (Goodfellow, 2016). A representação da informação, chamada codificação, pode ser usada para várias finalidades como redução dimensional dos dados ou detecção de anomalias (Sakurada, 2014).

Um autoencoder consiste de uma camada de entrada, uma ou mais camadas escondidas intermediárias, e uma camada de saída. Tradicionalmente, essas redes apresentam uma estrutura simétrica, na qual as camadas de entrada e de saída possuem o mesmo número de neurônios. A estrutura típica de autoencoder está mostrada na Figura 16, onde a entrada (x) é uma imagem convertida em vetor de comprimento N. Essa conversão ajuda a simplificar o processamento em uma aplicação computacional. A saída ($\hat{x}$) é a cópia reconstruída da entrada.

Figura 16 – Visão geral da estrutura de um autoencoder típico



Fonte: O Autor (2022).

A cópia fornecida pelo autoencoder não é idêntica à entrada, portanto, existe um erro residual que pode ser calculado como a diferença entre as duas imagens. Definimos essa diferença como erro de reconstrução do autoencoder, que é dado pela equação (1):

$$Err_{rec} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2. \qquad (1)$$

onde x_i e $\hat{x}_i$ são os elementos dos vetores que correspondem à entrada e à saída, respectivamente. Neste trabalho, autoencoders são utilizados para detectar anomalias e classificar substâncias utilizando imagens geradas a partir de espectrogramas (Seção 3.3) de sinais temporais. Esse processo é feito por meio do

treinamento do autoencoder com imagens pertencentes à classe padrão (ver definição de classes na Seção 0). O erro de reconstrução de imagens pertencentes à classe de treinamento é, em geral, menor do que o erro associado a imagens que pertencem a uma classe distinta daquela usada para treinamento.

A implementação do autoencoder envolve três etapas: definição da estrutura (tamanho, número de neurônios, número de camadas, etc.); treinamento e validação; e teste. A etapa de treinamento é equivalente ao aprendizado da rede neural. Nesta etapa, seus parâmetros são atualizados de forma a minimizar o erro de reconstrução para um conjunto de imagens apresentadas em sua entrada (dados de treinamento). Uma época denota o número de vezes que todos os dados de treinamento passaram pela rede neural no processo de treinamento/validação.

Como é o caso para toda rede neural, o treinamento do autoencoder é um processo não determinístico. Portanto, é comum que sessões distintas de treinamento apresentem resultados ligeiramente diferentes entre si. Por isso, nós treinamos a rede múltiplas vezes para garantir que os resultados sejam apresentados de forma estatisticamente relevante. Para evitar o fenômeno conhecido por *overfitting* (Seção 3.1.1) – quando o modelo se adapta bem aos dados com os quais está sendo treinado, porém, não generaliza para novos dados – aplicamos o procedimento de validação cruzada durante o treinamento do autoencoder.

Em nossos estudos, o autoencoder é implementado usando a linguagem Python e a biblioteca de código-aberto para redes neurais Keras (Francois, 2017). O autoencoder que utilizamos é baseado em uma estrutura famosa (Francois, 2016), na qual as camadas consistem de redes convolucionais e filtros. Um trecho do código em Python está ilustrado na Figura 17 e o resumo do modelo está representado na Figura 18.

Figura 17 – Trecho do código em Python onde o modelo é definido

```
#Definição do Modelo
#-------------------------------------------------------------------------------
        #ENCODER
inp = Input((96, 128,3))
e = Conv2D(32, (3, 3), activation='relu')(inp)
e = MaxPooling2D((2, 2))(e)
e = Conv2D(64, (3, 3), activation='relu')(e)
e = MaxPooling2D((2, 2))(e)
e = Conv2D(64, (3, 3), activation='relu')(e)
l = Flatten()(e)
l = Dense(2304, activation='softmax')(l)

        #DECODER
d = Reshape((24,32,3))(l)
d = Conv2DTranspose(64,(3, 3), strides=2, activation='relu', padding='same')(d)
d = BatchNormalization()(d)
d = Conv2DTranspose(64,(3, 3), strides=2, activation='relu', padding='same')(d)
d = BatchNormalization()(d)
d = Conv2DTranspose(32,(3, 3), activation='relu', padding='same')(d)
decoded = Conv2D(3, (3, 3), activation='sigmoid', padding='same')(d)
```

Fonte: O Autor (2022).

Figura 18 – Sumário do autoencoder implementado

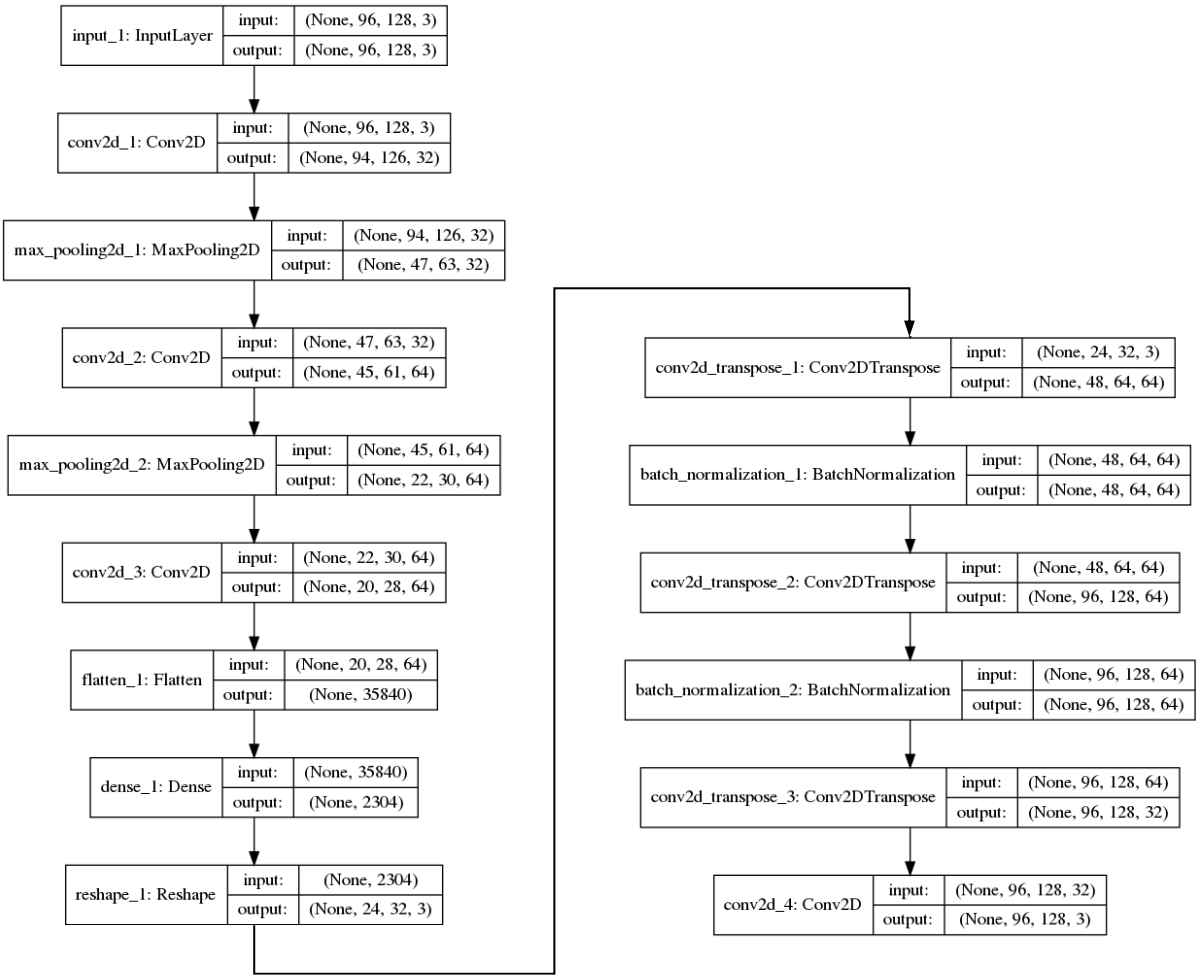
Model: "model_1"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 96, 128, 3)	0
conv2d_1 (Conv2D)	(None, 94, 126, 32)	896
max_pooling2d_1 (MaxPooling2	(None, 47, 63, 32)	0
conv2d_2 (Conv2D)	(None, 45, 61, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 22, 30, 64)	0
conv2d_3 (Conv2D)	(None, 20, 28, 64)	36928
flatten_1 (Flatten)	(None, 35840)	0
dense_1 (Dense)	(None, 2304)	82577664
reshape_1 (Reshape)	(None, 24, 32, 3)	0
conv2d_transpose_1 (Conv2DTr	(None, 48, 64, 64)	1792
batch_normalization_1 (Batch	(None, 48, 64, 64)	256
conv2d_transpose_2 (Conv2DTr	(None, 96, 128, 64)	36928
batch_normalization_2 (Batch	(None, 96, 128, 64)	256
conv2d_transpose_3 (Conv2DTr	(None, 96, 128, 32)	18464
conv2d_4 (Conv2D)	(None, 96, 128, 3)	867

Fonte: O Autor (2022).

Na Figura 19, ilustramos o diagrama em blocos do autoencoder. Trata-se, portanto, de um autoencoder com 13 camadas internas somadas às interfaces de entrada e saída.

Figura 19 – Diagrama em blocos das camadas do autoencoder implementado



Fonte: O Autor (2022).

Da Figura 19, observa-se que tanto entrada quanto saída do autoencoder são matrizes de dimensões 96 x 128 x 3. De fato, trata-se de imagens de 96 por 128 pixels em cores no padrão RGB (*Red* - vermelho*, Green* - verde e *Blue* - azul). Essas são as dimensões das imagens utilizadas em nosso trabalho. Contudo, isso pode ser alterado de acordo com o sistema (de detecção ou classificação) desenvolvido. Ou seja, a dimensão das imagens funciona como parâmetro de ajuste, que pode ser moldado de acordo com o problema ser resolvido. Outros parâmetros são considerados na Seção 3.3, onde descrevemos a geração das imagens a partir de espectrogramas.

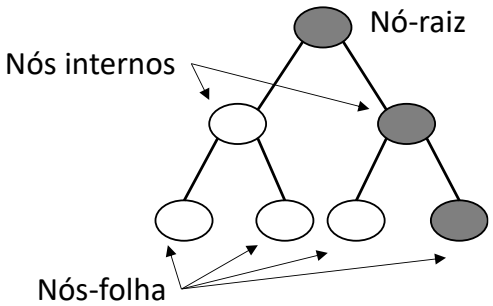
Para sua construção, a otimização utilizada foi Adam (Kingma, 2014), que é um método estocástico de gradiente descendente que se baseia na estimação adaptativa de momentos de primeira e segunda ordem. Utilizamos o erro médio quadrático (MSE) como função de perda. O MSE calcula a média dos quadrados dos erros entre as imagens reconstruídas (previsões) e as imagens de entrada (Ketkar, 2017). A métrica usada na construção do autoencoder foi acurácia (*Accuracy*). Essa métrica calcula a frequência com que a predição corresponde ao resultado correto. Essa frequência é retornada como acurácia binária: uma operação idempotente que simplesmente divide o total pela contagem (Ketkar, 2017).

Os demais detalhes do treinamento estão descritos nas respectivas seções que abordam as aplicações de autoencoder propostas neste trabalho.

3.1.6 Árvores de decisão e random forest

Árvores de decisão são modelos de aprendizado de máquina supervisionado que podem ser usados para classificação e regressão (Rokach, 2005). Ou seja, podem prever categorias discretas (OK ou NÃO-OK, por exemplo) ou prever valores numéricos. A estrutura de uma árvore de decisão é formada por nós de decisão que são relacionados entre si por meio de uma hierarquia, como ilustrado na Figura 20. Existem três tipos de nó: o nó-raiz, que fica no topo da estrutura; nós intermediários ou internos; e os nós-folha, que indicam os resultados finais. O nó-raiz corresponde a atributos dos dados de entrada, e o nó-folha informa a resposta do modelo, que pode ser a classe ou valor da regressão.

Figura 20 – Estrutura de uma árvore de decisão

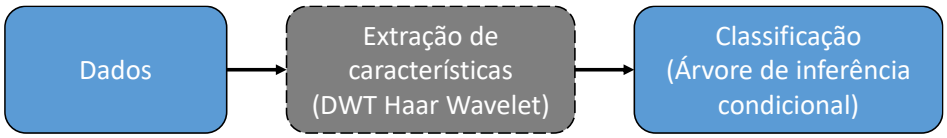


Fonte: O Autor (2022).

A criação de uma árvore de decisão requer o cálculo da entropia das classes e do ganho de informação em funções recursivas (Rokach, 2005). Em nosso trabalho, o código foi implemento na linguagem R (Cowpertwait, 2009) usando o pacote

"ctree" (Hothorn, 2015). Para melhorar o resultado do processo, a transformada *wavelet* (Saravanan, 2010; Yang, 2016) foi aplicada para extrair características das amostras, seguindo um procedimento como o ilustrado na Figura 21. Assim, o classificador lê as amostras, extrai suas características (coeficientes de detalhe e de aproximação) e as envia ao modelo treinado para classificação.

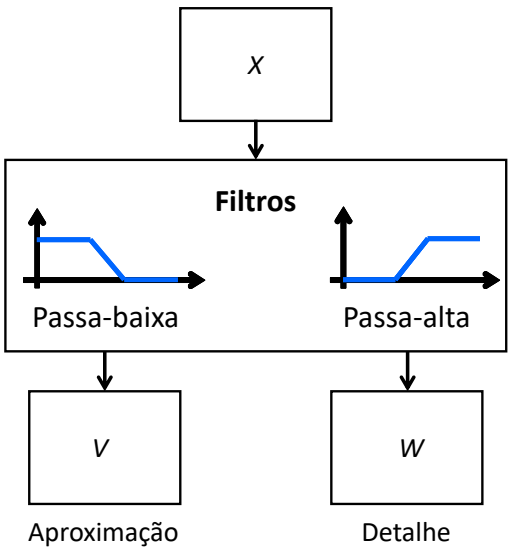
Figura 21 – *Clustering* de séries temporais com extração de características



Fonte: O Autor (2022).

Em análise de sinais com transformadas *wavelet*, utilizam-se os conceitos de coeficientes de aproximações e de detalhe. As aproximações são os componentes de alta escala (baixa frequência) do sinal. Enquanto os detalhes são os componentes de baixa escala (alta frequência). Uma maneira simples de entender o conceito é por meio do uso de banco de filtros, como mostrado na Figura 22. A implementação do modelo e da extração de características pode ser feita em R, usando o pacote "ctree" (Hothorn, 2015) para a árvore e o pacote "wavelets" (Aldrich, 2013) para análise *wavelet*. Deste último pacote, a função $dwt(X)$ retorna os coeficientes de detalhe (W) e de aproximação (V) do sinal temporal X. Estes coeficientes são utilizados como entrada da árvore de decisão.

Figura 22 – Obtenção dos coeficientes de aproximação e de detalhes da *wavelet* por meio de filtragem

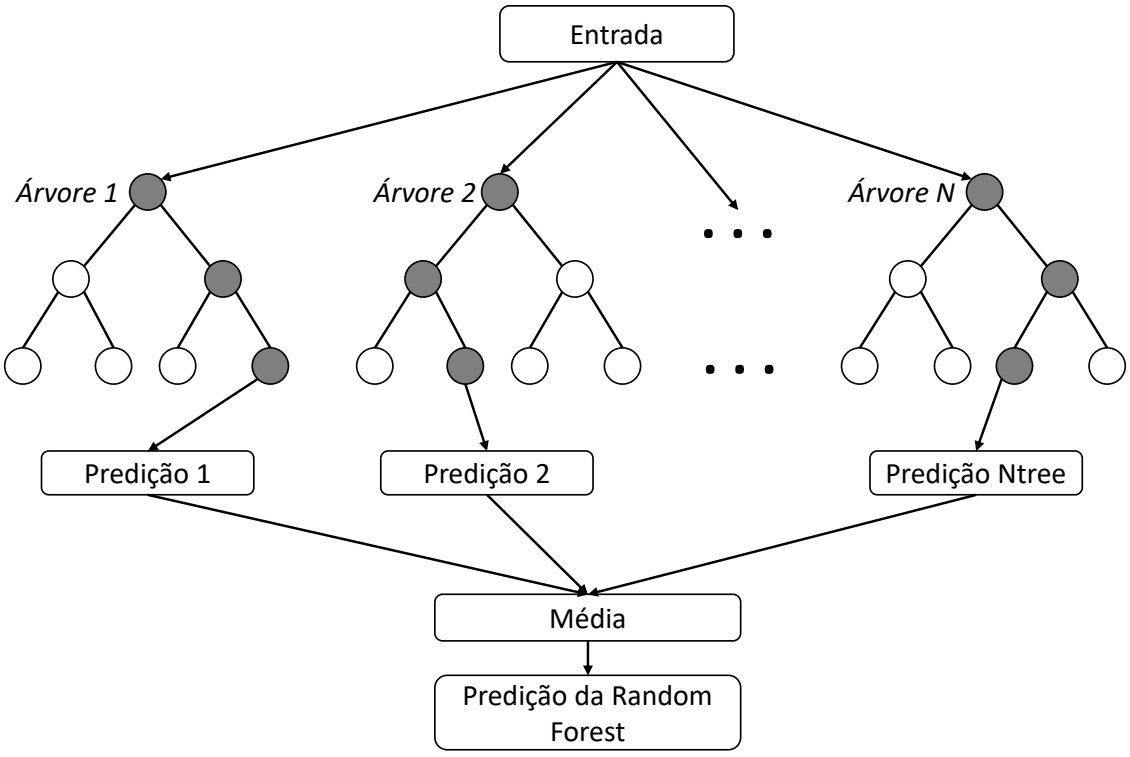


Fonte: O Autor (2022).

Árvores de Decisão são bastante eficientes em separar os dados, porém, para classificação de amostras não utilizadas no treinamento, seus resultados são limitados (Hastie et al., 2009; Seção 3.1.3). Ou seja, esses modelos apresentam um baixo viés e uma variância elevada (Geman, 1992). Então, para obter uma base de comparação com os resultados do sistema que nós propomos – baseado em autoencoders – analisamos as amostras experimentais com um modelo de *random forest* (Genuer, 2010; Rimal, 2019), que apresenta melhor generalização. Árvores de decisão foram usadas para *clustering* e demonstração da separação de classes.

A construção deste modelo usa a repetição da classificação feita no modelo árvore de decisão. Em cada repetição, um subconjunto das variáveis disponíveis ($mtry$) é considerado para construção de uma árvore. Ao final, múltiplas árvores ($Ntree$) são construídas e o resultado final de uma classificação é definido pela média das várias árvores (Figura 23).

Figura 23 – Estrutura de uma *random forest*

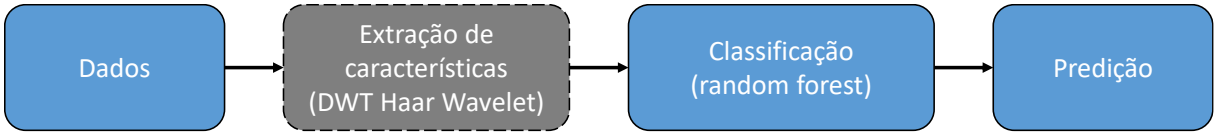


Fonte: O Autor (2022).

Em nosso trabalho, utilizamos o pacote “randomForest” do R (Breiman et al., 2018) para implementar o modelo *random forest*. O classificador lê as amostras, extrai suas características (coeficientes de detalhe e de aproximação) e as envia ao

modelo (Figura 24). Assim como na análise feita com árvores de decisão, as características (coeficientes de detalhe e de aproximação) das amostras são extraídas por meio de transformadas *wavelet* para melhorar a qualidade da classificação.

Figura 24 – Classificação de séries temporais com *random forest* e extração de características com transformada *wavelet*

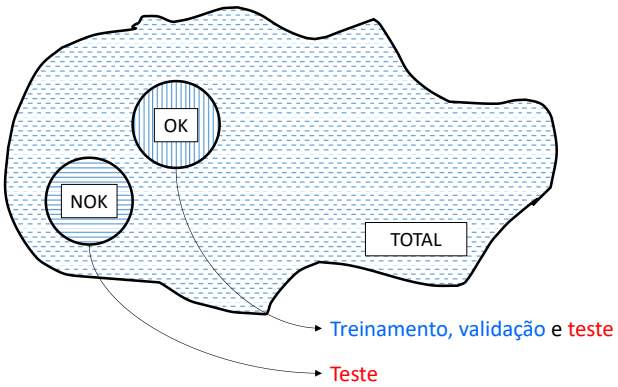


Fonte: O Autor (2022).

3.1.7 Classes dos dados

Em nossa abordagem, lidamos com dados de duas classes. Uma classe que contém os dados gerados a partir de um sistema livre de anomalias (para os sistemas de detecção de anomalias) ou os dados gerados a partir de uma substância de referência (para o sistema de classificação de substâncias). E outra classe para os dados que não pertençam à primeira. Para padronizar a escrita neste texto, denominamos a classe utilizada para treinamento/validação “classe OK”. As demais classes são denominadas “classe NOK”. Especificamente neste trabalho, consideramos os sistemas embarcados livres de anomalias e a substância de referência como sendo da classe OK. Nos experimentos, geramos uma fração de dados pertencentes a cada uma dessas classes para validar o modelo desenvolvido (Figura 25).

Figura 25 – Uma fração das amostras possíveis de dados das classes OK e NOK é gerada. O treinamento e a validação são realizados com dados da classe OK. O teste utiliza dados da classe OK e NOK

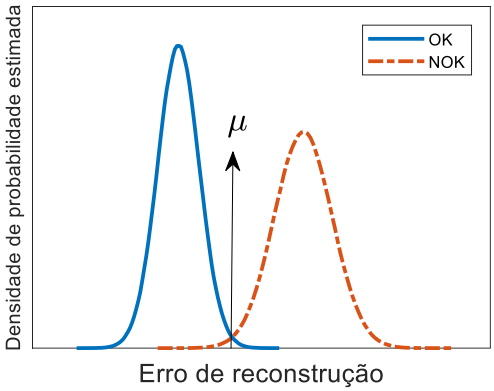


Fonte: O Autor (2022).

Observe que, durante o desenvolvimento de um modelo de aprendizado de máquina, de posse de um sistema funcional ou da substância de referência, é viável gerar os dados da classe OK. Dados da classe NOK, por outro lado, precisam ser simulados para testar o modelo. Ou seja, existe uma dificuldade intrínseca na obtenção dos dados da classe NOK. Pois, a quantidade de anomalias ou substâncias diferentes da substancia padrão é virtualmente infinita.

As curvas da Figura 26 ilustram típicas distribuições estatísticas (densidade de probabilidade estimada) para os erros de reconstrução obtidos com imagens das classes OK e NOK geradas por um autoencoder treinado com dados da classe OK. Em geral, os erros calculados a partir de dados da classe NOK são maiores que os erros obtidos com dados da classe OK. Portanto, existe um valor a partir do qual podemos considerar que o erro de reconstrução foi gerado por uma imagem da classe NOK apresentada ao autoencoder treinado. Este valor define o que chamamos de limiar de decisão, representado neste trabalho por μ.

Figura 26 – Distribuição estatística de erros de reconstrução para dados de classes OK e NOK



Fonte: O Autor (2022).

Dessa forma, definindo-se o limiar de decisão, é possível realizar detecção de anomalias e classificação (de uma única classe) por meio de autoencoders.

3.2 ALGORITMO SIFT

Com o objetivo de ter uma base de comparação para o sistema de detecção de anomalias em sistemas embarcados, implementamos um procedimento de teste usando o algoritmo de visão computacional SIFT (*Scale-Invariant Feature Transform*) (Lowe, 2004), no qual comparamos duas imagens usando um conjunto de recursos de imagem chamados pontos-chave. O algoritmo SIFT fornece esses

pontos-chave. O nível de similaridade (pontuação) entre uma imagem de teste e uma imagem de referência é definido como o número de correspondências de pontos-chave entre as duas imagens dividido pelo número de pontos-chave da imagem de teste. Dependendo do nível de similaridade, a classe da imagem testada pode ser estimada. Portanto, por meio da separação das classes, é possível detectar anomalias.

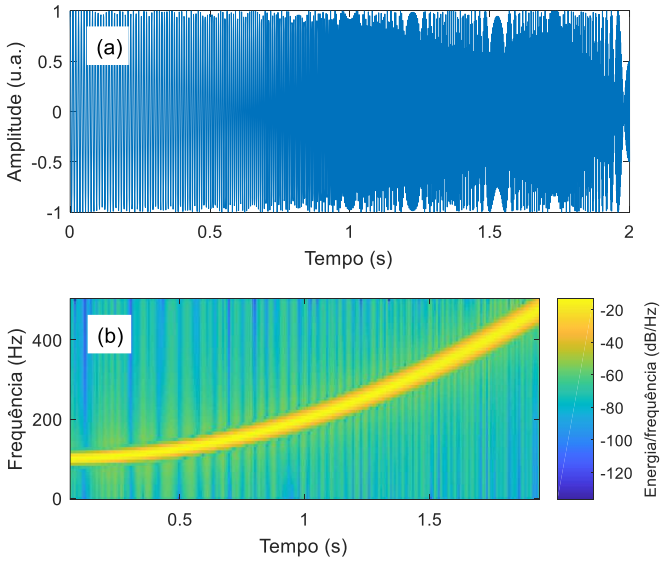
3.3 ESPECTROGRAMAS

Espectrogramas são definidos a partir da análise de Fourier em janelas curtas, também conhecida como transformada de Fourier de tempo curto (STFT) (Ojeda-Aguirre et al., 2019; Cohen, 1989). Dada a STFT F_x^g, o espectrograma S_x^g é definido pela equação (2):

$$S_x^g(t,\omega) = \left|F_x^g(t,\omega)\right|^2 \tag{2}$$

Espectrogramas são formas de representar e visualizar a evolução do conteúdo espectral de um sinal ao longo do tempo. O conceito do espectrograma pode ser melhor entendido por meio de auxílio visual. Considere um sinal $x(t)$ (Figura 27-a), cuja frequência varia de forma quadrática em função do tempo. Ao traçar o espectrograma de $x(t)$, observamos a evolução quadrática de seu conteúdo espectral, como ilustra a Figura 27-b.

Figura 27 – Espectrograma: visualização do sinal $x(t)$ e seu conteúdo espectro-temporal



Fonte: O Autor (2022).

Em nosso trabalho, utilizamos a função $spectrogram\ (x, S\_len, N\_overlap, Nf, Fs)$ do *Matlab* (Giannakopoulos, 2014) para construir os espectrogramas. A função $spectrogram()$ é utilizada da seguinte forma:

- O sinal x é amostrado com uma taxa Fs (*samples/s* ou Hz);
- O sinal x é dividido em seções de comprimento S_len por meio de uma janela de Hamming (Elliott, 2013);
- Ajustamos o valor $N\_overlap$ de amostras de sobreposição entre seções adjacentes;
- O espectro é computado em frequências $\lfloor Nf/2 + 1 \rfloor$ e intervalos de tempo $\lfloor [comprimento(x) - N\_overlap]/[Nf - N\_overlap] \rfloor$, onde "$\lfloor\ \rfloor$" denota o arredondamento para o inteiro imediatamente menor. Ou seja, maior inteiro menor que o argumento (*floor*).

A construção do espectrograma apresenta certa flexibilidade devido à possibilidade de ajuste dos seus vários parâmetros. Dessa forma, a geração de dados de treinamento e teste também é flexível. Isso representa uma liberdade de adaptação de acordo com o sistema embarcado testado ou características da substância a ser classificada. Após a obtenção do espectrograma, geramos uma imagem colorida que serve como entrada para o autoencoder. Logo, um parâmetro adicional que pode ser ajustado de acordo com a necessidade de cada projeto é o tamanho da imagem gerada. Esse ajuste deve ser refletido também no projeto do autoencoder, i. e., dimensão de entrada e saída do modelo.

3.4 TERMOGRAFIA

Fisicamente, o calor pode ser considerado uma radiação infravermelha. A faixa de comprimento de onda do infravermelho geralmente é definida entre 0,7 e 100 μm. Os raios infravermelhos são irradiados espontaneamente por todos os objetos com temperatura acima do zero absoluto (Speakman, 1998). A energia emitida está relacionada à temperatura do objeto.

Tradicionalmente, o uso de termografia em circuitos eletrônicos com uma densidade elevada de componentes é capaz de reduzir falhas precoces, i.e., falhas que ocorrem ainda no início da vida útil do dispositivo. A técnica consiste na transformação do calor emitido pela placa de circuito em imagem, que por sua vez é

obtida por um sistema capaz de detectar radiação infravermelha. Toda a informação coletada pelo sistema é utilizada na construção de um mapa de temperaturas da superfície em análise (Wagh, 2013). O método de detecção de anomalias pode ser automatizado com um *software* de análise apropriado. Sistemas de detecção da radiação se baseiam em câmeras de infravermelho capazes de detectar diferenças de temperatura tão baixas quanto 0,2 ºC. Os defeitos normalmente são detectados com o uso da técnica de subtração, i.e., o mapa de temperaturas de uma placa sob análise é comparado ao mapa de uma placa cujo funcionamento é considerado adequado (ou ideal), sendo a diferença entre ambos avaliada (Wagh, 2013).

A unidade de câmera contém um sistema óptico que varre o campo de visão em alta velocidade e foca a radiação infravermelha em um detector. A radiação capturada em uma região do circuito analisado é convertida em sinal elétrico que varia em função do tempo. Assim, em nossa abordagem, em vez de utilizar imagens da placa de circuito impresso, utilizamos sinais termográficos convertidos em espectrogramas (Seção 3.3). Os espectrogramas, por sua vez, são convertidos em imagens e aplicadas ao autoencoder. A vantagem é que o sinal termográfico contém informação comportamental (e. g., funcionamento do *software*) do circuito, além de estrutural (e. g., falta de componente ou solda defeituosa). Logo, anomalias tanto de *hardware* quanto de *software* podem ser detectadas pelo mesmo sistema.

3.5 ESPECTROSCOPIA

Quase todos os compostos com ligações covalentes absorvem várias frequências do espectro eletromagnético. Para análise química, os comprimentos de onda (λ) de interesse cobrem uma faixa de aproximadamente 2,5 e 25 μm (Pavia et al., 2014). Tradicionalmente, o número de onda ($k = 1/\lambda$) é usado na espectroscopia em vez do comprimento de onda. Assim, o alcance do infravermelho varia entre 4000 e 400 cm^{-1}. A conversão entre cm^{-1} e μm pode ser feita por meio da relação (3).

$$k(cm^{-1}) = \frac{10^4}{\lambda(\mu m)} \tag{3}$$

A banda entre 4000 e 400 cm^{-1} define a região do infravermelho conhecida como vibracional devido à natureza da absorção de energia: a energia absorvida aumenta

a amplitude de vibração de algumas ligações moleculares. A absorção depende do comprimento de onda e do tipo de ligação molecular. Portanto, é possível identificar as características de um composto por meio de sua resposta espectral. A espectroscopia se refere exatamente ao estudo dessa absorção (ou espalhamento) da radiação pela matéria.

Em relação ao equipamento mais utilizado em espectroscopia, existem dois tipos de espectrômetros: dispersivo e Fourier (FTIR: Fourier *Transform Infrared Spectroscopy*) (Pavia et al., 2014). Os mais modernos são do tipo FTIR, que gera um interferograma (Malacara, 2018). Então, usando a transformada de Fourier, o espectro de absorção é obtido. Em nosso trabalho, propomos uma abordagem que dispensa o uso do equipamento espectrômetro. Portanto, o que fazemos é uma análise baseada na resposta espectral de substâncias, entretanto, sem a necessidade de medir a absorção em todo o espectro analisado.

3.6 DISSIMILARIDADE ESTATÍSTICA

Nesta seção, damos uma visão geral a respeito de duas ferramentas usadas para inferir a distância estatística entre duas classes. Classes a partir das quais extraímos os dados usados para treinar, validar e testar o autoencoder. Em nossa abordagem, consideramos norma quadrática e a divergência de Kullback-Leibler (Kullback, 1997) por se tratarem de métricas tradicionalmente utilizadas no campo de aprendizado de máquina (Haykin, 2004; MONTEIRO, 2020a).

3.6.1 Norma quadrática média entre dois vetores

Utilizamos a norma quadrática média (NQ) para estimar a distância entre dois vetores. A NQ, calculado para duas imagens convertidas em vetores $\mathbf{x}$ e $\mathbf{y}$ de comprimento N com elementos x_i e y_i, é dado pela expressão:

$$NQ = \frac{1}{N^2}\sum_{i=1}^{N}(x_i - y_i)^2, \tag{4}$$

Em nosso trabalho, dado um conjunto de imagens, a NQ é utilizada para indicar quão distantes as imagens estão uma das outras.

3.6.2 Divergência de Kullback-Leibler

A divergência de Kullback-Leibler (divergência-KL) infere quão diferentes são duas distribuições de probabilidade. Essa dissimilaridade estatística é calculada usando uma medida de informação: entropia. Considere duas funções de densidade de probabilidade diferentes $f_x(x)$ e $g_x(x)$ de uma variável aleatória contínua x. A divergência-KL entre $f_x(x)$ e $g_x(x)$ é definida por (Haykin, 2004):

$$D_{KL}(f_x \parallel g_x) = \int_{-\infty}^{+\infty} f_x(\mathbf{x}) \log\left(\frac{f_x(x)}{g_x(x)}\right) dx. \quad (5)$$

A divergência-KL é precisamente zero quando $f_x(x) = g_x(x)$, e maior que zero caso contrário. Além disso, é invariante à permutação de componentes, escala de amplitude e transformação não linear monotônica do vetor $\mathbf{x}$ (Haykin, 2004). Também é importante saber que

$$D_{KL}(f_x \parallel g_x) \neq D_{KL}(g_x \parallel f_x). \quad (6)$$

Portanto, em nossa análise, usamos a seguinte expressão, que fornece o resultado médio das duas possibilidades:

$$D_{KL} = \frac{1}{2}[D_{KL}(f_x \parallel g_x) + D_{KL}(g_x \parallel f_x)]. \quad (7)$$

3.7 MÉTRICAS DE DESEMPENHO

Para validar as propostas apresentadas neste trabalho, precisamos formalizar uma forma de mensurar os resultados obtidos experimentalmente. Esta seção discute os parâmetros quantitativos que usamos para avaliar o desempenho geral dos sistemas de detecção de anomalias ou de classificação de substâncias.

3.7.1 Taxa de Erro

Um sistema de detecção ou classificação baseado em autoencoder compara a imagem recriada em sua saída com aquela apresentada à sua entrada. A diferença entre essas imagens define o erro de reconstrução (Err_{rec}). Se esse erro for menor

que um determinado limiar de decisão μ, a imagem apresentada ao autoencoder é considerada pertencente à classe com a qual foi treinada. Como o autoencoder é treinado com imagens da classe OK, um erro de reconstrução menor que μ indica que o sistema que gerou a imagem pertence à classe OK. Por outro lado, um erro maior que o limiar indica que a imagem pertence à classe NOK (ver descrição na Seção 0). Matematicamente, dada a distribuição de probabilidade dos erros de reconstrução de ambas as classes, a taxa de erro (de detecção ou de classificação, a depender do caso) é calculada pela equação (8):

$$Err = P(Err_{rec} > \mu | Classe = OK) \; + \; P(Err_{rec} < \mu | Classe = NOK), \tag{8}$$

onde, $P(A|B)$ denota a probabilidade do evento A, dado que o evento B ocorreu.

3.7.2 Matriz de confusão

Alternativamente, a taxa de erro pode ser calculada por meio da matriz de confusão (Ting, 2017), construída a partir dos dados experimentais. A estrutura típica para matriz de confusão usada em nosso trabalho é apresentada na Tabela 1.

Tabela 1 – Matriz de confusão usada para definir as métricas do sistema de detecção

		Classe prevista	
		OK	NOK
Classe real	OK	**TP**	FN
	NOK	FP	**TN**

Fonte: O Autor (2022).

Da matriz de confusão, podemos calcular a taxa de erro ou a acurácia com a equação (9):

$$Err = 1 - Acc = \; 1 - \frac{TP + TN}{TP + TN + FP + FN}; \tag{9}$$

onde: Acc = Acurácia; TP = Total de previsões positivas corretas; FP = Total de previsões positivas erradas; TN = Total de previsões negativas corretas; FN = Total de previsões negativas erradas. A equação (8) para a taxa de erro é mais uma definição matemática formal. Na prática, utilizamos os parâmetros da matriz de confusão e calculamos a taxa de erro por meio da equação (9).

Ainda usando os elementos da matriz de confusão, podemos definir outros parâmetros, como Taxa de Verdadeiros Positivos (TPR), Taxa de Falsos Positivos (FPR), Taxa de Falsos Negativos (FNR) e Taxa de Verdadeiros Negativos (TNR):

$$TPR = \frac{TP}{TP + FN}; \tag{10}$$

$$FPR = \frac{FP}{FP + TN}; \tag{11}$$

$$TNR = \frac{TN}{FP + TN}; \tag{12}$$

e

$$FNR = \frac{FN}{FN + TP}. \tag{13}$$

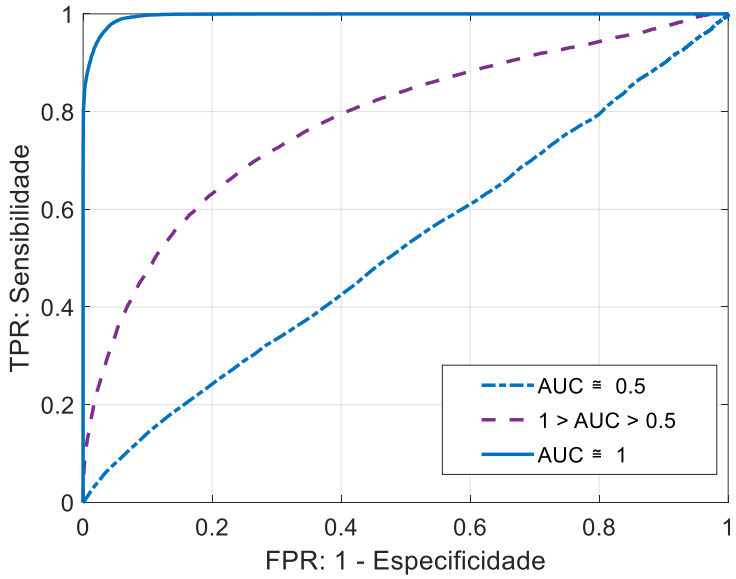
Usando as equações (10) a (13), outros parâmetros comumente empregados em análise estatística são definidos. Por exemplo, a TPR também é conhecida como Sensibilidade, enquanto $1 - FPR$ define a chamada Especificidade. Esses parâmetros são usados na construção das curvas ROC descritas na Seção 3.7.3.

Embora a acurácia e o erro denotem um indicador de desempenho direto, eles podem ocultar detalhes essenciais para o sistema de detecção de anomalias. Em alguns casos, FPR e FNR são melhores parâmetros de avaliação. Por exemplo: para sistemas de detecção de anomalias, classificar um produto defeituoso como normal implica custos mais elevados do que classificar como um produto normal como defeituoso. Pois, o produto defeituoso demandaria conserto ou substituição por parte do fornecedor, além de afetar negativamente a imagem do fabricante. Portanto, o mais indicado é um baixo FPR, ou seja, uma baixa fração de sistemas anômalos classificados como sistemas livres de anomalias. De toda forma, naturalmente, em qualquer caso, valores baixos de FPR e FNR são desejáveis.

3.7.3 Curvas ROC

Se ajustarmos o valor de μ de tal forma que FPR e TPR variem entre 0 e 1, podemos traçar a chamada curva ROC – *Receiver Operating Characteristics* – (Fan, 2006) para o modelo de aprendizado de máquina analisado. A área sob a curva ROC (AUC – *Area Under the Curve*) representa o grau ou medida de separabilidade, i. e., informa o quanto o modelo distingue as classes OK e NOK. Como exemplo, a Figura 28 mostra curvas ROC para três modelos arbitrários com desempenhos variando desde o pior caso ($AUC = 0.5$) até o caso próximo do ideal ($AUC = 1{,}0$).

Figura 28 – Exemplo fictícios de curvas ROC representando 3 modelos de desempenhos distintos



Fonte: O Autor (2022).

As curvas ROC são uma ferramenta bastante utilizada na avaliação de modelos de aprendizado de máquina (Bradley, 1997). Neste trabalho, utilizamos essa ferramenta para quantificar o desempenho dos resultados experimentais.

3.7.4 Parâmetro de desempenho

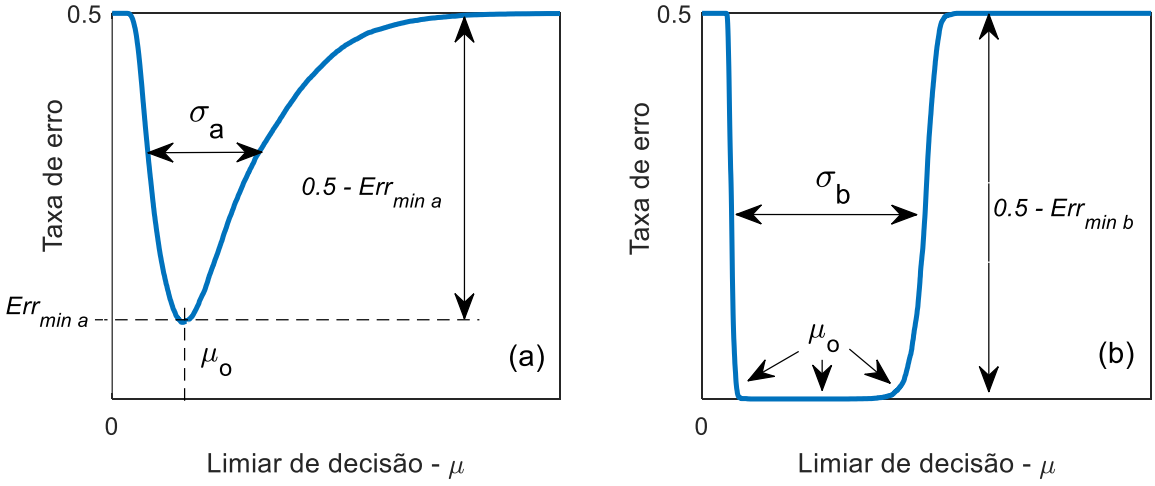
Aqui, propomos uma forma de medir o desempenho de autoencoders utilizados em sistemas de detecção de anomalias ou classificação. Definimos um parâmetro que considera a taxa de erro e o limiar de decisão do modelo a ser avaliado. Sua motivação é explicada a seguir.

O limiar de decisão ótimo (μ_o) é aquele para o qual o modelo apresenta uma taxa de erro mínima. O limiar ótimo está relacionado ao erro de reconstrução do

autoencoder (ver Figura 26). Portanto, quanto menor o erro de reconstrução para a classe OK, menor será o valor de μ_o. Para se estimar a taxa de erro, é necessário ter uma quantidade razoável de dados de ambas as classes. Em um cenário real, isso raramente é viável, pois dados da classe anômala são difíceis de se obter (Mujeeb et al., 2019). Portanto, na prática a taxa de erro é estimada e ajustada à medida que novos dados são obtidos do próprio ambiente de produção. Assim, é interessante que a taxa de erro permaneça baixa para uma ampla faixa de μ. Porque, dessa forma, o limiar de decisão pode ser definido de forma menos restritiva para sistemas reais.

Para ilustrar essa situação, considere as curvas mostradas na Figura 29. Elas representam comportamentos típicos da taxa de erro em função do limiar de detecção (ver detalhamento na Seção 3.8). A Figura 29-a ilustra o caso em que existe um ponto mínimo para a taxa de erro ($Err = Err_{min}$) em $\mu = \mu_o$. Na Figura 29-b, a taxa de erro é mínima para uma faixa de valores de μ. Isso ocorre quando o modelo tem desempenho excepcional. Neste caso, normalmente, a taxa de erro mínima é igual a zero.

Figura 29 – Curvas típicas para taxa de erro em função do limiar de decisão. (a) quando a taxa de erro mínima ocorre apenas em um ponto; (b) quando a taxa mínima ocorre em um intervalo



Fonte: O Autor (2022).

Na Figura 29, σ representa a faixa de valores de μ para os quais a taxa de erro cai abaixo de um valor específico. O parâmetro σ é semelhante ao FWHM (*Full Width at Half Maximum*), que é a diferença entre os dois valores da variável independente em que a variável dependente é igual à metade do seu valor máximo. No entanto, neste caso, a taxa de erro tem um pico mínimo em vez de um máximo.

Da Figura 29, nota-se que $\sigma_b > \sigma_a$ e $Err_{min\,b} < Err_{min\,a}$, indicando que o modelo “b” apresenta um desempenho superior. É intuitivo concluir que sistemas com maior σ, menor Err_{min} apresentam melhor desempenho. Assim, podemos definir a relação

$$Q = \frac{\sigma}{\mu_o \times (Err_{min} + \xi)} \tag{14}$$

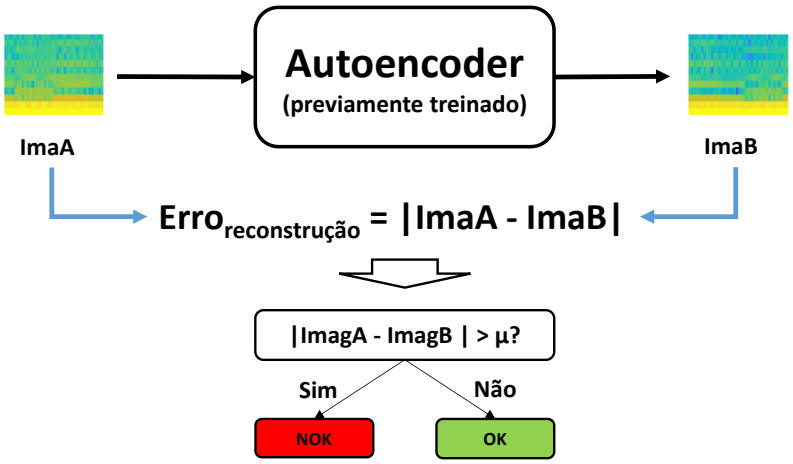
como índice de avaliação da qualidade (a letra Q faz alusão a esta palavra) de um sistema de classificação ou detecção de anomalias. Na equação (14), ξ é uma constante de valor muito pequeno para evitar que a relação tenda ao infinito nos casos em que a taxa erro mínima é nula. Neste trabalho, usamos $\xi = 10^{-10}$.

Da Figura 29 e da equação (14), concluímos que quanto maior o valor de Q, melhor o desempenho. É mais vantajoso usar esse indicador em vez da taxa mínima de erro (Err_{min}). Pois Q também considera toda a faixa de limiar de decisão para a qual a taxa de erro é aceitável. A necessidade e a versatilidade desse parâmetro ficam mais evidentes nos casos em que a taxa de erro mínima é nula para múltiplos valores de μ, como na Figura 29-b. Nesses casos, $Err_{min} = 0$, então do ponto de vista desse parâmetro os dois sistemas são equivalentes. Entretanto, aquele sistema que apresenta $Err_{min} = 0$ para uma faixa de μ maior, deve ser considerado como de melhor desempenho. Finalmente, nos casos em que existam múltiplos valores para μ_o, usamos o menor dos valores na equação (14). Exemplos desses casos são considerados nas Seções 6 e 7.

3.8 DETECÇÃO DE ANOMALIAS E CLASSIFICAÇÃO COM AUTOENCODER

O sistema de detecção de anomalias ou classificação baseado em autoencoder (Zhou, 2017) funciona conforme ilustrado na Figura 30. O autoencoder treinado reconstrói a imagem apresentada à sua entrada (ImaA).

Figura 30 – Arquitetura do sistema de detecção de anomalias com autoencoder



Fonte: O Autor (2022).

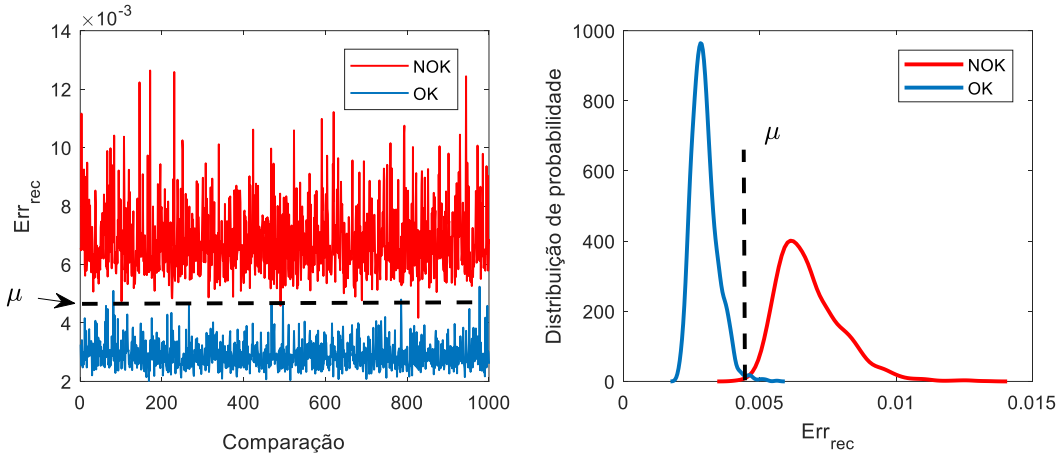
Caso a imagem de entrada pertença à mesma classe das imagens usadas para treinar o autoencoder (classe OK), a reconstrução tende a ser bem sucedida, i. e., a imagem reconstruída (ImaB) é similar à imagem de entrada. Dessa forma, espera-se que a diferença entre as duas imagens, que por sua vez define o erro de reconstrução dado pela equação (1), seja pequena. Por outro lado, quando a imagem de entrada não pertence à classe das imagens de treinamento, o erro de reconstrução tende a aumentar. Comparando-se o erro de reconstrução com um limiar μ, é possível classificar a imagem de entrada como sendo de uma classe ou de outra.

Durante o desenvolvimento de um modelo de aprendizado de máquina, seu desempenho é validado por meio da realização de múltiplas comparações do erro de reconstrução com o limiar de decisão. Tais comparações são realizadas com dados de ambas as classes. Do ponto de vista estatístico, uma validação consistente requer uma grande quantidade de comparações. Contudo, na prática, isso é limitado pela disponibilidade de dados. Sobretudo da classe NOK, pois esses dados representam todos os casos que não pertencem à classe OK.

Quantitativamente, o desempenho pode ser medido por meio de um dos parâmetros definidos na Seção 3.7. Entretanto, uma forma simples de visualizar os resultados é por meio dos erros de reconstrução (Err_{rec}) obtidos a partir das comparações feitas com dados de ambas as classes. Após a realização de uma certa quantidade de comparações, tanto Err_{rec} quanto sua distribuição estatística

podem ser traçados. A Figura 31 ilustra um exemplo arbitrário com 2000 comparações, 1000 para cada classe.

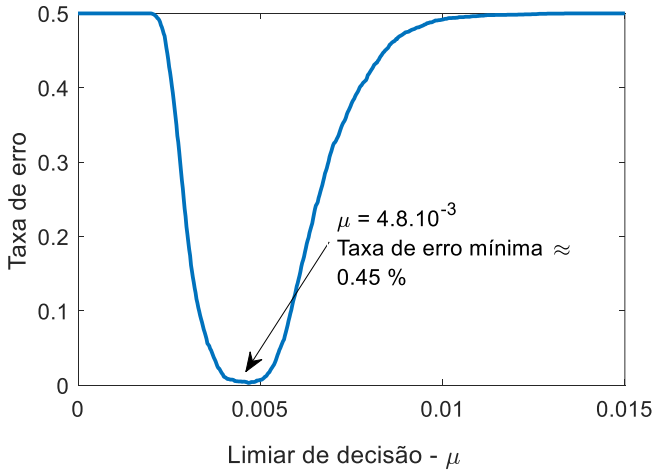
Figura 31 – Exemplos de erro de reconstrução e suas distribuições estatísticas para dados de classes OK e NOK



Fonte: O Autor (2022).

Observe que nesse caso é possível definir um limiar de decisão μ, capaz de separar as duas classes; ainda que de forma não perfeita, pois há sobreposição entre as duas distribuições. Na Figura 31, o limiar foi posicionado de tal forma que a separação entre as duas classes é máxima. Qualquer alteração no seu valor leva a uma diminuição no desempenho do modelo em detectar ou classificar dados. Isso pode ser verificado: basta calcular a taxa de erro do modelo em função de μ. O erro é calculado com a equação (9). O gráfico da taxa de erro obtida para as comparações da Figura 31 é mostrado na Figura 32.

Figura 32 – Exemplo de taxa de erro em função do limiar de decisão



Fonte: O Autor (2022).

Os resultados dos testes considerados nesse exemplo, portanto, avaliam quantitativamente o desempenho do autoencoder baseado em dados empíricos. Além da taxa de erro, outras métricas podem ser empregadas para avaliar o desempenho, conforme descrito na Seção 3.7.

Nos experimentos descritos nas Seções 4, 5, 6 e 7, a quantidade de vezes que calculamos o erro de reconstrução depende da quantidade de imagens disponíveis, ou seja, da quantidade de dados gerados. Os valores específicos de cada caso são informados na seção correspondente.

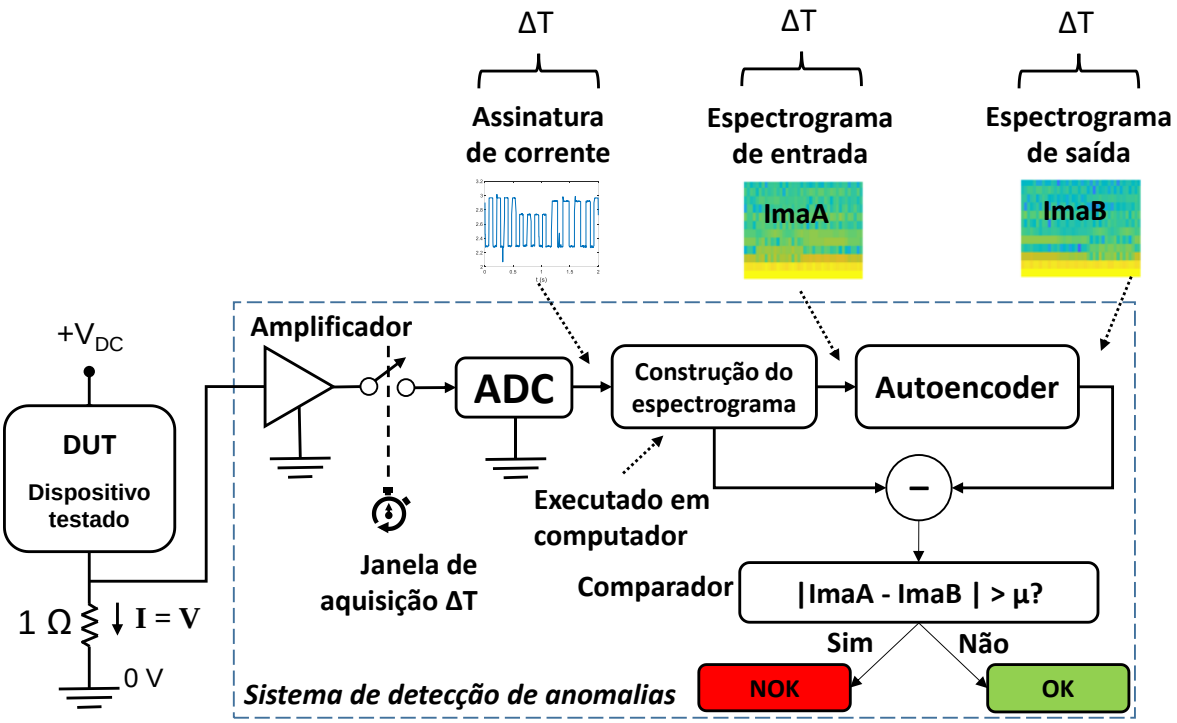
Uma simplificação que alcançamos em nossos experimentos é decorrente do fato de que as imagens utilizadas como entrada dos autoencoders em todas as abordagens (detecção de anomalias ou classificação de substâncias) possuem sempre o mesmo tamanho (96 x 128 x 3). Dessa forma, a estrutura do autoencoder não precisa ser adaptada caso a caso.

4 DETECÇÃO DE ANOMALIAS POR ASSINATURA ELÉTRICA

Nesta abordagem, a detecção de anomalias em sistema embarcado utiliza como dados a corrente elétrica consumida pelo DUT. Dessa forma, é possível obter informações comportamentais do sistema. É possível verificar se o DUT funciona de acordo com o esperado, tanto do ponto de vista do *hardware*, quanto do *software*. Além disso, a detecção ocorre de forma não invasiva, pois a medição da corrente requer apenas uma derivação para o nó de referência (terra) da própria fonte de alimentação. Neste capítulo, descrevemos o procedimento proposto para detecção de anomalias, o arranjo experimental e os resultados obtidos.

O diagrama em blocos do sistema de detecção está ilustrado na Figura 33. A corrente elétrica consumida pelo DUT é adquirida por um conversor analógico-digital (ADC) por um período de tempo ΔT pré-definido. O espectrograma do sinal adquirido é convertido em uma imagem (ImaA). O autoencoder treinado reconstrói a imagem de entrada em sua saída (ImaB). A imagem reconstruída e a imagem original são comparadas. Se a diferença entre as imagens for maior que um limiar μ, uma anomalia é detectada, conforme descrito na Seção 3.8.

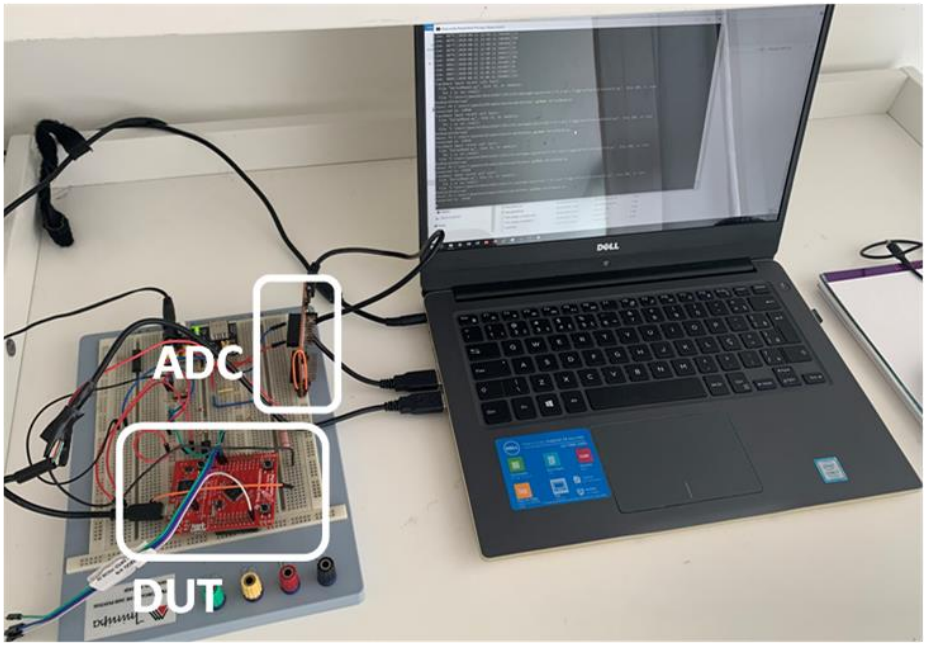
Figura 33 – Arquitetura do sistema de detecção de anomalias. Se $|ImaA - ImaB| > \mu \rightarrow$ Anomalia detectada. O resistor ($R = 1\,\Omega$) converte corrente elétrica (I) em tensão (V) pela lei de Ohm $V = RxI$; ADC – Conversor Analógico-Digital



Fonte: O Autor (2022).

A montagem experimental em laboratório mostrando o DUT e o sistema de aquisição de dados é apresentada na Figura 34.

Figura 34 – Arranjo experimental para medição e aquisição da corrente elétrica consumida pelo DUT



Fonte: O Autor (2022).

4.1 METODOLOGIA

Esta seção apresenta uma visão geral do método proposto, parâmetros-chave e geração e aquisição de dados. Aqui, são discutidos detalhes de implementação e considerações a respeito do sistema de detecção de anomalias baseados em assinatura de corrente elétrica.

4.1.1 Protótipo experimental

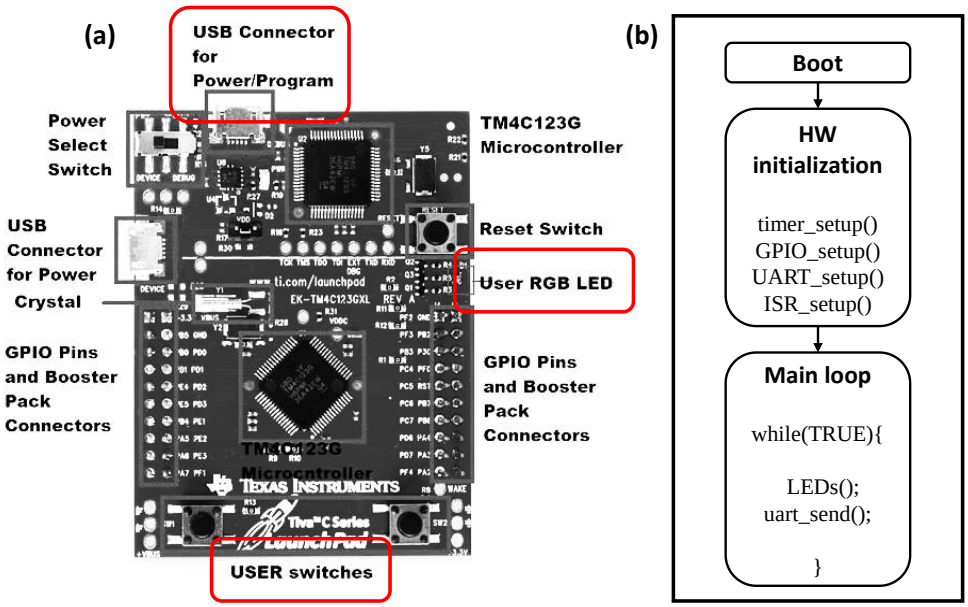
Um sistema embarcado típico (Figura 35) foi construído para validar experimentalmente os métodos, os algoritmos e as técnicas de detecção de anomalia propostos. Por sistema embarcado típico entende-se um sistema computacional implementado em um microcontrolador, que executa um *software* (nesse contexto, é comum se referir ao *software* como *firmware*). O microcontrolador, por sua vez, compõe um sistema computacional completo em forma de circuito integrado.

O DUT consiste de duas partes: o *hardware* e o *firmware*. O *hardware* foi implementado em uma plataforma de avaliação para microcontroladores baseados

em ARM Cortex-M4F chamada Tiva C (Mazidi et al., 2017). A placa de desenvolvimento contém interfaces de entrada/saída (GPIO – *General Purpose Input/Output*), que usamos para emular tarefas típicas do sistema embarcado, por exemplo, ativação de dispositivos e transferência de dados.

O *firmware* de teste que implementamos possui a estrutura tradicionalmente encontrada em tais sistemas: um bloco de inicialização e configuração de *hardware* seguido de um laço infinito (`while ( TRUE )`). Dentro do laço, portas de entrada/saída são lidas e/ou acionadas, dados são processados e informações são transferidas por meio de uma interface apropriada. Em nossa implementação específica, as atividades realizadas pelo sistema são: acionamento de LEDs multicolorido (RGB); tratamento de rotina de interrupção; transferência de dados via interface serial UART – *Universal Asynchronous Receiver/Transmitter* (Nanda, 2016). As funções `LEDs()` e `uart_send()` são responsáveis pelo funcionamento do LED e pela transmissão serial de dados, respectivamente. Essas funções emulam comportamentos típicos de um sistema embarcado reativo (LEE, 2016).

Figura 35 – Protótipo experimental desenvolvido para validar nossa pesquisa. Dispositivo em teste. (a) – Placa de desenvolvimento usada como DUT em nosso experimento; (b) – Arquitetura de *firmware* de teste



Fonte: O Autor (2022).

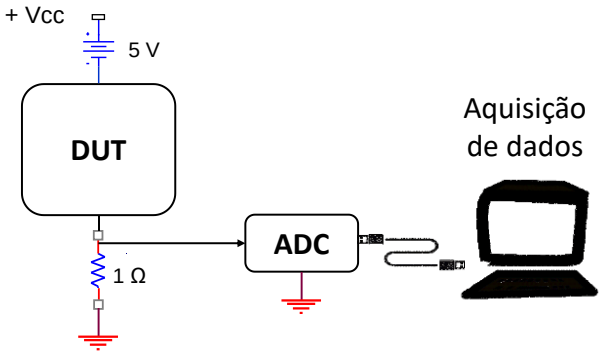
Em sistemas embarcados, o *firmware* é um código projetado e implementado para ser executado de forma determinística, geralmente com prazos em tempo real que gerenciam dados por meio de entradas, processamento, saídas e

armazenamento em várias formas. A execução ocorre geralmente de forma periódica, o que viabiliza o uso da assinatura de corrente elétrica como fonte de informação a respeito de seu comportamento para esses casos.

4.1.2 Dados experimentais

Os dados que usamos para treinar e testar o sistema de detecção de anomalias são obtidos a partir da corrente elétrica consumida pelo DUT. Assim, o primeiro passo para obter os dados é adquirir e armazenar em um computador a corrente convertida em tensão por um resistor de 1 Ω (Figura 36).

Figura 36 – Configuração de medição de corrente elétrica



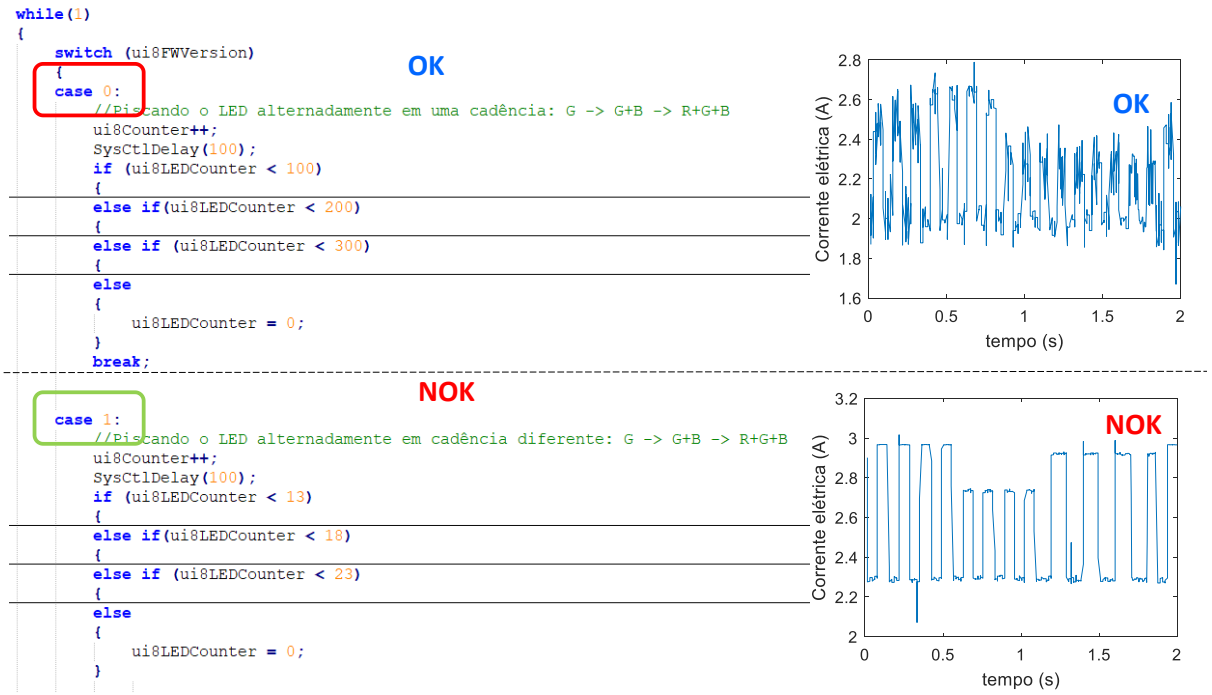
Fonte: O Autor (2022).

Neste trabalho, cada sinal adquirido é considerado uma amostra que possui uma largura (duração) denominada janela de aquisição ΔT. O número total de amostras que podemos gerar em período de tempo depende de ΔT. Esse parâmetro pode ser ajustado de acordo com as características do DUT. De fato, em nossos experimentos, estudamos a relação entre o desempenho do sistema de detecção de anomalias e a largura da janela (Seção 4.3.2). De posse dessa análise, podemos minimizar o tempo de aquisição sem prejudicar a qualidade.

No experimento, chaves de controle da placa foram utilizadas para alternar a execução entre as versões de *firmware*. O índice “OK” indica a versão padrão (sem anomalia) do *firmware*. Enquanto que os casos anômalos são indicados pelo índice “NOK”. Na Figura 37, estão ilustrados exemplos (trechos) de duas versões de *firmware* de teste utilizadas no experimento e as assinaturas das respectivas correntes consumidas, dadas em função do tempo. Cada sinal ilustrado corresponde a uma amostra. O comportamento de aparência ruidosa do sinal definido como OK na Figura 37 não está relacionado ao fato de que tenha sido adquirido de um

sistema livre de anomalias. O que define a classe a que pertence o firmware e, consequentemente o sinal gerado, é o comportamento funcional e não a forma de onda da corrente consumida ao longo do tempo.

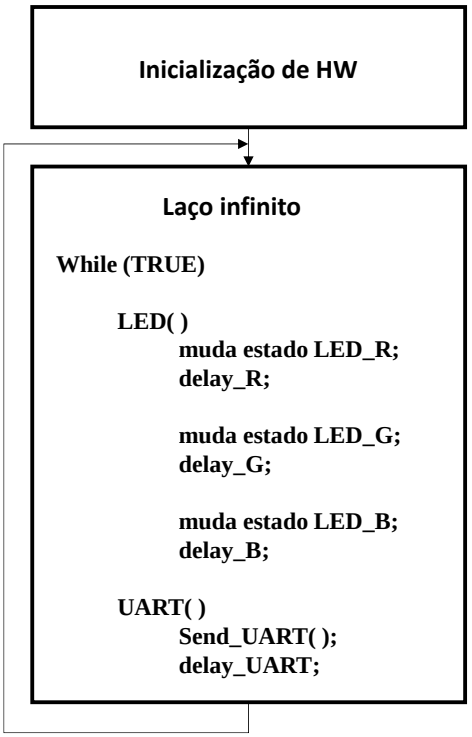
Figura 37 – Exemplo de duas versões do *software* embarcado e a assinatura da corrente consumida para cada caso



Fonte: O Autor (2022).

A estrutura geral dos *firmwares* é apresentada na Figura 38. Após a inicialização dos módulos de *hardware* do microcontrolador, o laço infinito executa as funções de ativação/desativação periódica dos LEDs e o envio de dados pela interface serial (UART). Os atrasos (*delay*) variam de acordo com a versão do *firmware* e têm como principal consequência variações no comportamento do sistema em relação ao consumo da corrente. Para simplificar a implementação do código, interrupções não são utilizadas na estrutura principal. A interrupção foi usada apenas para alternar entre as versões de firmware. Isso foi feito por meio de um botão de usuário que ativa a interrupção externa do microcontrolador. Em resposta a essa interrupção, o código executado é alterado, emulando uma nova versão de *firmware*.

Figura 38 – Estrutura dos *firmwares* utilizados para emular as versões OK e NOK do experimento



Fonte: O Autor (2022).

O modelo *random forest* utiliza as amostras temporais em sua análise. Já o autoencoder utiliza imagens geradas a partir de espectrogramas das amostras. Portanto, o próximo passo é obter os espectrogramas das amostras de duração ΔT. Em seguida, convertemos os espectrogramas em arquivos de imagem colorida de dimensões 96 x 128.

O processo de conversão de imagem nos permite limitar a faixa espectral definindo os limites do eixo da frequência (do espectrograma). Portanto, podemos filtrar ruídos e outros artefatos espectrais adicionados no processo de aquisição dos dados. A corrente medida pelo DUT contém apenas informações espectrais utilizáveis na faixa espectral inferior. Portanto, em nossos experimentos, frequências acima de 60 Hz foram cortadas. Esta filtragem pode ser ajustada para um novo DUT por meio do parâmetro de projeto f_{Range}, que define a faixa de frequências consideradas. A lista de valores dos parâmetros usados para construção dos espectrogramas nos experimentos está mostrada na Tabela 2 (Ver Seção 3.5 para definição dos parâmetros).

Tabela 2 – Parâmetros de geração do espectrograma para o caso assinatura de corrente elétrica

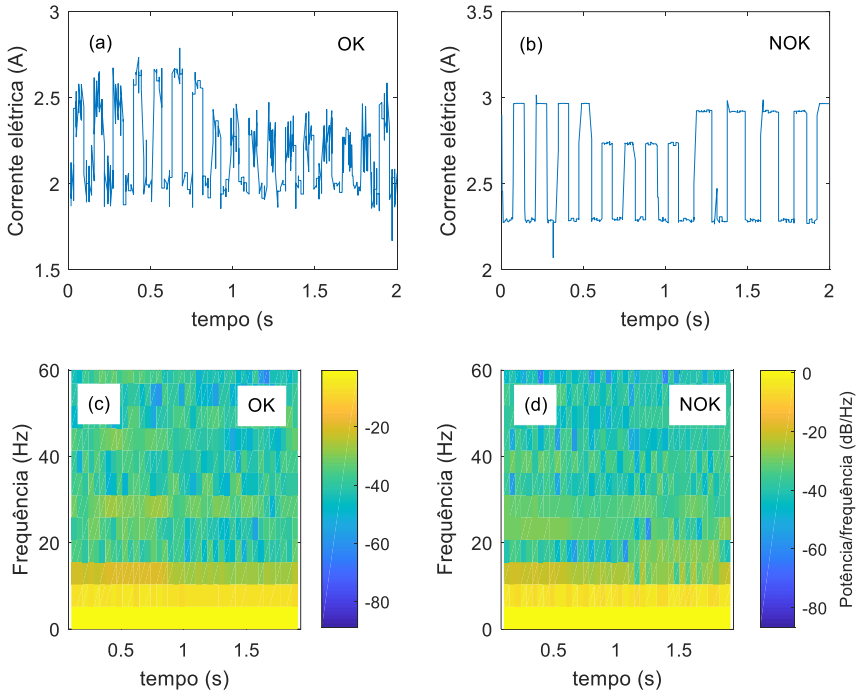
Parâmetro	Valor
S_len	64
N_overlap	50
Nf	60
Fs	310 Hz
ΔT	0,5 to 3,5 s
f_{Range}	0 a 60 Hz

Fonte: O Autor (2022).

A Figura 39 traz exemplos de amostras de sinal e espectrogramas das classes OK e NOK. Neste caso, apenas sinais com período de aquisição de dois segundos são ilustrados. Entretanto, em nosso estudo experimental outros valores também foram considerados. As imagens que são repassadas ao autoencoder não devem conter as informações dos eixos que aparecem nos espectrogramas dessa figura.

Dissimilaridades mínimas como aquelas entre os espectrogramas da Figura 39 conseguem ser detectadas pelo autoencoder treinado. Essa é a vantagem de usar um modelo de aprendizado de máquina para detectar tais diferenças.

Figura 39 – Exemplos sinais de corrente elétrica ($\Delta T = 2\ s$). (a) – OK; (b) – NOK. (c) e (d) – Exemplos de espectrogramas das janelas em (a) e (b), respectivamente



Fonte: O Autor (2022).

Nos experimentos, múltiplas versões de *firmware* foram implementadas, sendo uma considerada correta e as restantes emulam algum tipo de anomalia a ser detectada. Um sumário sobre essas versões e suas aplicações pode ser visualizado na Tabela 3. Cada versão representa uma forma de operação simulada do sistema.

Tabela 3 – Resumo das versões de *firmwares* usados nos experimentos: descrição operacional de cada versão

Designação →	***OK***	***NOK1***	***NOK2***	***NOK3***	***NOK4***	***NOK5***
Operação →	Normal (sem anomalia)	Atraso na ativação do LED	Atraso na transmissão pela UART	Ordem de ativação do LED invertida	Atraso na execução da interrupção	Simulação de defeito no LED vermelho

Fonte: O Autor (2022).

Conforme explicamos na Seção 0, apenas amostras da classe OK são usadas para o treinamento/validação do autoencoder. Amostras das classes restantes ($NOK_{1\text{-}5}$) são usadas nos testes de detecção de anomalia realizada pelo autoencoder previamente treinado.

Além de múltiplas versões de *firmware*, amostras com larguras de aquisição (ΔT) diferentes também foram geradas. O objetivo, neste caso, é analisar o desempenho em função dessa largura. A Tabela 4 contém as informações a respeito das amostras geradas e suas quantidades, assim como sua utilização no experimento (treinamento/validação $\rightarrow$ classe OK; teste $\rightarrow$ Classe $NOK_{1\text{-}5}$). Para janelas mais curtas ($\Delta T \leq 2000\, ms$), o número de amostras de treinamento/validação é 12000. Para janelas mais longas, esse valor é menor porque o tempo total de aquisição é proporcional a ΔT. Portanto, o número de amostras é reduzido para evitar aquisições com duração muito longa. De qualquer forma, como o conteúdo espectral é mais rico para janelas mais longas, o número reduzido de amostras não impacta negativamente o treinamento do autoencoder nem o desempenho da detecção de anomalias, conforme pode-se averiguar pelos resultados obtidos (Seção 0).

Tabela 4 – Denominação das amostras geradas para os experimentos e quantidade gerada (#)

Janela de aquisição - ΔT (ms)	***Designação***	**#treinamento**	**#validação**	**#teste**
500	ΔT_500	9600	2400	2000
750	ΔT_750	9600	2400	2000
1000	ΔT_1000	9600	2400	2000
1250	ΔT_1250	9600	2400	2000
1500	ΔT_1500	9600	2400	2000
1850	ΔT_1850	9600	2400	2000
1750	ΔT_1750	9600	2400	2000
2000	ΔT_2000	9600	2400	2000
2250	ΔT_2250	8811	2209	2000
2500	ΔT_2500	7767	1942	2000
2750	ΔT_2750	6770	1692	2000
3000	ΔT_3000	6202	1551	2000
3500	ΔT_3500	5080	1270	2000

Fonte: O Autor (2022).

O tempo total de aquisição do experimento foi de aproximadamente oito horas para as versões de *firmware* das classes OK e aproximadamente 75 minutos para as versões restantes (NOK1 a NOK5). Por isso, a menor quantidade de amostras dessas últimas classes.

4.2 ANÁLISE EXPLORATÓRIA

Inicialmente, para assegurar que, a partir dos dados obtidos, é viável diferenciar uma classe de dados de outra, fizemos uma análise preliminar. Para tanto, utilizamos ferramentas de análise de eficácia comprovada: árvores de decisão (Fratello, 2018); *random forest* (Fratello, 2018); transformada *wavelet* (Debnath, 2002); e SIFT (*Scale-Invariant Feature Transform*), um algoritmo de processamento de imagens digitais (Lowe, 2004).

Como se trata de uma análise preliminar, apenas uma largura de aquisição foi considerada. Especificamente, amostras com janela de aquisição de 2 segundos, ou seja $\Delta T = 2\,s$, foram usadas. A Tabela 5 lista as versões de *firmware* e as quantidades de amostras geradas.

Tabela 5 – Resumo das versões de *firmwares* usados nos experimentos: quantidade de amostras geradas; e descrição operacional de cada versão (NA - Não Aplicável)

Designação →	***OK***	***NOK1***	***NOK2***	***NOK3***	***NOK4***	***NOK5***
Operação	Normal (sem anomalia)	Atraso na ativação do LED	Atraso na transmissão pela UART	Ordem de ativação do LED invertida	Atraso na execução da interrupção	Simulação de defeito no LED vermelho
Total de amostras	14599	13909	2000	2000	2000	2000
Treinamento da Random Forest	12599	11909	0	0	0	0
Teste da Random Forest	2000	2000	0	0	0	0
SIFT	2000	2000	0	0	0	0

Fonte: O Autor (2022).

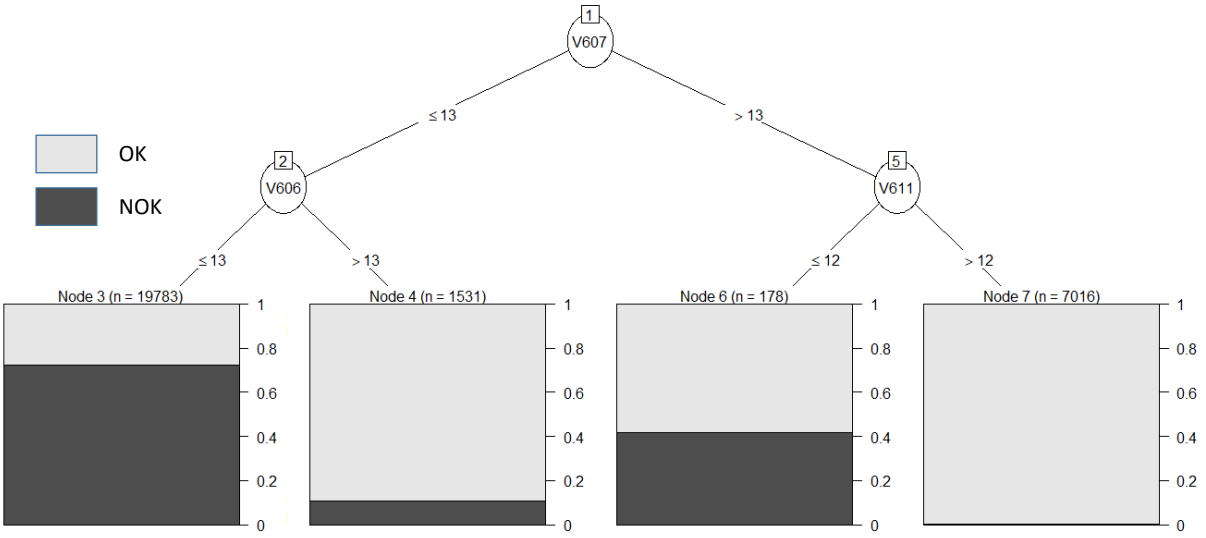
4.2.1 Análise com árvore de decisão

Utilizamos árvore de decisão (Seção 3.1.6) com o simples intuito de averiguar a viabilidade de *clustering* dos dados de classes distintas. Utilizando as amostras de corrente adquiridas das duas versões de *firmware* de teste OK e NOK1 (por uma questão de simplificação), executamos uma *clustering* hierárquica com árvore de decisão. A quantidade de amostras e sua descrição se encontram na Tabela 5.

Sem a extração de características, os resultados obtidos são muito limitados, portanto, irrelevantes para nossa análise. Então, antes de apresentar a amostra à entrada da árvore, utilizamos transformada *wavelet* para extrair as características do sinal. Com extração de características, foi possível separar (classificar) as amostras em duas classes, conforme esperado. Repetimos o procedimento para árvores com diferentes números de camadas com o objetivo de encontrar a composição que oferecesse os melhores resultados. O resultado para a versão com duas camadas está ilustrado na Figura 40. Os coeficientes de detalhe (W) e de aproximação (V) do sinal temporal X são obtidos conforme explicado na Seção 3.1.6. Os coeficientes são usados como entradas da árvore. Em nossa análise, consideramos uma decomposição com 5 níveis e *wavelet* de Haar como *wavelet* mãe. Outras famílias não foram testadas, dados os bons resultados obtidos com a *wavelet* Haar. Para uma árvore construída com 2 camadas, os coeficientes V607, V606 e V611 (a nomenclatura e a numeração são próprias do pacote “wavelets”) foram escolhidos

como as variáveis dos nós raiz e internos da árvore. Neste caso, isso significa que três coeficientes de aproximação *wavelet* do nível 5 foram utilizados.

Figura 40 – Resultado da classificação com árvore de decisão de duas camadas



Fonte: O Autor (2022).

Para este caso, a seguinte matriz de confusão foi obtida:

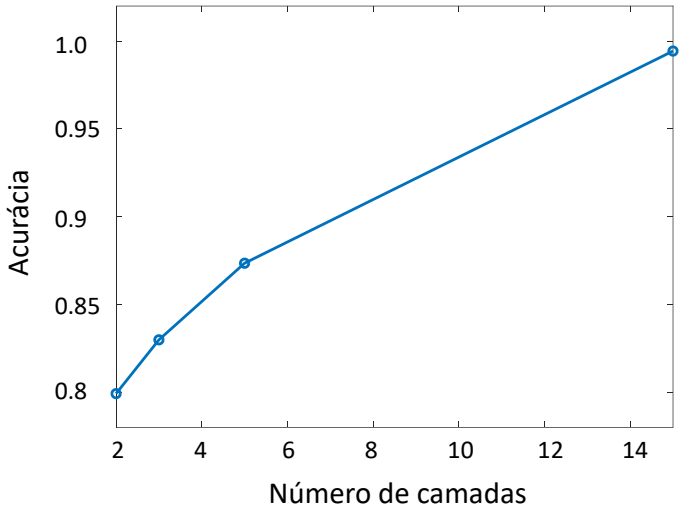
Tabela 6 – Matriz de confusão obtida com Árvore de Decisão da Figura 40

		Classe prevista	
		OK	NOK
Classe real	OK	**14340**	259
	NOK	5443	**8466**

Fonte: O Autor (2022).

Isso corresponde a uma acurácia de 80%. Árvores mais complexas tendem a escolher uma variedade maior de coeficientes W e V como as variáveis dos nós raiz e internos da árvore. Os resultados para árvores com maior número de camadas indicam uma tendência de melhoria na qualidade da classificação, conforme mostra a curva da Figura 41. Para uma árvore com 15 camadas, por exemplo, a acurácia é aproximadamente 99,44%. Árvores com mais que 15 camadas tendem a resultar em desempenhos similares a este, porém, com um custo computacional mais elevado.

Figura 41 – Acurácia em função do número de camadas da árvore de decisão

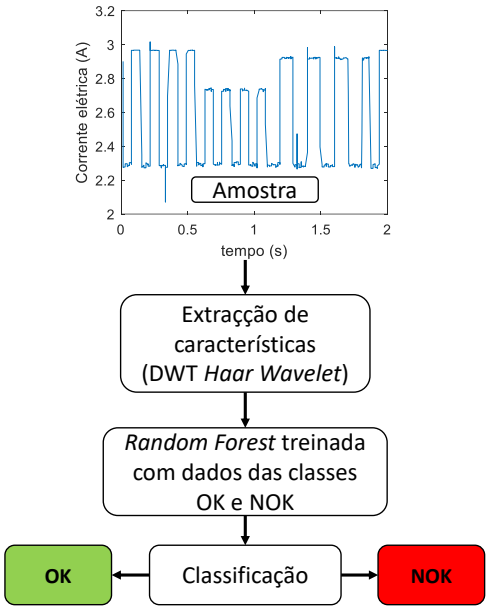


Fonte: O Autor (2022).

4.2.2 Análise com random forest

Árvores de decisão são exemplos de modelo que apresentam baixo viés e alta variância (Geman, 1992). Por isso, analisamos as amostras com um modelo *random forest* (Seção 3.1.6). Por questões de simplificação, construímos um classificador e treinamos apenas com duas versões de *firmware* da Tabela 5: OK e NOK1. Para melhorar a qualidade da classificação, as características das amostras foram previamente extraídas por meio de análise *wavelet*. Em nossa análise, consideramos uma decomposição com 5 níveis e *wavelet* de Haar como *wavelet* mãe. Outras famílias não foram testadas, dados os bons resultados obtidos com a *wavelet* Haar. Resumidamente, o classificador lê as amostras, extrai suas características (coeficientes de aproximação e de detalhe *wavelet*) e as envia ao modelo treinado para classificação. Observe que solução baseada em *random forest* processa o sinal termográfico diretamente, ou seja, a série temporal é usada como entrada da extração dos coeficientes da análise *wavelet*. Não é preciso gerar espectrogramas nem imagens. O procedimento está ilustrado na Figura 42. Neste experimento, os dados foram separados em uma razão 75% e 25% para treinamento e validação.

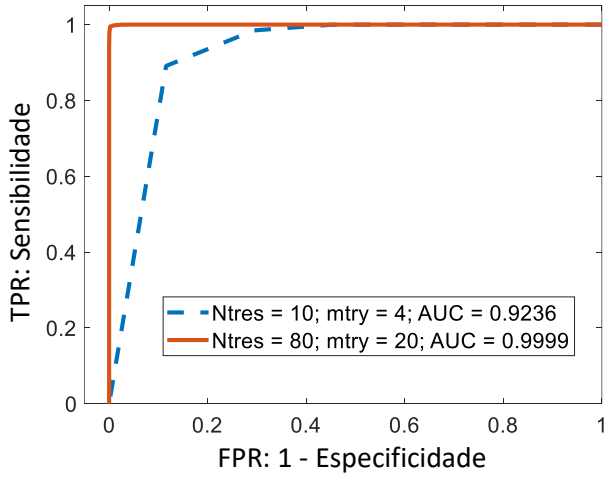
Figura 42 – Estrutura de classificação com r*andom forest* e extração de características com *wavelet*



Fonte: O Autor (2022).

É possível ajustar os parâmetros do modelo de modo a buscar melhores resultados. Então, investigamos o desempenho do modelo em função dos parâmetros $Ntree$ e $mtry$. $Ntree$ é o número de árvores usadas no modelo e $mtry$ é o número de variáveis aleatoriamente escolhidas em cada processo de separação de dados na construção do modelo. O modelo foi implementado em R e os melhores resultados foram encontrados para $Ntree = 80$ e $mtry = 20$. A curva ROC obtida está exibida na Figura 43 para dois casos testados. O erro de classificação mínimo praticamente nulo ($AUC = 0.9999$) indica que o modelo consegue realizar uma classificação quase ideal.

Figura 43 – Curvas ROC para o modelo *random forest* implementado



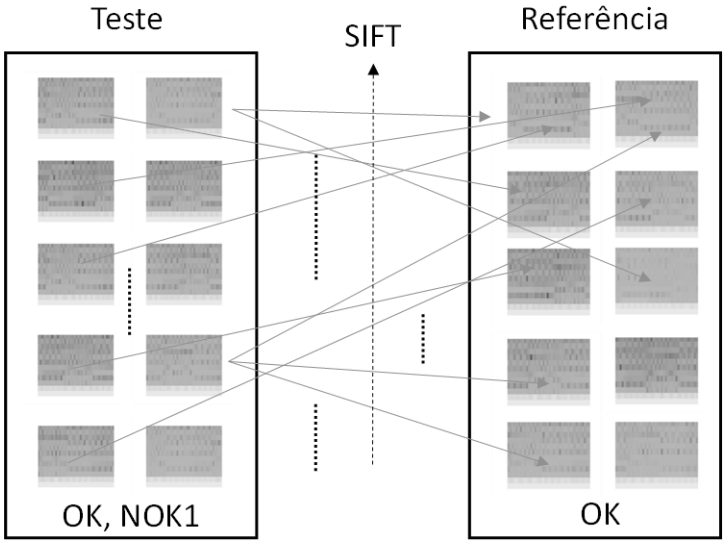
Fonte: O Autor (2022).

Os resultados dessa análise preliminar (Figura 40, Figura 41, Figura 43 e Tabela 6) indicam que é viável separar os dados (amostras das correntes) em classes distintas utilizando modelos matemáticos.

4.2.3 Análise com algoritmo SIFT

Nesta seção, descrevemos outra abordagem possível para classificação das amostras. Neste caso, entretanto, as imagens geradas a partir dos espectrogramas das amostras são utilizadas como dado de entrada do modelo de classificação. O modelo, que na verdade é um algoritmo de visão computacional denominado SIFT – *Scale-Invariant Feature Transform* – (Lowe, 2004), compara duas imagens usando um conjunto de características obtidas das próprias imagens. Essas características são chamadas de pontos-chave (*key points*). A similaridade é calculada como o número de pontos-chave equivalentes nas duas imagens dividido pelo número total de pontos-chave em uma das imagens (definida como a imagem referência). No nosso caso, comparamos 2000 imagens da classe OK e 2000 da classe NOK1 com outras 2000 imagens da classe OK (imagens de referência). Portanto, 4000 x 2000 comparações no total foram realizadas. Observe o esquema de comparação entre imagens ilustrado na Figura 44. O algoritmo funciona apenas para imagens em escala de cinza.

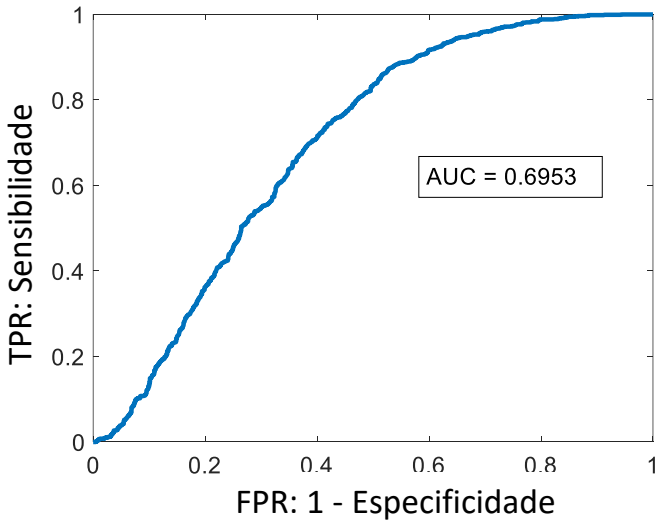
Figura 44 – Comparação entre imagens de classes distintas usando SIFT. O algoritmo SIFT só funciona para imagens em escala de cinza



Fonte: O Autor (2022).

O valor médio da similaridade obtida de cada 2000 comparações para cada classe (uma imagem de teste para as 2000 imagens de referência) foi utilizado para obter a curva ROC mostrada na Figura 45. O valor reduzido do parâmetro AUC encontrado para essa curva indica a limitação com que o SIFT realiza detecção de anomalia a partir das imagens dos espectrogramas. Em outra abordagem de detecção de anomalias em que imagens da PCI são usadas, o algoritmo apresentou melhores resultados (Mujeeb et al., 2019). Porém, no teste que realizamos, ao contrário do autoencoder (ver resultados das Seções 4.3.3 e 0), o SIFT falha, neste caso, em detectar pequenas diferenças entre as imagens. Além disso, o tempo de execução de uma comparação obtido em nosso computador foi cerca de três ordens de grandeza mais lento do que o obtido com o autoencoder (Seção 0).

Figura 45 – Curva ROC obtida a partir dos testes com o SIFT



Fonte: O Autor (2022).

4.3 RESULTADOS COM AUTOENCODER

A seguir, descrevemos a analisamos os dados obtidos experimentalmente para validar nossa proposta, que é o uso de autoencoders para detectar anomalias. Especificamente para o caso descrito neste capítulo, os seguintes resultados são relatados:

- Treinamento do autoencoder
- Custo computacional do treinamento do autoencoder
- Desempenho x Janela
- Curvas ROC e AUC

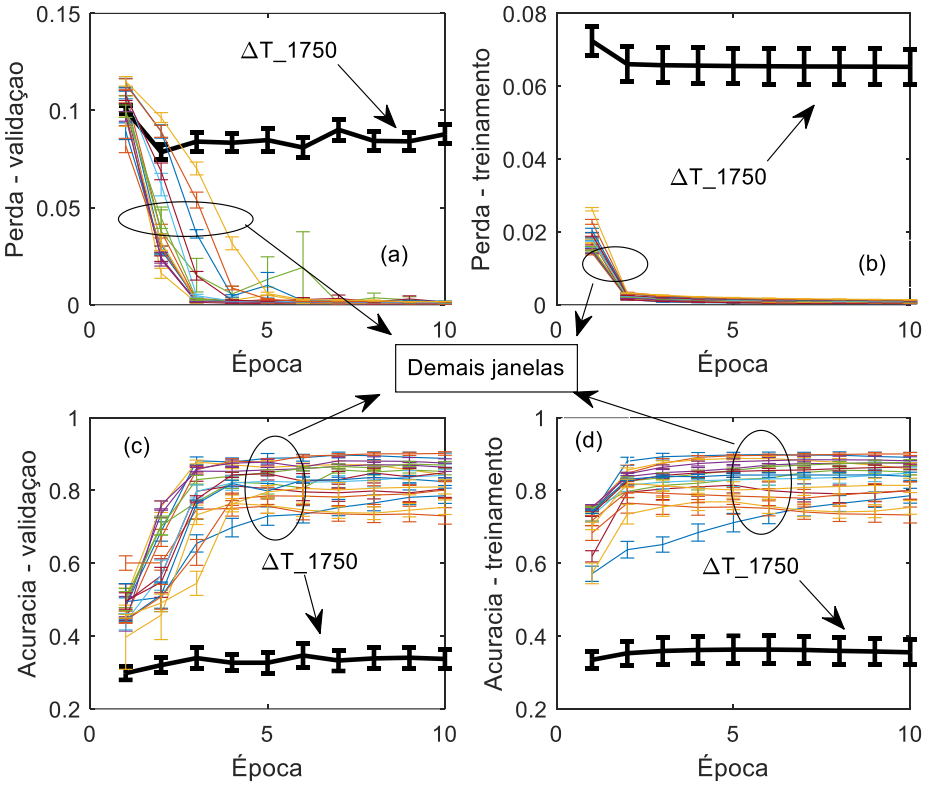
– Parâmetro de qualidade Q

4.3.1 Treinamento do autoencoder

Para cada janela da Tabela 4 e considerando as versões de *firmware* da Tabela 3, repetimos o processo de treinamento/validação 15 vezes, assim temos um resultado estatisticamente relevante. Ou seja, repetimos o treinamento de modo que a variância dos resultados, graficamente observada por meio de gráficos do tipo *boxplot* ou com barra de erro, fosse limitada. Portanto, a variância das taxas de erro e demais parâmetros é tal que, cada resultado pode ser estatisticamente comparado aos demais. Os resultados compilados são ilustrados na Figura 46 com barra de erro, indicando as variações estatísticas. O treinamento para quase todas as janelas converge. Ou seja, ao final do treinamento, o autoencoder consegue reconstruir as imagens de tal maneira que a diferença entre a imagem de entrada e a imagem de saída é próxima de zero (Figura 46-a e -b). Observe ainda que isso ocorre tanto para o treinamento quanto para a validação, ou seja, não foi observado o dilema de viés versus variância (Geman, 1992) nesses casos.

A exceção é a janela ΔT_1750 que não atingiu bons resultados: seu treinamento termina com valores de Perda e Acurácia muito limitados quando comparados aos resultados das outras janelas na mesma figura, tanto para validação quanto para treinamento. Essa janela específica é analisada em separado no Capítulo 7, onde estudamos a aplicação de transformações não lineares aos sinais temporais antes da geração das imagens espectrográficas.

Figura 46 – Validação e treinamento do autoencoder com barras de erro para todas as janelas testadas. Média para 15 testes



Fonte: O Autor (2022).

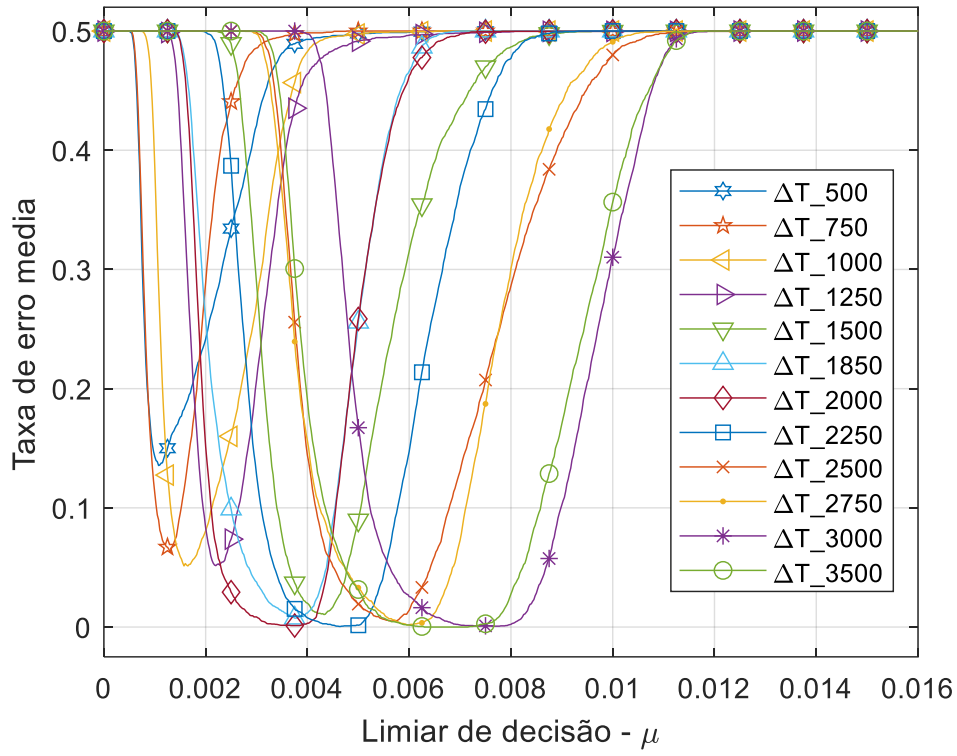
O treinamento foi realizado em um computador portátil equipado com placa de processamento gráfico (ver Seção 3.1.4). Assim, o treinamento pôde ser executado em um menor tempo. Para as janelas listadas na Tabela 4, o tempo médio foi de aproximadamente 8 minutos, ou seja, em torno de 45 segundos por época. Após o treinamento, cada teste realizado com 4000 dados das classes OK e NOK leva menos de 30 segundos para ser finalizado.

4.3.2 Desempenho

Em relação à detecção de anomalias, os resultados são apresentados a seguir. Após o treinamento, testamos a detecção de anomalias. Primeiro, definimos um valor para o limiar de detecção e obtemos os valores de cada elemento da matriz de confusão. Repetindo-se o procedimento, podemos calcular a taxa de erro em função do limiar de decisão a partir dos resultados dos testes. Ajustamos o valor de μ e calculamos as taxas de erro correspondentes, com as quais construímos as curvas mostradas na Figura 47.

O resultado para a janela ΔT_1750 não está apresentado, pois o erro médio é próximo 40% (50% é o máximo). Ou seja, a detecção de anomalias para a janela ΔT_1750 não é viável. Isso é consequência da falha em treinar o autoencoder adequadamente. Para as demais janelas, os valores mínimos da taxa de erro média ficam abaixo de 1% para a maioria dos casos. Podemos considerar, portanto, que uma taxa de erro abaixo desse valor (1%) pode ser usada como referência para aceitar o desempenho de um modelo para detecção de anomalias.

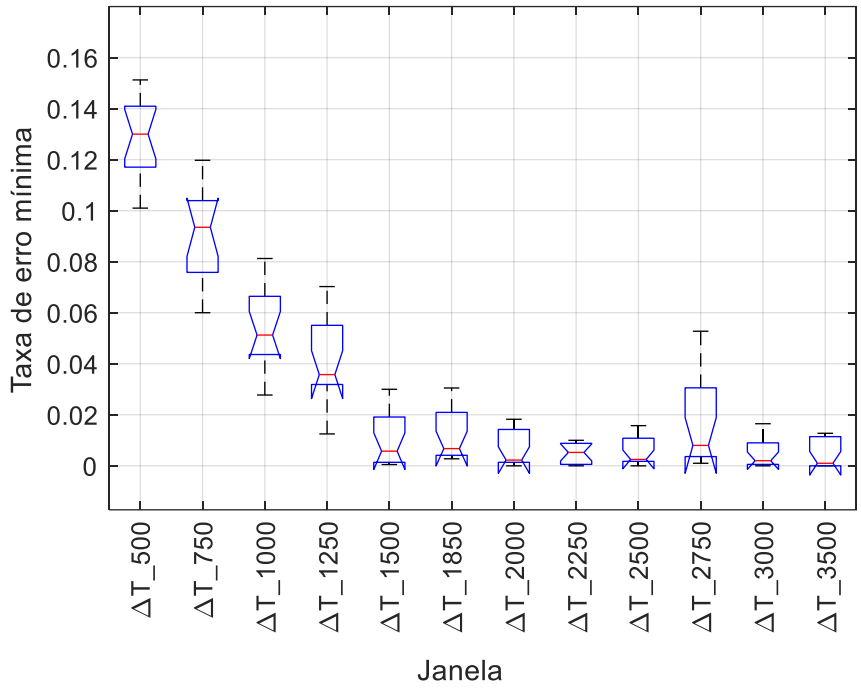
Figura 47 – Taxa média de erro de detecção em função do limiar de decisão para cada uma das janelas indicadas – média de 15 experimentos



Fonte: O Autor (2022).

Se traçarmos as taxas médias de erro mínimo em função de ΔT para os 15 casos de testes (Figura 48), observamos que o desempenho aumenta com a largura da janela. Isso é esperado, pois, quanto mais larga a janela, maior o conteúdo espectral da amostra. Consequentemente, mais fácil para o autoencoder diferenciar imagens de classes distintas. Da Figura 48, o erro mínimo tende a zero para janelas maiores que 1500 ms. Este resultado é importante porque, em aplicações práticas indica a menor largura de janela de aquisição com a qual o desempenho é ótimo. Assim, economiza-se tempo, tanto na implantação (coleta de dados e treinamento) quanto no processo de testagem.

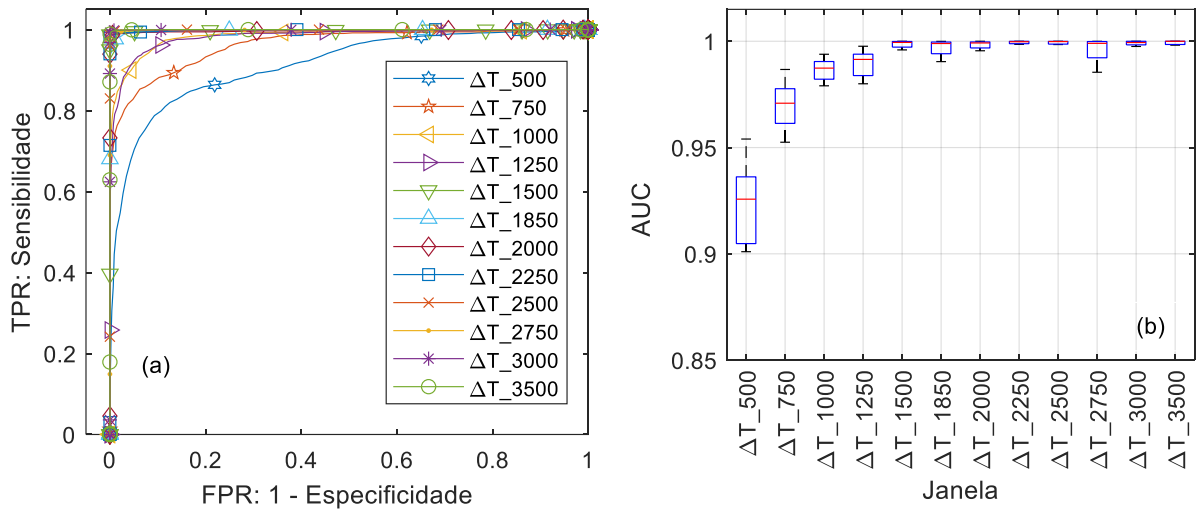
Figura 48 – Taxa de erro de detecção mínima em função da largura da janela de aquisição. *Boxplot* para 15 testes



Fonte: O Autor (2022).

O resultado médio das curvas ROC e de suas respectivas AUC está exibido na Figura 49. Esses resultados corroboram o que foi obtido por meio da taxa mínima de erro média. A partir da janela ΔT_1500, a AUC tende à unidade, que é seu valor ótimo.

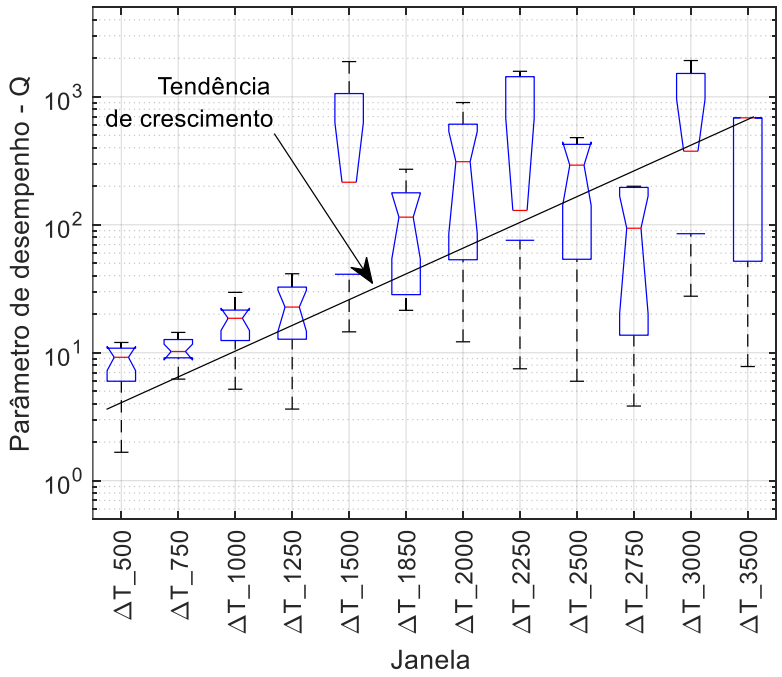
Figura 49 – (a) Curvas ROC; (b) – AUC para cada janela. Resultados médios para 15 testes



Fonte: O Autor (2022).

Também podemos fazer a análise do desempenho utilizando o parâmetro de medição de desempenho proposto na Seção 3.7.4. A Figura 50 mostra os resultados médios obtidos a partir dos 15 testes realizados.

Figura 50 – Parâmetro de qualidade Q em função da largura da janela de aquisição



Fonte: O Autor (2022).

Esse resultado mostra diferenças entre as janelas acima de 1500 ms, diferentemente do que ocorre quando consideramos apenas a taxa de erro (Figura 48) ou AUC (Figura 49). Nota-se que há uma tendência de melhoria do desempenho à medida que a largura da janela aumenta. Isso é esperado, já que amostras de janelas maiores apresentam conteúdo espectral mais rico. Consequentemente, seus espectrogramas carregam mais informações. Isso facilita o processo de separação das classes por parte do autoencoder.

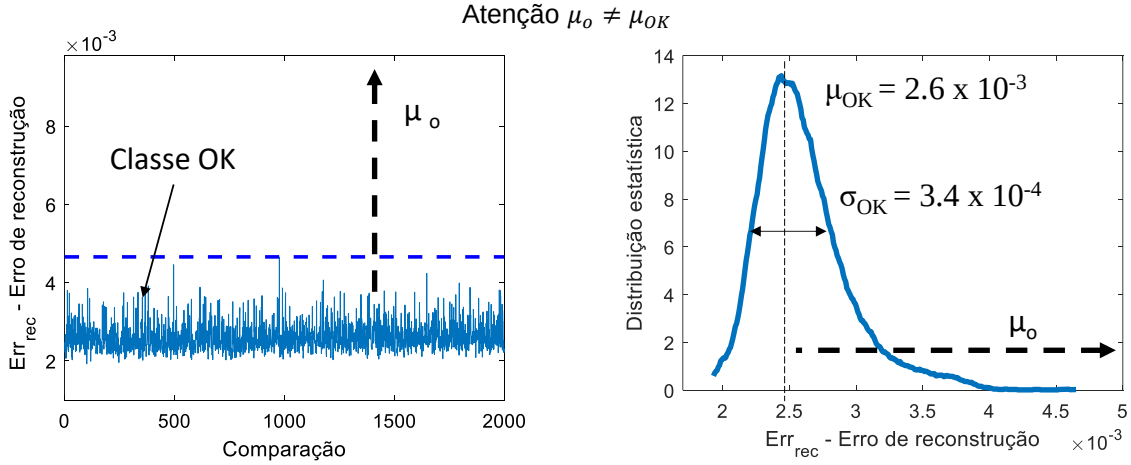
4.3.3 Estimativa do limiar de decisão

Quando utilizamos autoencoders para detectar anomalias, a taxa de erro da detecção depende do limiar de decisão μ, conforme descrito na Seção 0 e indicado pela Figura 47. Em aplicações práticas de testes de sistemas embarcados, contudo, não há como determinar o valor ótimo de μ sem a disponibilidade de amostras de todas as classes de anomalia possíveis. Em ambiente reais, a definição do limiar de decisão é um processo gradativo. Os valores são refinados à medida que novos

casos de anomalia são determinados e dados empíricos são coletados para treinamento do autoencoder (Dai et al., 2020). Aqui, apresentamos um procedimento para estimar o valor do limiar de decisão usando apenas os dados relativos aos sistemas livres de anomalia, i. e., da classe OK.

Partimos da seguinte premissa: para o sistema de detecção funcionar com alta taxa de acerto, a variância do erro de reconstrução para dados da classe OK precisa ser limitada. Dessa forma, existe uma separação nítida e bem definida entre as distribuições dos erros obtidos com dados de duas classes distintas (OK e NOK). Isso pode ser confirmado observando-se o posicionamento do limiar de decisão na curva da Figura 26. Contudo, sem os dados da classe anômala, é impossível apontar essa separação. Graficamente, essa situação está representada na Figura 51: qual a posição do limiar ótimo de decisão (μ_o)?

Figura 51 – Representação hipotética do erro de reconstrução para o caso não anômalo – classe OK. Atenção $\mu_o \neq \mu_{OK}$



Fonte: O Autor (2022).

Para contornar a falta de dados da classe anômala, parametrizamos o limiar μ em função do valor médio (μ_{OK}) e do desvio padrão (σ_{OK}) do erro de reconstrução para dados da classe OK:

$$\mu = \mu_{OK} + \gamma \times \sigma_{OK}. \quad (15)$$

Na equação (15) γ é um parâmetro de ajuste, cujo valor é positivo. Para apresentar um exemplo visual, utilizamos a equação (15) e os dados OK e NOK1 da Tabela 7 para traçar a taxa de acerto da detecção em função do limiar μ.

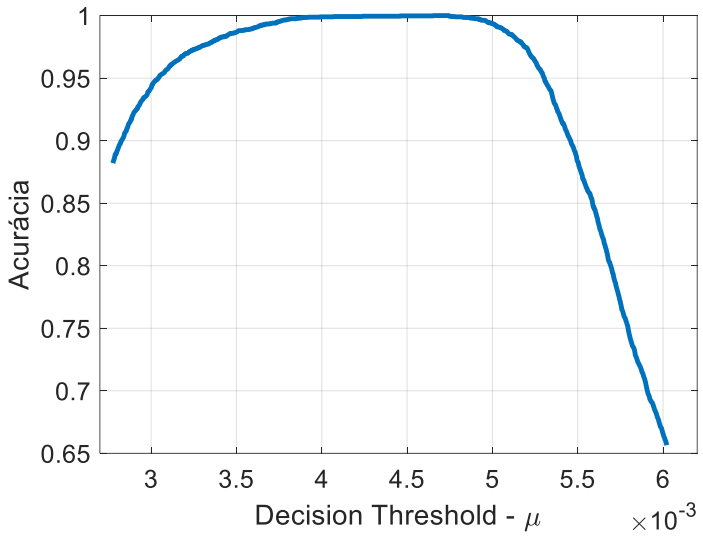
Tabela 7 – Resumo das versões de firmwares usados para estimativa do limiar de decisão: quantidade de amostras geradas; e descrição operacional de cada versão (NA - Não Aplicável)

Designação →	***OK***	***NOK1***	***NOK2***	***NOK3***	***NOK4***	***NOK5***
Operação	Normal (sem anomalia)	Atraso na ativação do LED	Atraso na transmissão pela UART	Ordem de ativação do LED invertida	Atraso na execução da interrupção	Simulação de defeito no LED vermelho
Total de amostras	14599	13909	2000	2000	2000	2000
Treinamento do autoencoder	12599	NA	NA	NA	NA	NA
Teste do autoencoder	2000	2000	2000	2000	2000	2000

Fonte: O Autor (2022).

Para traçar curva da Figura 52, o valor de γ é alterado e μ é obtido a partir da equação (15). Da curva, é possível observar que há uma faixa de valores de μ que maximizam a taxa de acerto.

Figura 52 – Ajuste do limiar de decisão por meio do fator γ. Neste caso: $\sigma_{OK} = 3{,}4\ x\ 10^{-4}$ e $\mu_o = 2{,}6\ x\ 10^{-3}$. Obtidos com dados NOK e NOK1

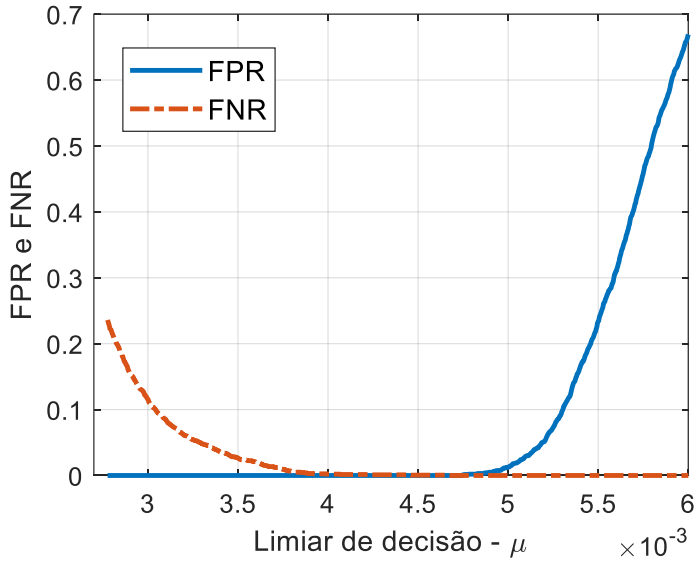


Fonte: O Autor (2022).

Para Sistemas de detecção de anomalia, os parâmetros Taxa de Falsos Positivos (FPR) e Taxa de Falsos Negativos (FNR) são mais adequados do que a taxa de acerto (acurácia). Para analisar os resultados de acordo com esses índices, na Figura 53, estão apresentadas as curvas de FPR e FNR. Para valores de FPR mais baixos, há um aumento de FNR e vice-versa, pois, são variáveis conflitantes.

Para detecção de anomalia, é preferível um sistema com valores mais baixos de FPR do que o contrário (ver Seção 3.7.2). Assim, o sistema é mais robusto, pois, classifica produtos anômalos como se fossem funcionais com menor probabilidade. Observando as curvas na Figura 52 e na Figura 53, podemos escolher um valor para μ que leve a baixos valores de FPR e, ao mesmo tempo, limita FNR. Uma boa escolha seria, por exemplo, $\mu = 4 \text{ x } 10^{-3}$ (para $\gamma = 4$). Assim, a partir da Figura 53, conclui-se que FNR é menor que 0,3%, enquanto FPR é praticamente 0%. Valores de μ entre $3{,}5 \text{ x } 10^{-3}$ e $5 \text{ x } 10^{-3}$ quase não afetam FPR e FNR, que são nulos ou praticamente nulos nessa faixa.

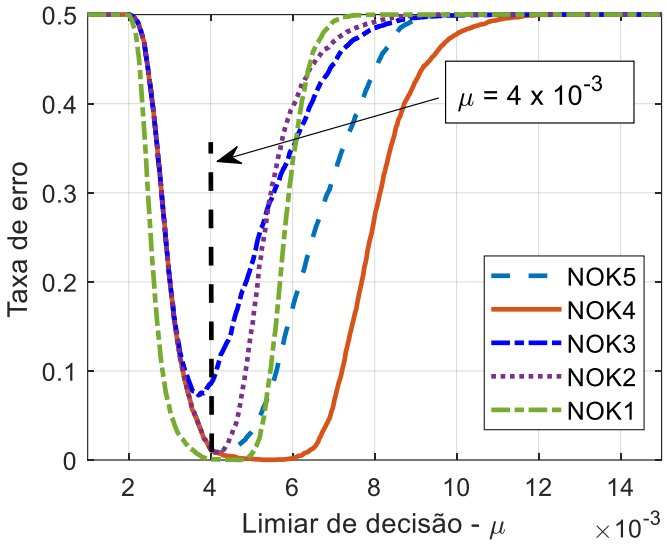
Figura 53 – Taxas de falsos positivos FPR e falsos negativos FNR. No caso de detecção de anomalias, é preferível um FPR mais baixo. Obtidos com dados NOK e NOK1



Fonte: O Autor (2022).

Usando o valor escolhido ($\mu = 4 \text{ x } 10^{-3}$) e os dados das janelas restantes da Tabela 7, podemos testar o uso desta técnica. Primeiramente, considere o comportamento das taxas de erro obtidas em função do limiar de decisão para cada classe disponível, que estão traçadas na Figura 54.

Figura 54 – Taxa de erro de detecção: (a) obtida a partir da comparação dos dados OK com os dados NOK $_{[1-5]}$; $\gamma = 4, \sigma_{OK} = 3{,}4 \times 10^{-4}$ *e* $\mu_{OK} = 2{,}6 \times 10^{-3} \rightarrow \mu_o = 4 \times 10^{-3}$



Fonte: O Autor (2022).

Nas curvas, para $\mu = 4 \times 10^{-3}$, estimado a partir dos dados da classe OK (μ_{OK} *e* σ_{OK}), o modelo apresenta baixa taxa de erro de detecção para todos os conjuntos de dados anômalos. Mesmo no pior caso da figura (amostras NOK3), a taxa de acerto é de aproximadamente 91,5%. Ressaltamos que, em um cenário real, os dados NOK não estariam disponíveis, portanto, as informações da Figura 54 não poderiam ser utilizadas. De qualquer forma, os resultados servem para ilustrar a aplicabilidade da técnica.

Os dados apresentados na Figura 54 foram obtidos a partir de testes separados de cada janela NOK apenas para simplificar a visualização. No entanto, todos os dados anômalos podem ser testados de maneira unificada. O resultado para do teste com os dados NOK unificados pode ser resumido pela matriz de confusão da Tabela 8. A taxa total de acerto para os 4000 (quatro mil) casos combinados testados é de 97,47%, sendo $FPR = 1{,}5\%$ e $FNR = 1{,}025\%$.

Tabela 8 – Matriz de confusão para 4000 testes combinados $\mu = \mu_{OK} + \gamma.\sigma_{OK} = 4\ x\ 10^{-3}$

		Classe prevista	
		OK	NOK $_{(1-5)}$
Classe real	OK	**1959**	41
	NOK $_{(1-5)}$	60	**1940**

Fonte: O Autor (2022).

Os resultados indicam uma FPR maior que a FNR, o que não é desejável. No entanto, isso reflete o fato de termos utilizado um limiar de decisão não ótimo. Para obter melhores resultados, um valor menor para γ deveria ser usado na equação (15). De toda sorte, na prática, o limiar de decisão deve ser ajustado empiricamente, logo esse tipo de imprecisão é esperado. Ainda assim, a aplicação da equação (15) aos dados experimentais produziu resultados promissores, ou seja, com baixa taxa de erro.

4.4 SÍNTESE DO CAPÍTULO

Neste capítulo, investigamos a aplicação de aprendizado de máquina e aquisição da corrente elétrica consumida para detectar anomalias em sistemas embarcados. Projetamos e construímos um DUT para validação experimental, a partir do qual a assinatura elétrica é adquirida.

Em uma análise preliminar, a viabilidade de separação das amostras em classes foi testada com um modelo de árvore de decisão. Para fins de comparação, um modelo de aprendizado de máquina baseado em *random forest* foi usado para detectar anomalias na mesma placa que usamos para validar a abordagem autoencoder. Também com a finalidade de comparar os resultados do autoencoder, utilizamos um algoritmo de processamento de imagens (SIFT) com o intuito de detectar anomalias no mesmo sistema.

As taxas de erro de detecção obtidas experimentalmente e apresentadas nas Seções 4.2 e 0 evidenciam a capacidade do autoencoder e do modelo de *random forest* de distinguir espectrogramas de diferentes classes de *firmware*. Autoencoders, contudo, apresentam a vantagem de poderem ser treinados apenas com os dados de sistemas não anômalos. O uso de uma abordagem algorítmica simples (SIFT), por outro lado, é muito limitado.

Considerando a dependência do desempenho do modelo em relação à duração de aquisição das amostras, a janela ΔT_1750 se destaca negativamente. Para essas amostras, não foi possível treinar adequadamente o autoencoder. Consequentemente, sua aplicação como detector de anomalias é inviável. No Capítulo 7, onde propomos o uso de transformações para melhorar o desempenho, tratamos especificamente dessa janela. Para as demais janelas, os resultados

demonstram a eficácia do método de teste de sistemas embarcados proposto. Para as seis versões de *firmware* testadas, a taxa de erro da detecção é significativamente baixa. Para diferenciar o desempenho de janelas que apresentam taxas de erro similares (por exemplo, janelas a partir de 1500 ms), propomos um parâmetro quantitativo Q que se baseia não apenas na taxa de erro, mas também na facilidade em se definir o limiar de decisão ótimo. Os resultados indicam uma melhoria do desempenho (aumento de Q) à medida que a duração da janela aumenta.

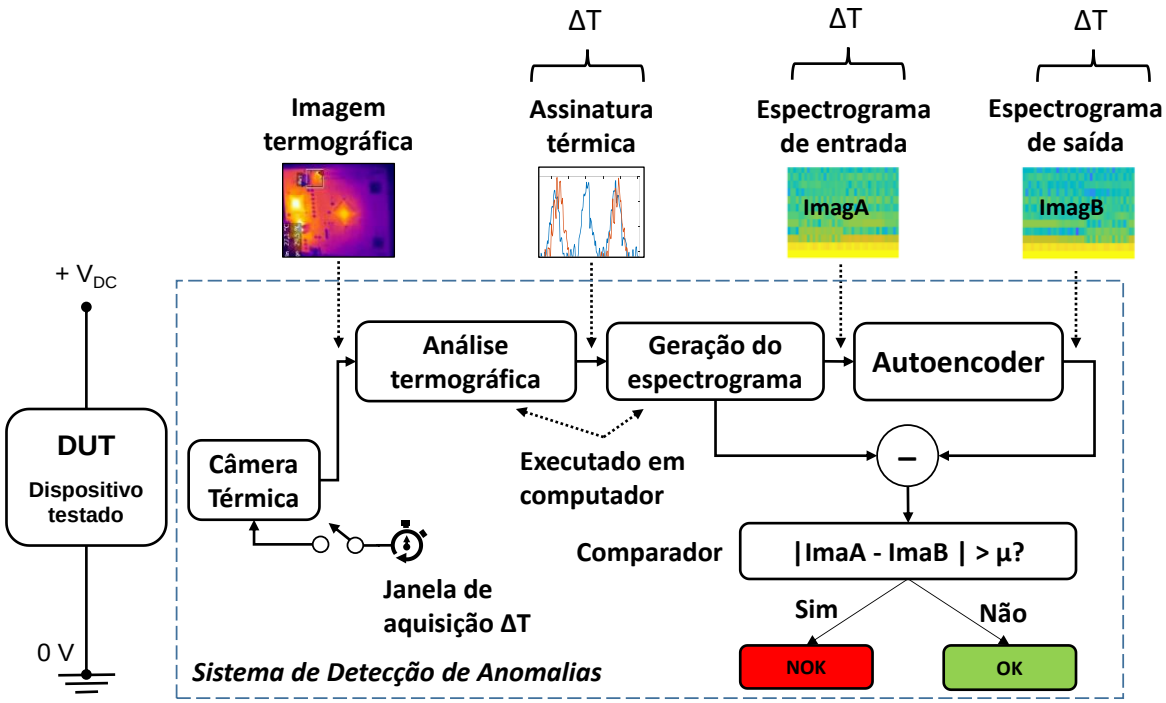
Em relação à definição do limiar de decisão, propomos uma forma de estimativa baseada no comportamento autoencoder, considerando apenas dados da classe OK. De posse apenas dos dados da classe de treinamento, ou seja, OK, usamos a equação (15) para obter uma estimativa (não arbitrária) para μ. Embora o valor não seja ideal, ele pode ser usado inicialmente enquanto os dados empíricos são coletados.

Finalmente, ressaltamos que a detecção de anomalias da forma proposta é realizada com o mínimo de interferência do sistema. Sendo assim, não há limitações relacionadas às especificações físicas do sistema, como tamanho e potência de componentes ou da PCI. O método proposto pode ser facilmente adaptado a outros sistemas embarcados, treinando o autoencoder com espectrogramas gerados a partir de sua assinatura de corrente elétrica.

5 DETECÇÃO DE ANOMALIA POR ASSINATURA TERMOGRÁFICA

Neste capítulo, apresentamos uma abordagem alternativa para a detecção não invasiva de anomalias em sistemas embarcados. Nesta abordagem, o modelo de detecção utiliza como dados a assinatura termográfica de uma região da placa de circuito impresso do DUT. Por meio dessa assinatura, é possível obter informações comportamentais do sistema. Ou seja, é possível verificar se o DUT funciona de acordo com o esperado, tanto do ponto de vista do *hardware*, quanto do *software*. Neste capítulo, descrevemos o procedimento proposto para detecção de anomalias, o arranjo experimental e os resultados obtidos. O diagrama em blocos do sistema de detecção está ilustrado Figura 55.

Figura 55 – Arquitetura do sistema de detecção de anomalias. Se $|ImaA - ImaB| > \mu \rightarrow$ Anomalia detectada

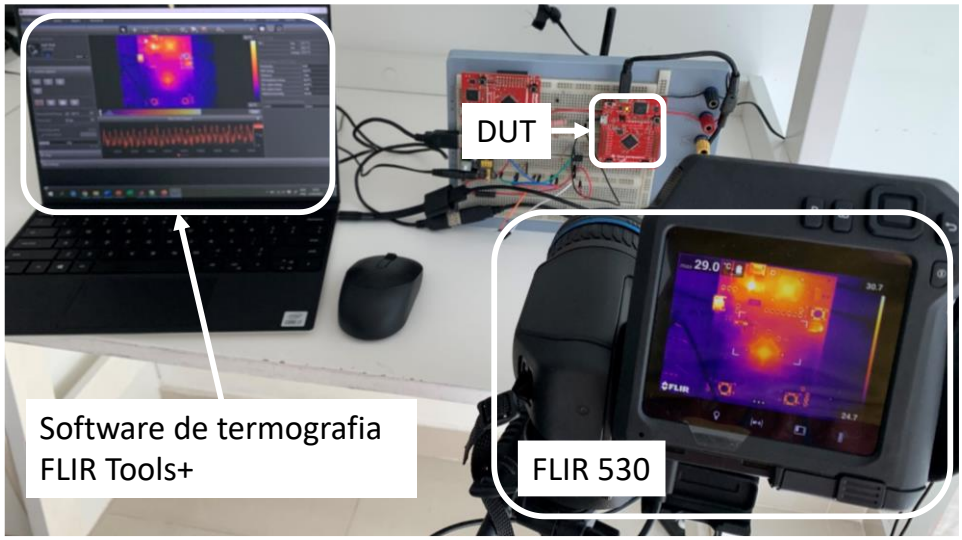


Fonte: O Autor (2022).

A assinatura termográfica de uma região de interesse (ROI – *Region of Interest*) do DUT é capturada por uma câmera de infravermelho. Um aplicativo de computador processa a informação capturada e gera uma amostra do sinal termográfico em função do tempo. Em nosso experimento, usamos uma câmera infravermelha FLIR T530 e o *software FLIRTools+* (Tools, 2016) para processamento de dados (detalhe na Figura 56). A aquisição de uma amostra é feita por um período de tempo pré-definido ΔT. Em seguida, o espectrograma do sinal

adquirido é construído e convertido em uma imagem (ImaA na Figura 55). O autoencoder treinado reconstrói a imagem de entrada em sua saída (ImaB). A imagem reconstruída e a imagem original são comparadas. Se a diferença entre as imagens for maior que um limiar μ, uma anomalia é detectada, conforme descrito na Seção 3.8.

Figura 56 – Sistema de aquisição de dados infravermelho com uma câmera FLIR T530



Fonte: O Autor (2022).

5.1 METODOLOGIA

Esta seção apresenta uma visão geral do método proposto, parâmetros-chave e geração e aquisição de dados. Aqui, são discutidos detalhes de implementação e considerações a respeito do sistema de detecção de anomalias baseados em assinatura termográfica.

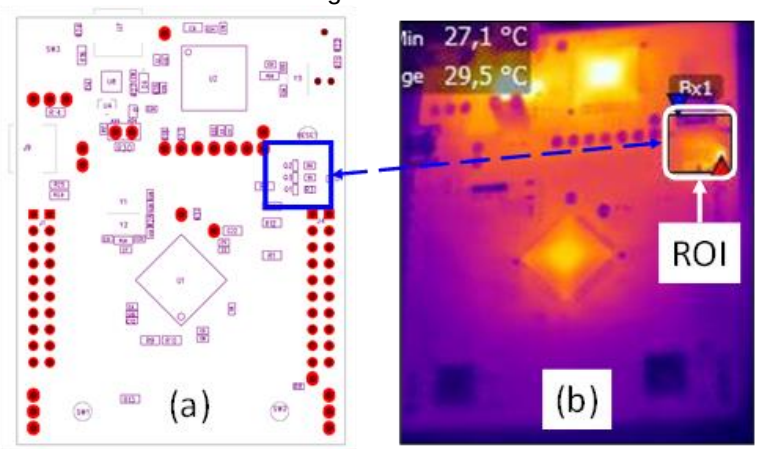
5.1.1 Protótipo experimental

O DUT desenvolvido para os testes experimentais é idêntico ao utilizado no Capítulo 4 (detecção de anomalias por assinatura elétrica). Portanto, do ponto de vista do *hardware*, a descrição em 4.1.1 é suficiente para descrever o DUT utilizado nos experimentos relatados neste Capítulo. Do ponto de vista funcional, duas versões de *firmware* foram desenvolvidas para testar o modelo desenvolvido. Detalhamos as diferenças entre essas versões na Seção 5.1.2.

5.1.2 Dados experimentais

A assinatura termográfica da ROI é capturada por uma câmera de infravermelho. Essa assinatura corresponde à temperatura máxima da ROI, observada em função do tempo. A Figura 57 ilustra a definição da região de captura e um exemplo da imagem termográfica da PCI. Nessa imagem, regiões mais claras (amarelo – vermelho) apresentam temperaturas mais elevadas, enquanto que cores mais escuras (vede – azul) indicam temperaturas mais baixas. O valor exato das temperaturas é obtido a partir da aplicação *FlIRTools+*.

Figura 57 – Região de interesse (ROI) da PCI. A temperatura máxima no quadrado marcado é adquirida e traçada em função do tempo. (a) – leiaute da placa de circuito impresso; (b) – imagem termográfica da PCI



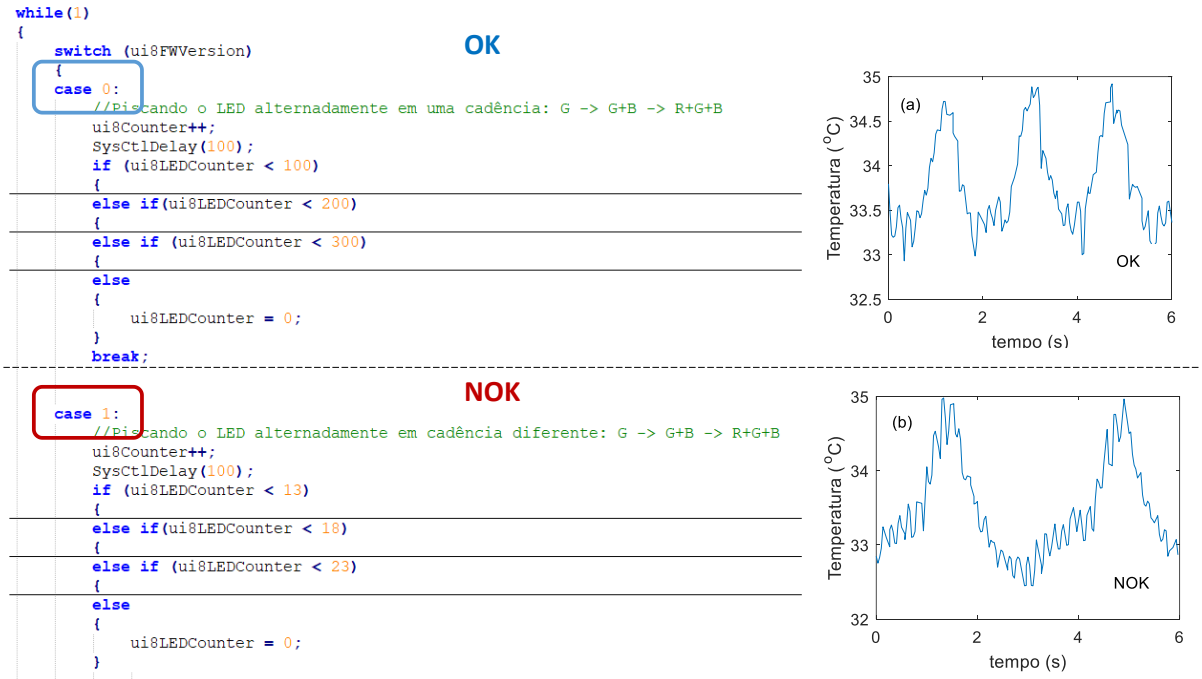
Fonte: O Autor (2022).

Em nosso experimento, a ROI indicada na Figura 57 foi escolhida por apresentar uma variação temporal de temperatura suficiente para detecção de anomalias. Nessa região se encontram os terminais do LED RGB, que é acionado (liga/desliga) ao longo da execução do *firmware*. Em uma aplicação real, a definição da ROI é parte do projeto de desenvolvimento do sistema de detecção de anomalias e depende de cada tipo de sistema a ser testado. É importante frisar que a variação de temperatura na ROI é decorrente (majoritariamente) da variação na corrente elétrica que percorre os terminais do LED e não pela irradiação luminosa, que aliás, é visível.

O *firmware* executado pelo DUT possui estrutura típica de uma aplicação embarcada: uma configuração inicial do *hardware* seguido por um loop infinito. Especificamente, o sistema aciona periodicamente um LED RGB (Vermelho Verde Azul) e realiza a transmissão de dados via interface serial (UART). Nos testes

experimentais, foram implementadas duas versões de *firmware*, uma das quais representa uma forma operacional sem anomalias. A outra versão simula uma anomalia a ser detectada pelo autoencoder. No experimento, chaves de controle da placa foram utilizadas para alternar a execução entre as versões de *firmware*. O índice “OK” indica a versão padrão (sem anomalia) do *firmware*. Enquanto que os casos anômalos são indicados pelo índice “NOK”. Na Figura 58, estão ilustrados exemplos (trechos) de duas versões de *firmware* de teste utilizadas no experimento e as respectivas assinaturas termográficas, dadas em função do tempo. Cada sinal ilustrado corresponde a uma amostra do sinal termográfico.

Figura 58 – Exemplo de duas versões do *software* embarcado e a assinatura termográfica para cada caso



Fonte: O Autor (2022).

A estrutura geral do *firmware* é similar à apresentada na Figura 38 da Seção 4.1.2. Um resumo das versões de *firmware* com a descrição de seu comportamento pode ser visto na Tabela 9.

Tabela 9 – Resumo das versões de *firmwares* usados nos experimentos: descrição operacional de cada versão

Designação →	**OK**	**NOK**
Comportamento esperado	Acionamento periódico do LED RGB	Acionamento periódico do LED RGB
Modificação (anomalia simulada)	Nenhuma (sem anomalia)	Padrão de acionamento do LED modificado + atraso na transmissão via UART
Quantidade de amostras	1204	1204

Fonte: O Autor (2022).

Diferentemente do caso elétrico, por uma questão de simplificação, geramos amostras com apenas uma largura de janela. Assim, os sinais termográficos possuem apenas a duração de aquisição $\Delta T = 6000\,ms$. A explicação para essa largura específica é a seguinte. Por se tratar de um sinal térmico, o conteúdo espectral é bem mais limitado do que no caso da corrente elétrica. Pois, mesmo que a frequência de acionamento (liga/desliga) do LED RGB seja elevada, a resposta do sinal térmico é mais lenta. Dessa forma, a amostra do caso termográfico precisa ser mais longa para garantir uma quantidade de informação mínima para a separação entre as classes dos sinais. E assim, viabilizar a detecção de anomalias. Com amostras mais longas, o tempo total de aquisição dos dados também aumenta. Entretanto, durante os experimentos, notamos que o sistema câmera + *FLIRTools+* apresenta instabilidade após duas horas de aquisição. Isso acaba limitando a quantidade de amostras capturadas para o experimento. Apesar dessa limitação, os resultados obtidos foram satisfatórios, sobretudo se aplicarmos as técnicas de aperfeiçoamento discutidas no Capítulo 7.

Após a geração dos sinais de temperatura, geramos os espectrogramas e os convertemos em arquivos de imagem colorida de dimensões 96 x 128. O processo de conversão de imagem nos permite limitar a faixa espectral definindo os limites do eixo da frequência (do espectrograma), o que nos permite filtrar ruídos e outros artefatos espectrais adicionados no processo de aquisição dos dados. Para o caso considerado, o sinal termográfico contém informações espectrais importantes apenas em frequências abaixo de 10 Hz. Esta filtragem pode ser ajustada para um novo DUT por meio do parâmetro de projeto f_{Range}, que define a faixa de frequências consideradas. Uma lista com os valores dos parâmetros usados nos

experimentos está mostrada na Tabela 10 (Ver seção 3.3 para definição de parâmetros).

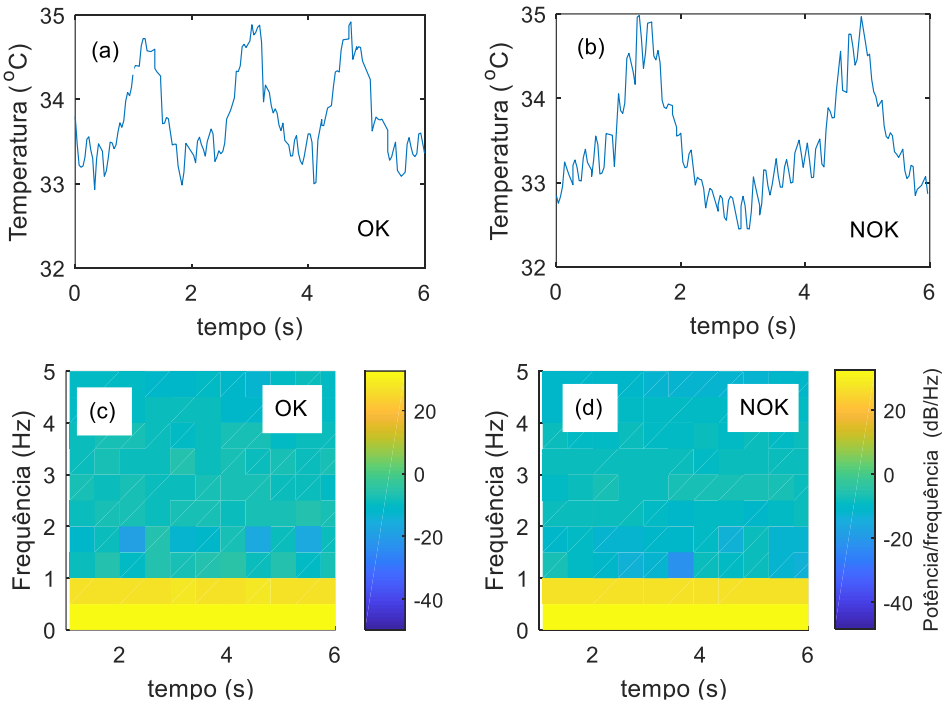
Tabela 10 – Parâmetros de geração do espectrograma para o caso termográfico

Parâmetro	Valor
S_len	64
N_overlap	50
Nf	60
Fs	30 Hz
ΔT	6 s
f_{Range}	0 a 5 Hz

Fonte: O Autor (2022).

A Figura 59 traz exemplos de amostras de sinal e espectrogramas das classes OK e NOK. Observe que as imagens que são repassadas ao autoencoder não possuem as informações dos eixos que aparecem nos espectrogramas dessa imagem.

Figura 59 – Exemplos de sinais termográficos ($\Delta T = 6\ s$). (a) – OK; (b) – NOK. (c) e (d) – Exemplos de espectrogramas das janelas (a) e (b), respectivamente



Fonte: O Autor (2022).

A Tabela 11 lista as versões de *firmware*, as quantidades de imagens geradas de cada classe e sua utilização no experimento.

Tabela 11 – Quantidade de amostras geradas de cada versão e sua utilização (NA - Não Aplicável)

	Classe	
	OK	**NOK**
Uso	**Quantidade**	
Treinamento do autoencoder	643	NA
Validação do autoencoder	160	NA
Teste de do autoencoder	401	401
Treinamento da random forest	903	903
Teste da random forest	301	301
Total	1204	1204

Fonte: O Autor (2022).

O tempo total de aquisição do experimento foi de aproximadamente duas horas para cada classe devida às limitações do sistema de aquisição (câmera + *FLIRTolls+*).

5.2 RESULTADOS

A seguir, descrevemos a analisamos os dados obtidos experimentalmente para validar nossa proposta. Diferentemente do caso de detecção de anomalias por assinatura de corrente elétrica, não realizamos uma análise exploratória, pois a viabilidade de detecção de anomalias usando espectrogramas e autoencoders já havia sido experimentalmente confirmada (Seção 4.2). Especificamente para o caso descrito neste capítulo, os seguintes resultados são relatados:

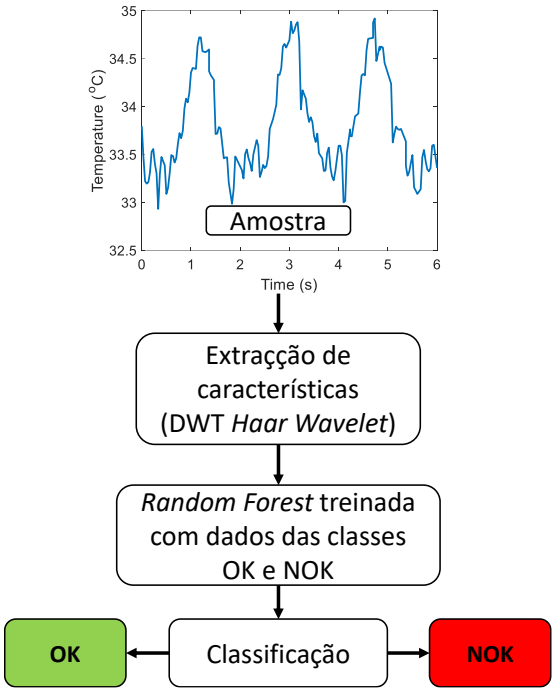
- *Random forest*
- Treinamento do autoencoder
- Custo computacional do treinamento do autoencoder
- Considerações sobre uso de transformações dos dados para viabilizar a detecção

5.2.1 Random Forest

O modelo *random forest* (ver descrição na Seção 3.1.6) foi construído usando a linguagem R (Cowpertwait, 2009) e uma biblioteca bem estabelecida para este fim. Para melhorar a qualidade da classificação, as características das amostras foram previamente extraídas por meio de análise *wavelet*, de acordo com a ilustração explicativa da Figura 60. O classificador lê as amostras, extrais suas características e as envia ao modelo treinado para classificação. Observe que a *random forest*

processa o sinal termográfico diretamente, ou seja, em forma de série temporal. Não é preciso gerar espectrogramas nem imagens. Neste experimento, os dados foram separados em uma razão 75% e 25% para treinamento e teste, como indicado na Tabela 11.

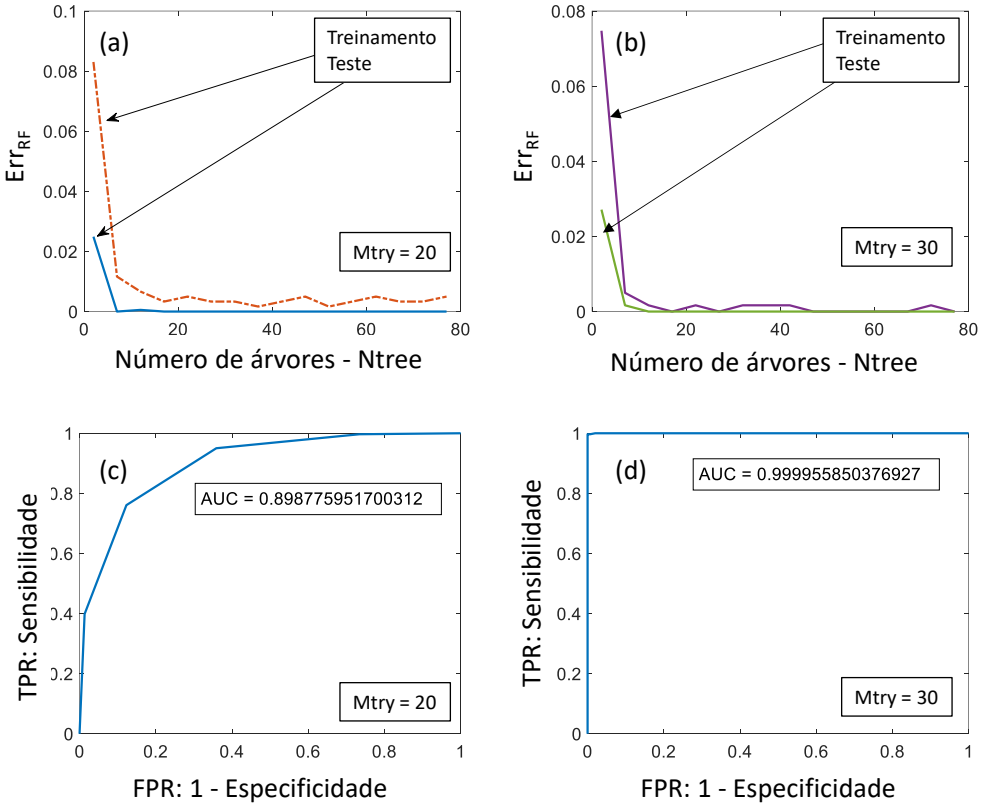
Figura 60 – Estrutura de classificação com *random forest* e extração de características com *wavelet*



Fonte: O Autor (2022).

O desempenho do modelo pode ser aperfeiçoado por meio do ajuste de seus parâmetros construtivos. Assim, observamos o desempenho do modelo em função dos parâmetros $Ntree$ e $Mtry$, seu número de árvores e o número de variáveis amostradas aleatoriamente como candidatas em cada rodada. Para o experimento, para detecção de anomalias, escolhemos o modelo com $Ntree = 80$ e $Mtry = 30$ por ter apresentado os melhores resultados, com um erro de teste (Err_{RF}) de aproximadamente 0.16 %, conforme indicam as curvas da Figura 61.

Figura 61 – Resultados. (a) e (b) - erros de treinamento e de teste em função do número de árvores ($Ntree$) e para dois valores de $Mtry$. (c) e (d) - Curvas ROC e AUC para dois valores de $Mtry$



Fonte: O Autor (2022).

A matriz de confusão resultante dos testes realizados com as 602 amostras (301 OK e 301 NOK) é mostrada na Tabela 12, de onde se confirma o $Err_{RF} = 1/602 = 0.0016\ (0.16\%)$.

Tabela 12 – Matriz de confusão do teste da *random forest* para o caso termográfico

		Classe prevista	
		OK	NOK
Classe real	OK	**300**	0
	NOK	1	**301**

Fonte: O Autor (2022).

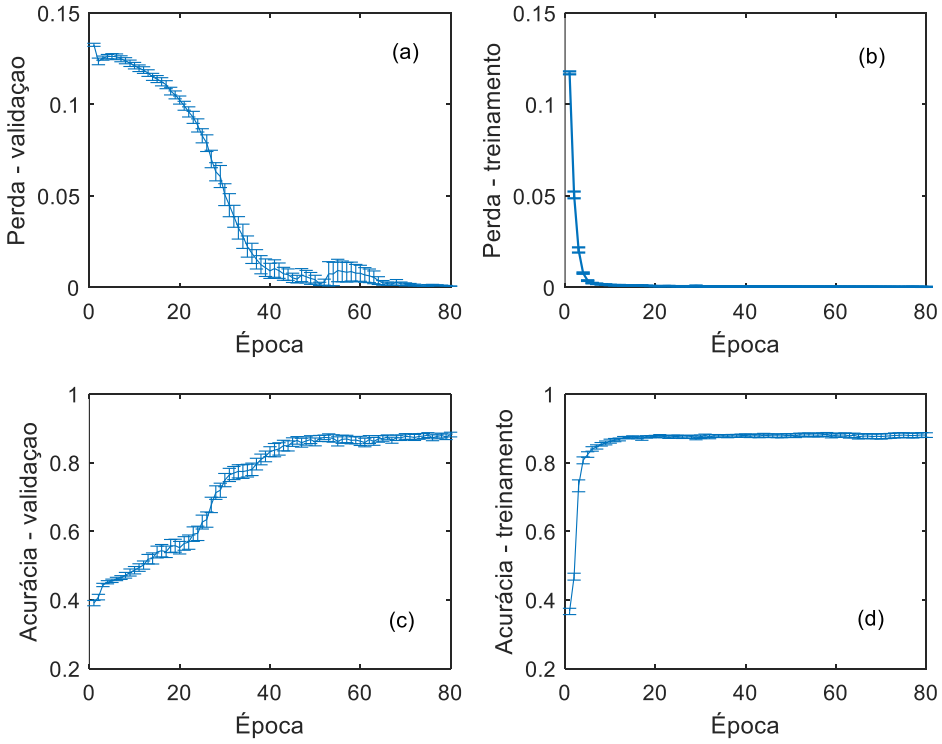
Apesar do resultado satisfatório ($Err_{RF} = 0.0016\ e\ AUC \cong 1$), ressaltamos que modelos do tipo *random forest* requerem dados de todas as classes para seu treinamento, o que para detecção de anomalias em sistemas embarcados reais é um fator limitante. Por isso, em nossa proposta, utilizamos autoencoders.

5.2.2 Autoencoder

Realizamos validação cruzada durante o processo de treinamento para evitar o *overfitting* (Seção 3.1.1) do autoencoder. Usamos subconjuntos de dados para realizar treinamento e validação, conforme listado na Tabela 11. Cada procedimento de treinamento/validação e teste foi repetido 50 vezes para termos um resultado estatisticamente relevante. Ou seja, repetimos o treinamento de modo que a variância dos resultados, graficamente observada por meio de gráficos do tipo boxplot ou com barra de erro, fosse limitada. Portanto, a variância das taxas de erro e demais parâmetros é tal que, cada resultado pode ser estatisticamente comparado aos demais.

Os resultados de treinamento e validação com as respectivas barras de erro são mostrados na Figura 62. Tanto a perda quanto a acurácia convergem para valores razoáveis. Assim, o processo de treinamento pode ser considerado bem sucedido. A quantidade de épocas necessárias para o treinamento convergir, entretanto, neste caso é bem mais alta do que para o treinamento do autoencoder da Seção 4.3.1. Isso ocorre devido à menor quantidade de imagens de treinamento que temos neste caso.

Figura 62 – Validação e treinamento do autoencoder com barras de erro

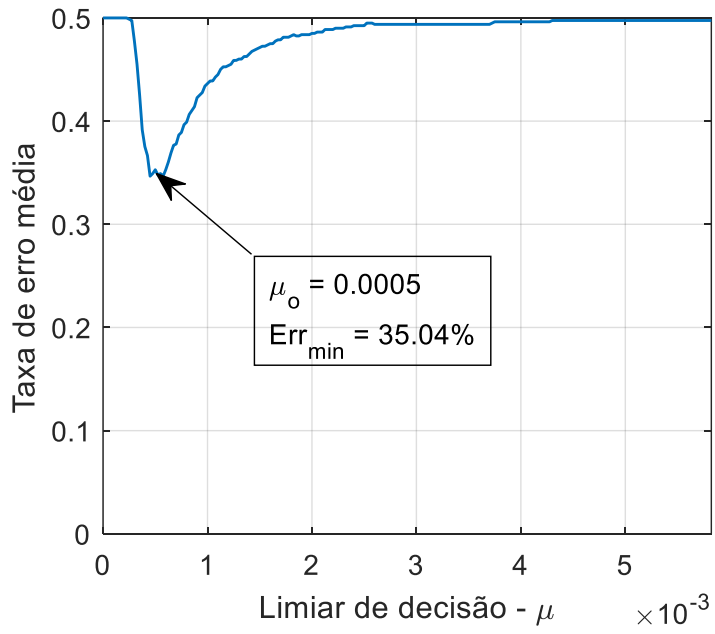


Fonte: O Autor (2022).

Utilizando o computador descrito na Seção 3.1.4, cada sessão de treinamento/validação e teste levou cerca de 10 minutos.

Após o treinamento, testamos a detecção de anomalias. Primeiro, definimos um valor para o limiar de detecção e obtemos os valores de cada elemento da matriz de confusão. Variando-se o valor do limiar de decisão, obtemos as taxas de erro e construímos as curvas mostradas na Figura 63. A taxa de erro de detecção indica a porcentagem de detecções executadas incorretamente, sejam elas falsos positivos ou falsos negativos. A Figura 63 ilustra o resultado médio dos 50 testes realizados. Existe um valor de erro mínimo (melhor caso). Nesse ponto, definimos o limite de decisão ótimo (μ_o). Pelos resultados encontrados, percebemos que o autoencoder não consegue detectar anomalias da forma esperada, uma vez que a taxa de erro mínima é de 35%, um valor demasiadamente elevado.

Figura 63 – Taxa média de erro de detecção de anomalias para amostras termográficas – média de 50 testes



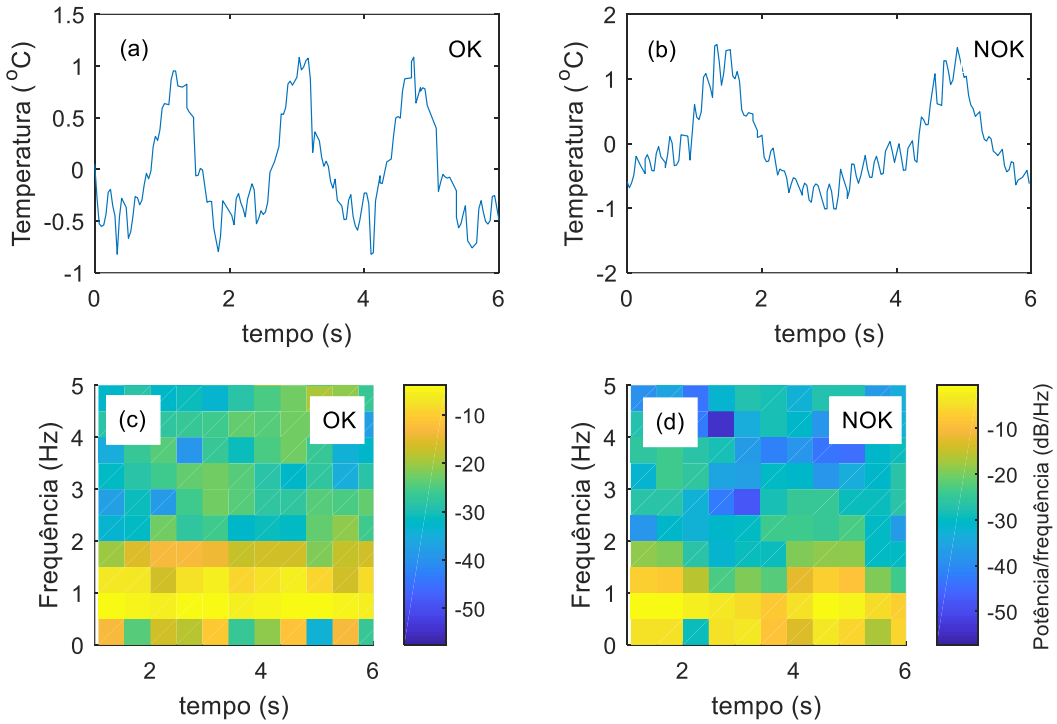
Fonte: O Autor (2022).

Esse resultado é certa forma decepcionante, sobretudo quando comparado com os resultados obtidos para o caso da assinatura de corrente elétrica da Seção 4.3.2, mas a explicação é simples. Sinais de temperatura variam muito lentamente – as frequências observadas em nosso experimento são da ordem de 5 a 10 Hz. Portanto, os espectrogramas gerados diretamente a partir do sinal obtido apresentam uma pequena variação de intensidade da potência espectral. Assim, o

autoencoder não consegue distinguir as imagens que pertencem à classe OK daquelas que pertencem à classe NOK.

Para contornar essa restrição, alteramos o sinal de temperatura utilizando uma transformação não linear simples: o nível DC do sinal é removido antes de obter o espectrograma. Esta simples transformação representa uma forma de pré-processamento aplicada aos dados. No *Matlab*, isso é feito simplesmente através da função $detrend()$. As curvas apresentadas na Figura 64 ilustram exemplos das séries temporais transformadas e seus respectivos espectrogramas.

Figura 64 – Exemplo de pré-processamento usando a função $detrend()$ do *Matlab*



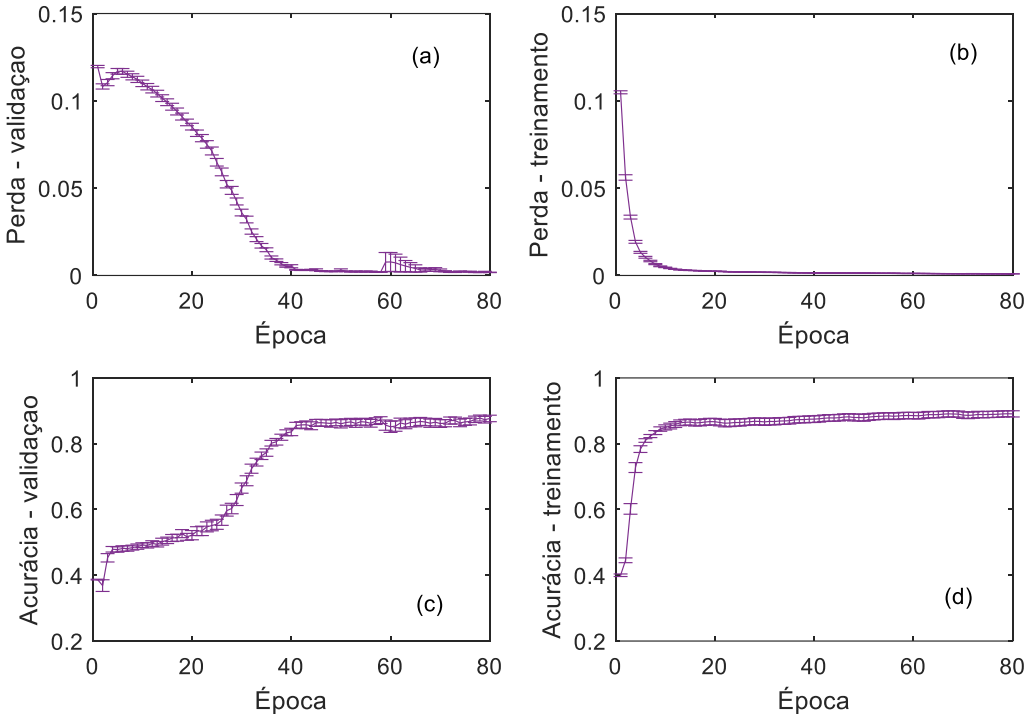
Fonte: O Autor (2022).

Se compararmos os espectrogramas da Figura 59 com os da Figura 64, podemos notar visualmente que as diferenças entre espectrogramas de classes distintas são mais pronunciadas após a aplicação da transformação $detrend()$. Uma análise quantitativa a respeito dessas diferenças é realizada no Capítulo 7, onde discutimos de forma mais aprofundada transformações que podem ser aplicadas aos dados, antes da geração dos espectrogramas, com o objetivo de aprimorar a detecção de anomalias ou classificação das amostras.

Para o caso dos sinais termográficos transformados pela função $detrend()$, os resultados são descritos a seguir. O treinamento, como pode ser averiguado na

Figura 65, é realizado com sucesso, conforme indicam os valores finais de perda e acurácia.

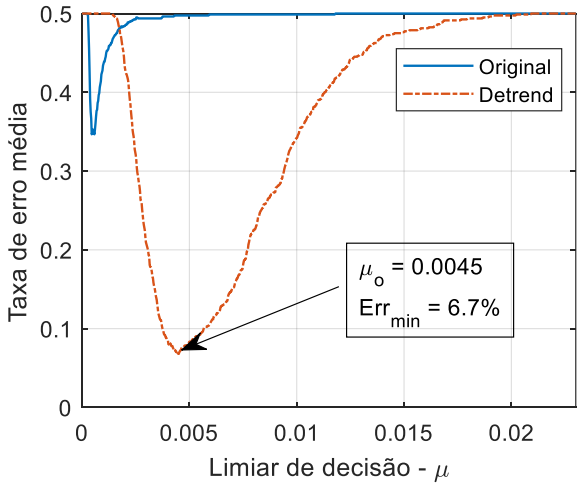
Figura 65 – Acurácia e perda de treinamento e de validação do autoencoder usando amostras processadas com a função $detrend()$



Fonte: O Autor (2022).

O desempenho do autoencoder quando submetido aos dados transformados são exibidos na Figura 66, juntamente com os resultados para a taxa de erro dos dados originais (ou seja, sem transformação) da Figura 63 para fins de comparação.

Figura 66 – Taxa média de erro de detecção de anomalias para amostras termográficas originais e transformadas com a função $detrend()$ – média de 50 testes



Fonte: O Autor (2022).

Com a aplicação da transformação $detrend()$, percebe-se que é possível detectar anomalias com uma taxa média de erro mínima menor que 7%. Apesar do valor relativamente elevado se comparado com os resultados mostrados na Figura 48 para o caso de assinatura elétrica, é um resultado consideravelmente melhor do que o obtido sem a transformação dos sinais termográficos. Além disso, essa melhoria alcançada após o uso da transformação $detrend()$ serve como inspiração para buscar outras transformações capazes de aumentar ainda mais o desempenho de sistemas de classificação ou detecção baseados em autoencoders e espectrogramas. Dedicamos o Capítulo 7 a esse assunto, onde fazemos uma análise quantitativa das melhorias obtidas com algumas transformações e propomos uma explicação para o efeito obtido.

5.3 SÍNTESE DO CAPÍTULO

Neste capítulo, investigamos a aplicação de aprendizado de máquina e termografia para detectar anomalias em sistemas embarcados. Projetamos e construímos um DUT para validação experimental, a partir do qual a assinatura termográfica é capturada. Para fins de comparação, um modelo de aprendizado de máquina baseado em *random forest* foi usado para detectar anomalias na mesma placa que usamos para validar a abordagem autoencoder.

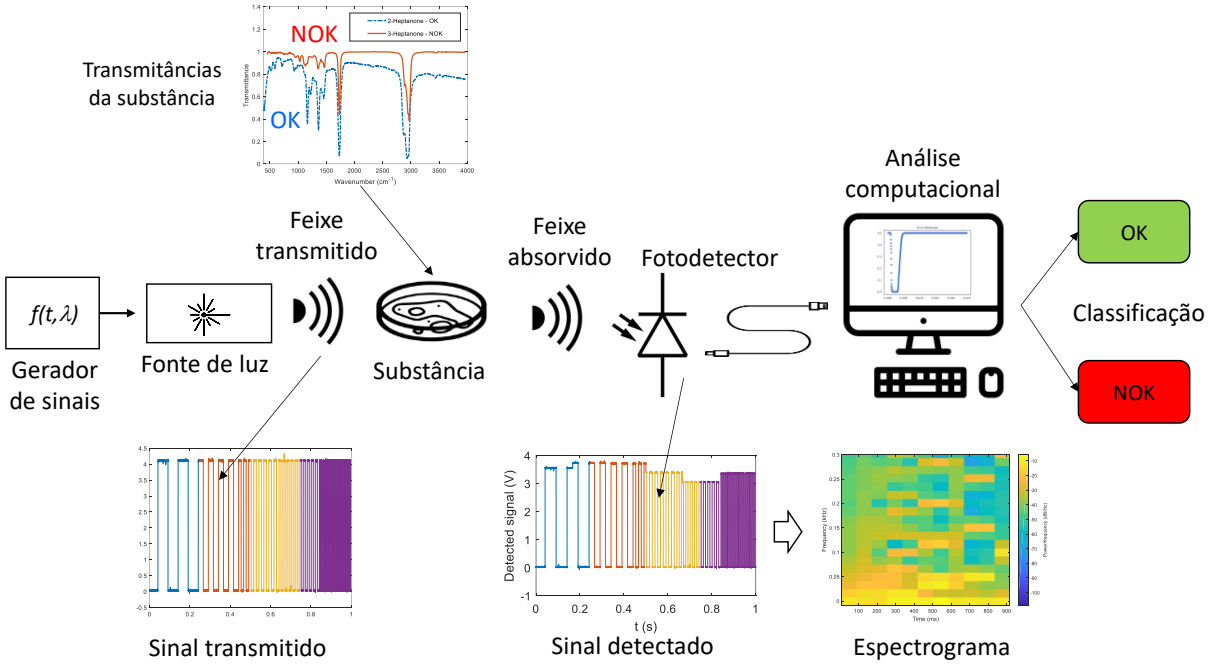
Na abordagem com autoencoder, ao usar diretamente os sinais termográficos obtidos da placa, os resultados para a taxa de erro indicam que o modelo é incapaz de detectar anomalias. Entretanto, é possível contornar essa limitação por meio da transformação dos dados originais.

Considerando, então, o uso do autoencoder com dados transformados, os resultados dos testes mostram que ambos os modelos (*random forest* e autoencoder) são viáveis. O modelo *random forest* alcança uma taxa de acerto melhor que o autoencoder. No entanto, o treinamento da *random forest* requer dados de todas as classes. Autoencoders, por outro lado, requerem apenas a classe de referência (livre de anomalias). Isso é fundamental porque, em um cenário prático, os dados de classes anômalas são difíceis de obter (Mujeeb et al., 2019).

6 CLASSIFICAÇÃO DE SUBSTÂNCIAS

Neste capítulo, apresentamos uma prova de conceito de um classificador de substâncias químicas ou biológicas. Na abordagem proposta, em vez de utilizar um espectrômetro, usamos autoencoders. O princípio de funcionamento do classificador é o seguinte. O autoencoder é treinado com imagens de espectrogramas gerados a partir de sinais ópticos detectados após a propagação através de uma substância de referência. O diagrama da arquitetura desse sistema está ilustrado na Figura 67.

Figura 67 – Arquitetura do sistema de classificação de amostras. O sinal detectado pode ser transmitido para um computador remoto via Internet, onde a classificação pode ser executada



Fonte: O Autor (2022).

A forma de onda do sinal enviado é especificamente projetada e o comprimento de onda óptico varia em função do tempo. Assim, ao se propagar através da substância, o sinal é parcialmente absorvido de acordo com os comprimentos de onda do sinal e a transmitância. Como a absorção depende da transmitância da substância, o sinal detectado representa uma assinatura da própria substância. Ou seja, alterando-se a substância, altera-se a forma de onda detectada. Assim, com base no erro de reconstrução da imagem, o autoencoder pode classificar a substância como pertencente à mesma classe da substância de referência ou não (classificador de uma classe), conforme descrito na Seção 3.8. As transmitâncias

das duas substâncias de teste utilizadas em nossa pesquisa são apresentadas na Seção 6.1.1.

6.1 METODOLOGIA

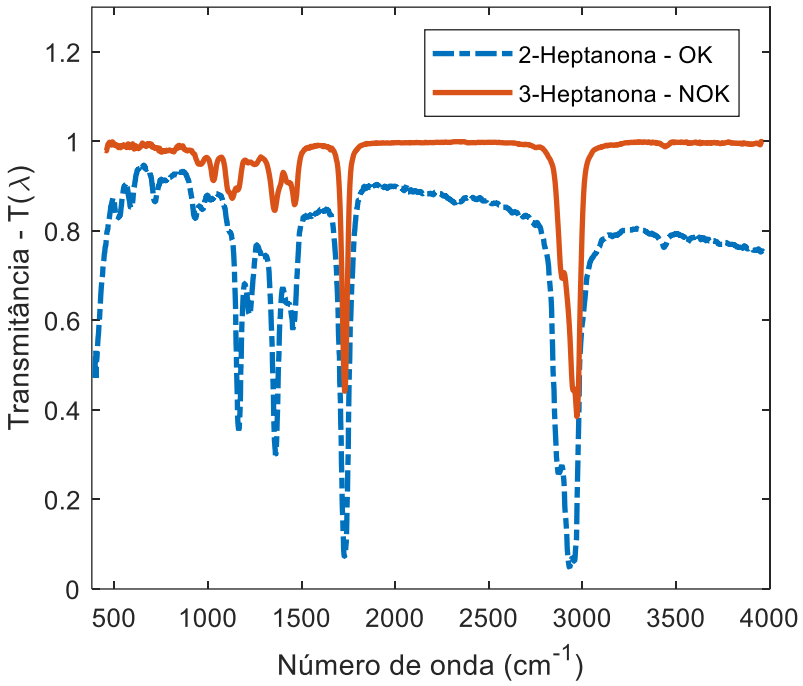
Esta seção apresenta uma visão geral do método proposto, parâmetros-chave e geração e aquisição de dados. Aqui, são discutidos detalhes de implementação e considerações a respeito do sistema de classificação de substâncias.

6.1.1 Substâncias

Nossa pesquisa considerou dois compostos químicos como substâncias de teste: 2-heptanona; 3-heptanona. A 2-heptanona, também conhecida como metil n-amil cetona, é uma cetona com a fórmula molecular $CH_3(CH_2)_4COCH_3$. Fisicamente, trata-se de um líquido incolor, semelhante à água, com um odor frutado semelhante a banana. A 2-heptanona tem uma carga formal neutra e é apenas ligeiramente solúvel em água e é um componente chave no desenvolvimento de sabores artificiais (Van der Schaft et al., 1992). A 3-Heptanona (ou butil etil cetona) é uma cetona de sete carbonos com fórmula $CH_3(CH_2)_3COCH_2CH_3$. É um líquido incolor com um aroma frutado. É frequentemente usado como perfume/fragrância e como solvente para celulose (Siegel, 2021).

Escolhemos esses compostos específicos devido à disponibilidade de seus dados espectrográficos (NIST 2-Heptanone, 2022; NIST 3-Heptanone, 2022). Os dados brutos precisaram ser convertidos do formato JCAMP-DX (Lampen et al., 1994) para um formato compatível com *Matlab*. Após esse pré-processamento, é possível obter as curvas ilustradas na Figura 68. Para nossa aplicação, separamos as substâncias em duas classes de forma arbitrária: OK (2-heptanona) e NOK (3-heptanona). Nesse caso, consideramos que a 2-heptanona é a substância de referência em nossos testes. A denominação OK está relacionada ao fato de usarmos amostras da classe OK para treinar o autoencoder.

Figura 68 – Espectros de transmitância das duas amostras de teste: 2-heptanona (Classe OK) e 3-heptanona (Classe NOK)



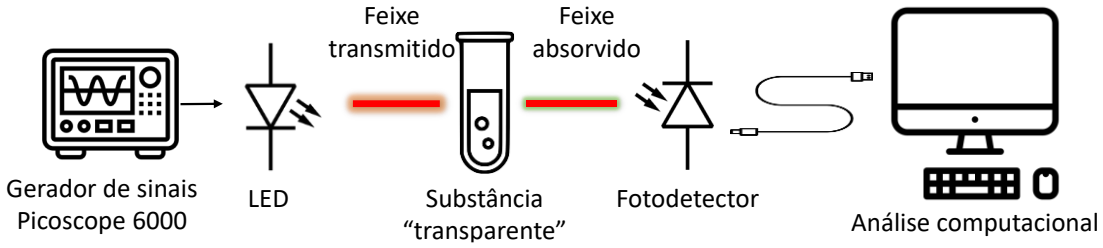
Fonte: O Autor (2022).

6.1.2 Protótipo experimental

Na ausência de amostras reais disponíveis durante o desenvolvimento desta prova de conceito, simulamos computacionalmente o comportamento das transmitâncias a partir de dados digitalizados das transmitâncias das substâncias reais. Além disso, não dispomos de uma fonte óptica sintonizável, ou seja, capaz de controlar o comprimento de onda emitido ao longo do tempo. Então, optamos por um experimento híbrido, em que simulamos computacionalmente o processo de absorção em função do comprimento de onda.

Apesar das limitações mencionadas, a transmissão e a detecção do sinal óptico foram implementadas experimentalmente. Usamos um par LED-fotodetector infravermelho e o Picoscope 6000 [21], que é um gerador de sinal e um registrador de dados. Enviamos e recebemos 14000 sinais cuja duração é de um segundo. Cada sinal detectado corresponde a um arquivo gerado pelo Picoscope 6000 e gravado em um computador. A estrutura do arranjo experimental montado em laboratório está ilustrada na Figura 69.

Figura 69 – Arranjo experimental para transmissão e detecção do sinal óptico modulado

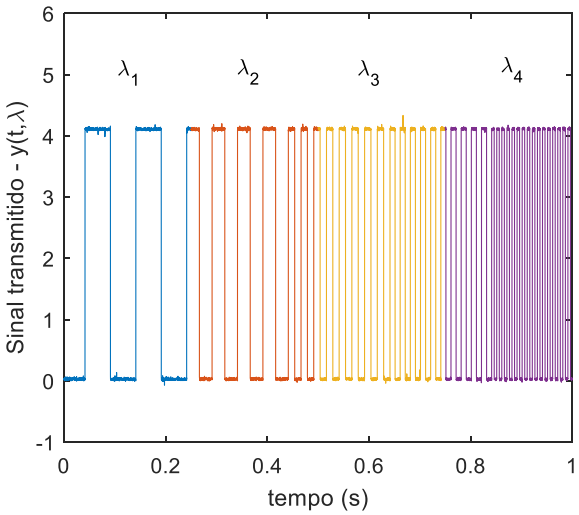


Fonte: O Autor (2022).

6.1.3 Dados experimentais

O princípio fundamental do experimento é transmitir um sinal óptico com um segundo de duração. O formato do sinal é especificamente projetado: onda quadrada cuja frequência elétrica fundamental aumenta com o tempo. O objetivo é obter um espectrograma com características que podem ser aprendidas pelo autoencoder durante o processo de treinamento. Como não dispomos das substâncias reais, simulamos o sinal óptico com múltiplos comprimentos de onda e os efeitos da absorção após propagação através da substância. Um exemplo do sinal transmitido simulado é mostrado na Figura 70. Para esse exemplo especificamente, foram usados quatro comprimentos de onda diferentes ($N_\lambda = 4$), sendo modificado a cada 0,25 s.

Figura 70 – Sinal transmitido. O comprimento de onda emitido varia ao longo do tempo. As cores são meramente ilustrativas, pois comprimentos de onda tipicamente utilizados em espectroscopia estão na faixa do infravermelho

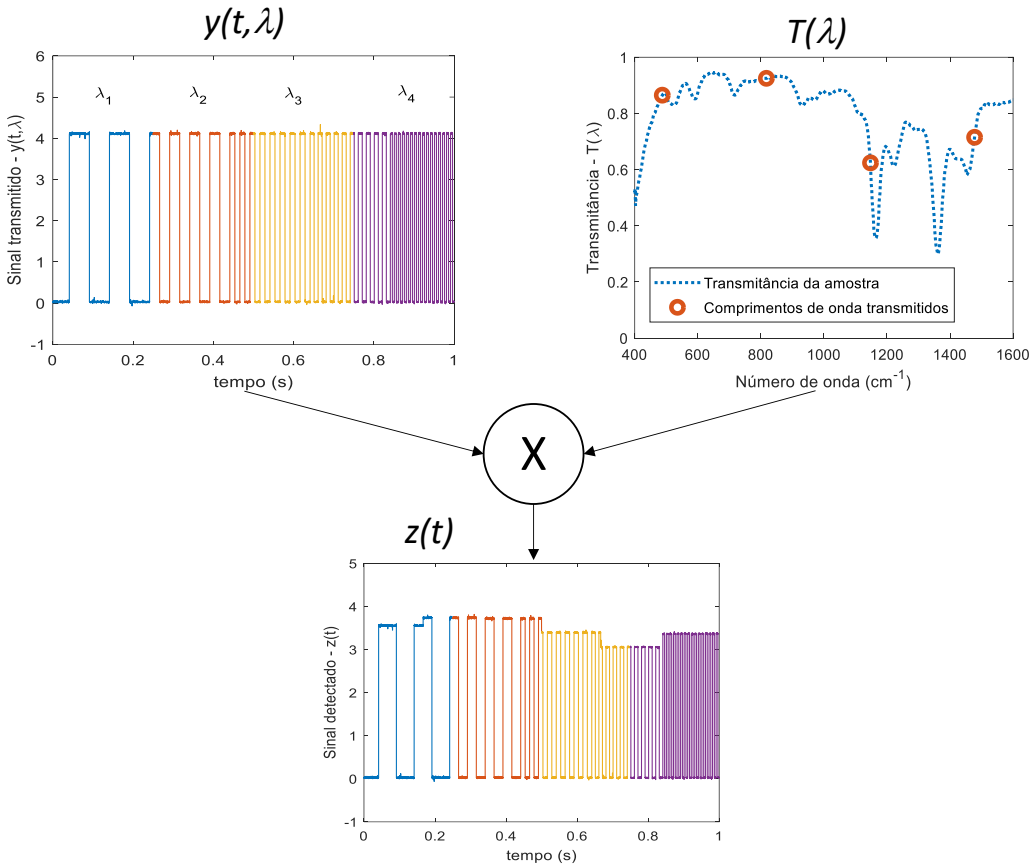


Fonte: O Autor (2022).

Após a propagação através da amostra testada, o sinal óptico é detectado. A amostra absorve cada componente espectral do sinal transmitido de uma maneira

específica, que depende da substância utilizada. Logo, a forma do sinal detectado depende do espectro de transmitância da amostra. Em nosso experimento, simulamos o sinal parcialmente absorvido multiplicando o sinal detectado pelo valor da transmitância em cada comprimento de onda presente no sinal. Esse procedimento está ilustrado na Figura 71. Multiplicando o valor da transmitância $T(\lambda)$ – nos comprimentos de onda transmitidos – pelo sinal recebido $y(t,\lambda)$, obtém-se o sinal simulado parcialmente absorvido $z(t)$.

Figura 71 – Etapas de simulação para obter o sinal parcialmente absorvido, após propagar pela amostra, $z(t) = y(t,\lambda)T(\lambda)$. Os círculos em (b) mostram as localizações dos comprimentos de onda transmitidos



Fonte: O Autor (2022).

A forma do sinal transmitido é fundamental para obter um desempenho de classificação aceitável. Assim, a modulação de frequência garante um padrão espectral que varia ao longo do tempo. Tal padrão é fortemente refletido no espectrograma do sinal detectado correspondente. A absorção experimentada pelo sinal ao passar pela amostra define as características espectro-temporal do sinal detectado. Assim, por meio de seu espectrograma, é possível diferenciar as substâncias.

De posse dos sinais detectados, geramos os espectrogramas e os convertemos em arquivos de imagem colorida de dimensões 96 x 128. O processo de conversão de imagem nos permite limitar a faixa espectral definindo os limites do eixo da frequência (do espectrograma), o que nos permite filtrar ruídos e outros artefatos espectrais adicionados no processo de aquisição dos dados. Para o caso considerado, o sinal termográfico contém informações espectrais importantes apenas em frequências abaixo de 200 Hz. A filtragem pode ser ajustada para novas substâncias por meio do parâmetro de projeto f_{Range}, que define a faixa de frequências consideradas. A lista dos valores dos parâmetros usados nos experimentos está mostrada na Tabela 13 (Ver seção 3.3 para definição de parâmetros).

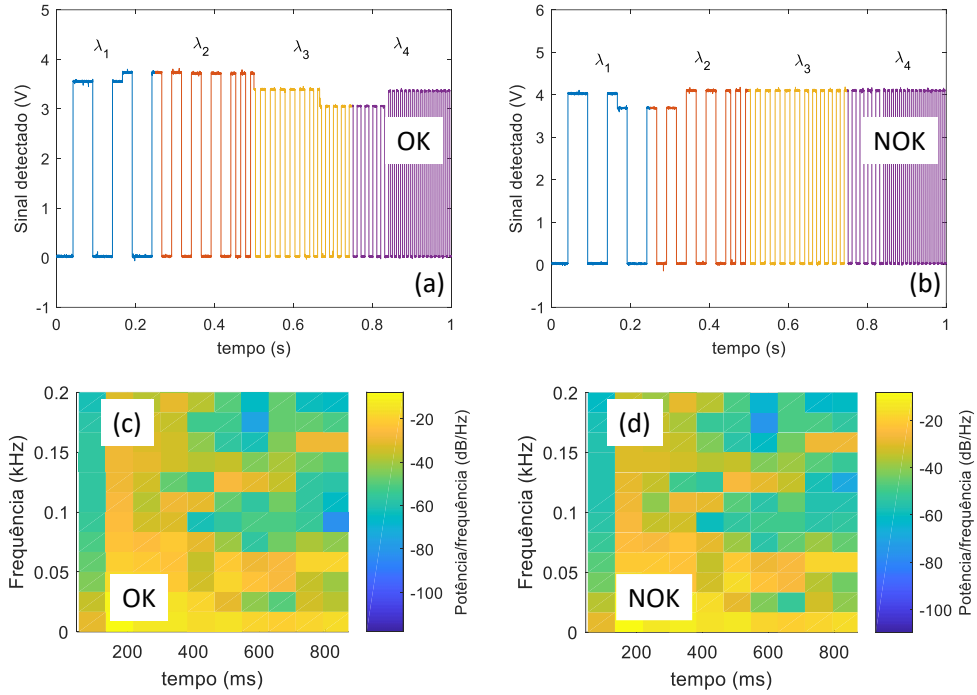
Tabela 13 – Parâmetros de geração do espectrograma para classificação de substâncias

Parâmetro	**Valor**
S_len	64
N_overlap	50
Nf	60
Fs	10000 Hz
ΔT	1 s
f_{Range}	0 a 200 Hz

Fonte: O Autor (2022).

A Figura 72 traz exemplos de amostras de sinal e espectrogramas das classes OK e NOK. Observe que as imagens que são repassadas ao autoencoder não possuem as informações dos eixos que aparecem nos espectrogramas na imagem.

Figura 72 – Exemplos de sinais detectados ($\Delta T = 1\ s$). (a) – OK; (b) – NOK. (c) e (d) – Exemplos de espectrogramas das amostras (a) e (b), respectivamente



Fonte: O Autor (2022).

Após a construção dos espectrogramas, geramos arquivos de imagens coloridas. Em seguida, dividimos as imagens em dois grupos: treinamento/validação (6000 imagens OK); e teste (4000 imagens OK e 4000 imagens NOK). O treinamento do autoencoder é realizado com imagens (espectrogramas) obtidas da amostra de referência (OK). Um sumário das classes de imagens e suas aplicações pode ser visualizado na Tabela 14. O processo de validação cruzada separar as amostras OK em dois grupos numa razão de 75% e 25% para treinamento e validação, respectivamente.

Tabela 14 – Resumo das classes das amostras usadas no experimento

Designação →	**OK**	**NOK**
Substância	2-heptanona	3-heptanona
Quantidade usada no treinamento	6000	*NA*
Quantidade usada no teste	4000	4000

Fonte: O Autor (2022).

O tempo total de aquisição do experimento foi de aproximadamente quatro horas para armazenar no computador os 14000 (6000 OK + 4000 OK + 4000 NOK) sinais

transmitidos. A obtenção de todos sinais detectados por meio de simulação em um computador portátil (ver Figura 71) pessoal durou em torno de uma hora.

6.2 RESULTADOS

A seguir, descrevemos a analisamos os dados obtidos para validar nossa proposta. Especificamente para o caso descrito neste capítulo, os seguintes resultados são relatados:

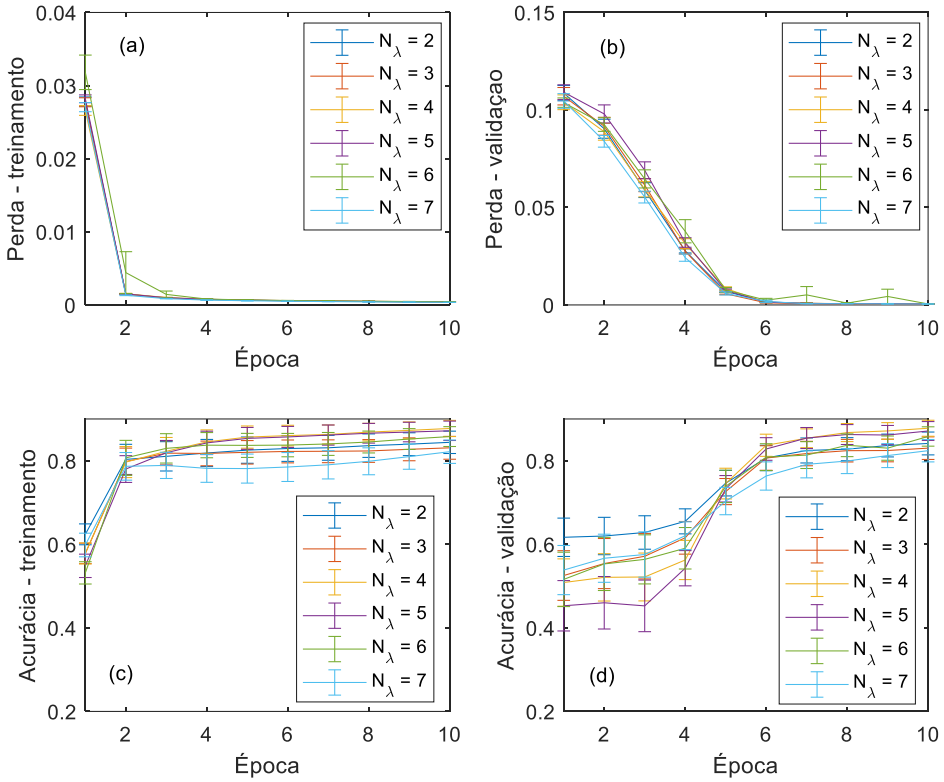
- Treinamento do autoencoder
- Custo computacional do treinamento do autoencoder
- Desempenho do sistema de classificação em função do número de comprimentos de onda transmitidas

Para estimar a complexidade necessária do sistema, avaliamos a relação entre o desempenho e o número de comprimentos de onda transmitidos (N_λ). Consideramos seis configurações para o sinal transmitido: $N_\lambda = 2, 3, 4, 5, 6\ e\ 7$. Em todos os casos, o número de onda varia de $1/\lambda = 500$ a $1/\lambda = 1500$ cm^{-1}, com distribuição uniforme entre esses extremos. Entretanto, em aplicações reais, isso pode ser ajustado de acordo com as características de cada substância.

6.2.1 Treinamento do autoencoder

Para garantir que os resultados sejam estatisticamente relevantes, treinamos o autoencoder várias vezes (21 sessões de treinamento neste estudo). Ou seja, repetimos o treinamento de modo que a variância dos resultados, graficamente observada por meio de gráficos do tipo *boxplot* ou com barra de erro, fosse limitada. Portanto, a variância das taxas de erro e demais parâmetros é tal que, cada resultado pode ser estatisticamente comparado aos demais. Os resultados do treinamento são exibidos na Figura 73. Os valores finais de Perda e Acurácia, após 10 épocas, demonstram que o autoencoder foi treinado de forma consistente.

Figura 73 – Validação e treinamento do autoencoder com barras de erro para todas as janelas testadas



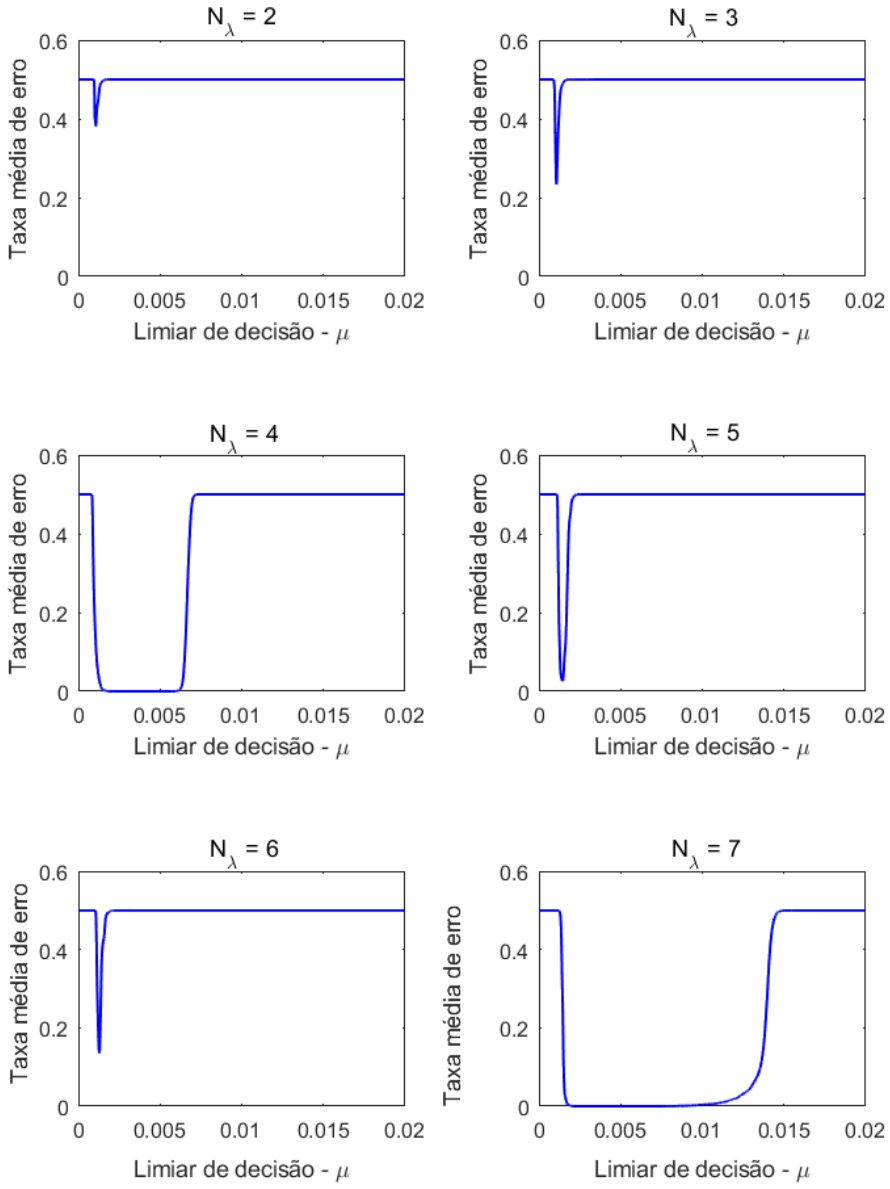
Fonte: O Autor (2022).

O treinamento foi realizado em um computador portátil equipado com placa de processamento gráfico (ver Seção 3.1.4). Para as janelas listadas na Tabela 14, o tempo médio foi de aproximadamente 6:40 minutos, ou seja, em torno de 40 segundos por época. Após o treinamento, cada teste realizado com os dados das classes OK e NOK leva menos de 30 segundos para ser finalizado.

6.2.2 Desempenho

A Figura 74 mostra o desempenho de classificação obtido em função do limiar de decisão para diferentes configurações do sinal transmitido, i. e., a quantidade de comprimentos de onda transmitidos N_λ. Cada gráfico representa a média dos 21 testes. Os resultados mostram que o erro mínimo médio é zero (100% de precisão) para os casos $N_\lambda = 4$ e 7. Isso confirma a eficácia do classificador proposto.

Figura 74 – Taxa média de erro de classificação mínima em função do limiar de decisão μ para várias configurações do sinal transmitido N_λ



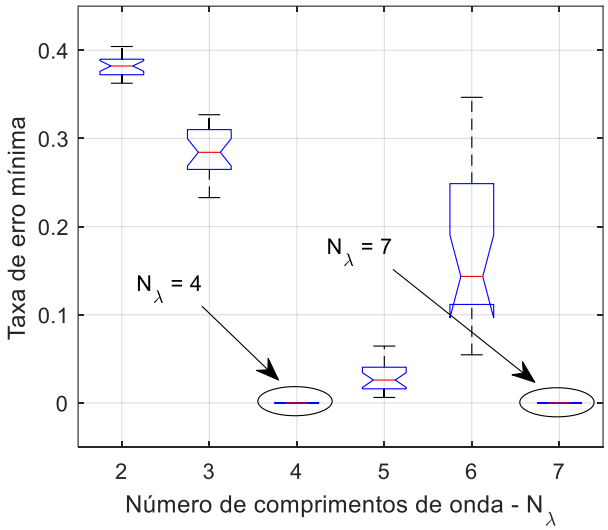
Fonte: O Autor (2022).

Os resultados da também mostram que a taxa de erro mínima como uma função de N_λ não é monotônica. Isso significa que aumentar o número de comprimentos de onda transmitidos não leva necessariamente a uma melhora no desempenho. A razão desse comportamento é a distribuição de comprimentos de onda ao longo do espectro de transmitância. Se os comprimentos de onda estiverem posicionados em pontos onde as diferenças entre as transmitâncias das classes OK e NOK são pequenas, os sinais detectados serão mais semelhantes. Isso torna a classificação mais difícil, pois neste caso as imagens reconstruídas das classes OK e NOK produzem erros de reconstrução com valores próximos. Logo, vale ressaltar que,

embora a implementação do sistema proposto não exija conhecimento prévio do espectro de transmitância da amostra, tal conhecimento pode ajudar a aumentar o desempenho, já que fornece informação que pode ser usada na distribuição dos comprimentos de onda transmitidos.

De uma forma mais explícita, na Figura 75 temos a taxa mínima de erro em função de N_λ considerando os valores estatísticos (*boxplot*) para os 21 testes realizados. O resultado mostra que os casos $N_\lambda = 4$ e 7 apresentam os melhores desempenhos de forma estatisticamente relevante.

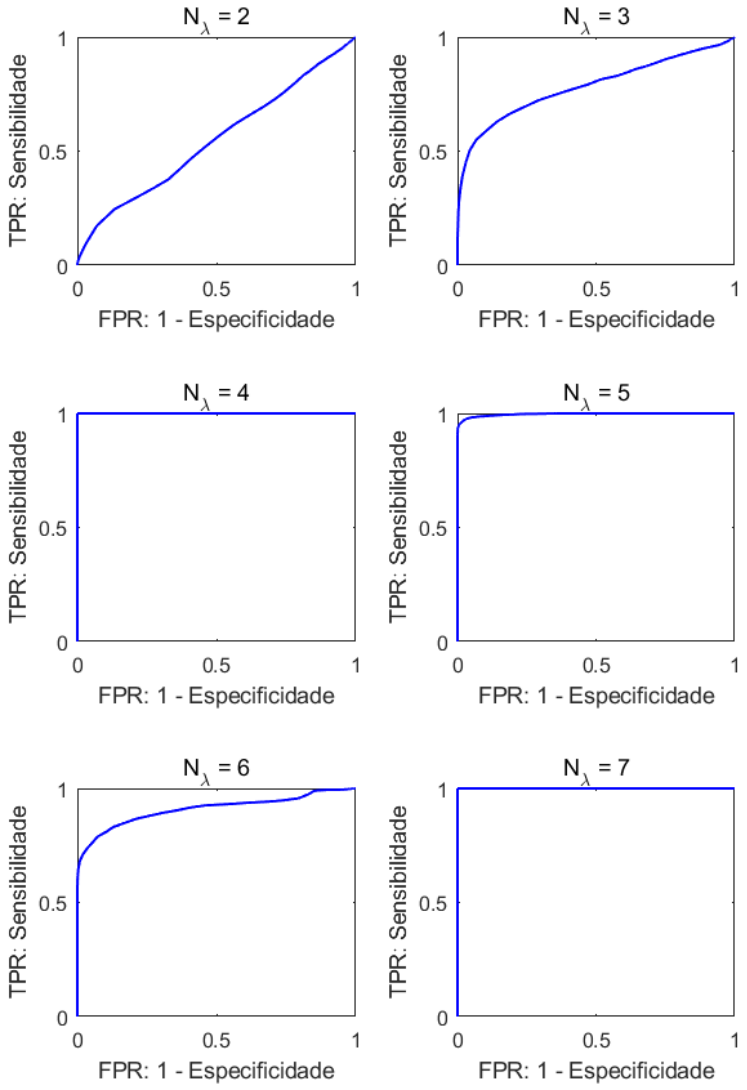
Figura 75 – Taxa mínima de erro para várias configurações do sinal transmitido N_λ. *Boxplot* considerando os 21 testes



Fonte: O Autor (2022).

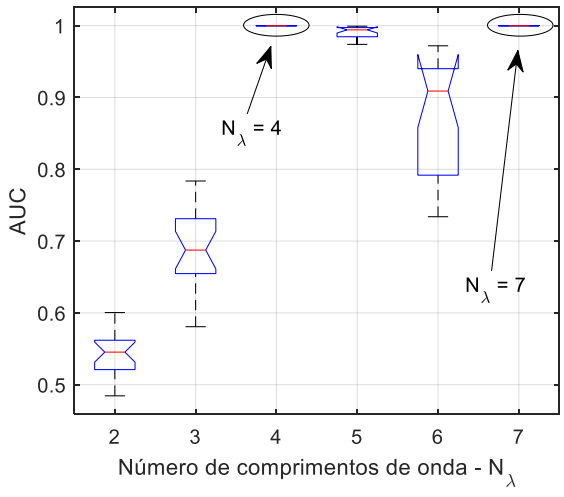
O resultado médio das curvas ROC é exibido na Figura 76. E o resultado médio para a AUC correspondente está mostrado na Figura 77. Esses resultados apenas corroboram com o que foi obtido por meio da taxa mínima de erro média.

Figura 76 – Taxa mínima de erro para várias configurações do sinal transmitido N_λ. *Boxplot* considerando os 21 testes



Fonte: O Autor (2022).

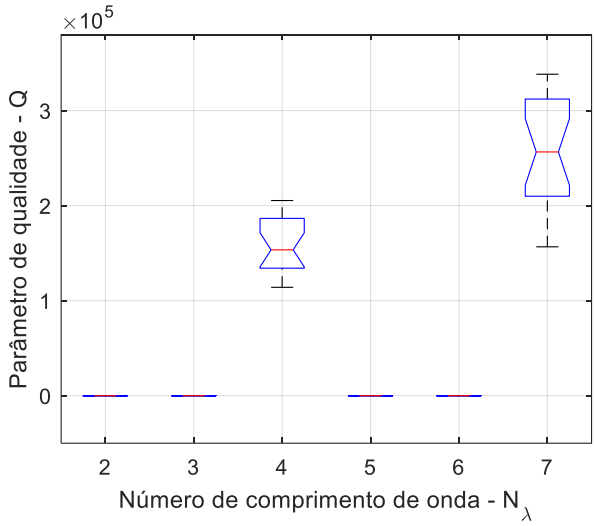
Figura 77 – Área sob a curva (AUC) para várias configurações do sinal transmitido N_λ. *Boxplot* considerando os 21 testes



Fonte: O Autor (2022).

Como consideração final, é importante ressaltar que, embora os casos $N_\lambda = 4$ e 7 apresentem os mesmos resultados quanto à taxa mínima de erro de classificação, as curvas da Figura 74 indicam que o caso $N_\lambda = 7$ tem vantagem sobre o caso $N_\lambda = 4$. Pois, a faixa do limiar de decisão, onde o valor da taxa de erro é zero, é maior para o primeiro caso. Assim, em um cenário real, estimar o valor ótimo para o limiar de decisão torna-se mais simples (para $N_\lambda = 7$). Quantitativamente, podemos comparar os resultados para $N_\lambda = 4$ e $N_\lambda = 7$ por meio do parâmetro de desempenho Q definido pela equação (14). A Figura 78 mostra os valores de Q para as duas situações.

Figura 78 – Parâmetro de qualidade Q em função do número de comprimento de ondas transmitidos. *Boxplot* considerando os 21 testes



Fonte: O Autor (2022).

Os resultados da Figura 78, além de comprovarem a vantagem que o caso $N_\lambda = 7$ apresenta em relação ao caso $N_\lambda = 4$, demonstram a efetividade do parâmetro Q para avaliação quantitativa de modelos de classificação. Pois, usando apenas os parâmetros taxa de erro mínima e AUC, os resultados seriam idênticos para esses dois casos.

6.3 SÍNTESE DO CAPÍTULO

Neste capítulo, apresentamos uma Prova de Conceito de um sistema de classificação de substâncias químicas ou biológicas: um classificador de aprendizado de máquina baseado em análise espectral e autoencoders. Validamos a técnica proposta através da transmissão/detecção de 14000 sinais. Cada sinal foi modificado computacionalmente de acordo com o espectro de transmitância de dois

compostos químicos de teste. O classificador é capaz de distinguir qual das duas substâncias é testada.

O diferencial da abordagem proposta é o formato do sinal transmitido, composto por múltiplas frequências (elétricas) e múltiplos comprimentos de onda (ópticos). Em princípio, qualquer forma de onda rica em conteúdo espectral pode ser usada. As ondas quadradas foram escolhidas por sua simplicidade e pela alta precisão alcançada em testes preliminares.

Também observamos que o desempenho da classificação depende fortemente da localização do comprimento de onda em relação ao espectro de transmitância. Uma precisão de classificação de 100% foi alcançada para $N_\lambda = 4$ e 7. A dependência do desempenho do número de comprimentos de onda pode ser usada para estimar quantos transmissores e fotodetectores são necessários na prática. Na configuração descrita neste trabalho em particular, apenas um par LED/fotodetector foi utilizado, pois a absorção espectral foi simulada. Na prática, é sempre conveniente reduzir o número de componentes. Por essa razão, é relevante conhecer a relação entre o desempenho e o número mínimo de comprimentos de onda possíveis. No experimento descrito, o parâmetro Q indica que, do ponto de vista da implementação prática, o sistema com $N_\lambda = 7$ é mais vantajoso. Contudo, é importante ressaltar que, utilizar mais comprimentos de onda no transmissor implica aumento de complexidade, portanto de custo.

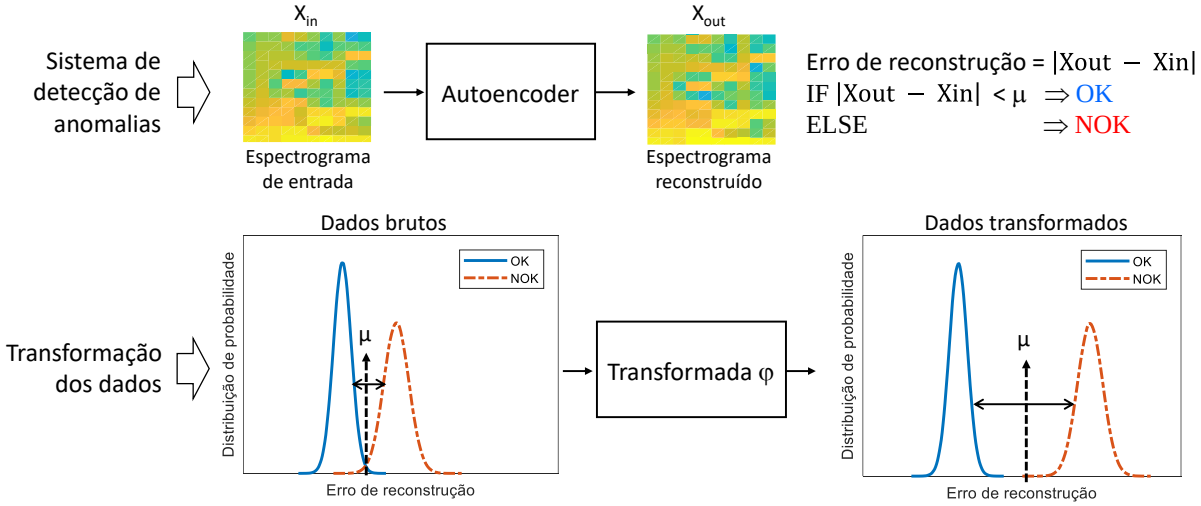
Em relação às aplicações potenciais, o arcabouço de classificação aqui proposto pode ser adaptado para diversas aplicações na indústria, por exemplo: avaliação da qualidade da água (Heibati et al., 2017); detecção de patógenos (como COVID-19) (Cho et al., 2014); certificação de conformidade de substâncias químicas ou alimentos orgânicos (Tres et al., 2012); verificação da saúde de baterias (Galatro et al., 2022).

7 APERFEIÇOAMENTO POR MEIO DE PRÉ-PROCESSAMENTO

Os resultados descritos em 0, 5.2 e 6.2 certamente podem ser considerados positivos, do ponto de vista da precisão do modelo de aprendizado de máquina. Contudo, há espaço para melhoria em todos os casos estudados. A busca pelo aperfeiçoamento do modelo, portanto, se justifica. Sobretudo, se considerarmos o fato de que, em um ambiente real, existe uma dificuldade inerente de se obter amostras da classe não anômala (NOK).

Neste capítulo, apresentamos uma técnica aplicada aos dados utilizados no treinamento e no teste do autoencoder, capaz de aumentar seu desempenho como detector de anomalia ou classificador. Demonstramos como, por meio de pré-processamento do sinal antes da construção do respectivo espectrograma, a distância estatística entre as classes OK e NOK aumenta. Tal aumento implica um melhor desempenho do modelo treinado. O diagrama da Figura 79 resume o procedimento proposto. As transformadas (φ) que utilizamos são descritas na Seção 7.1.2.

Figura 79 – Diagrama explicativo do método proposto para aperfeiçoamento do modelo por meio de transformação dos dados. Objetivo: aumentar a separabilidade entre as classes OK e NOK



Fonte: O Autor (2022).

7.1 METODOLOGIA

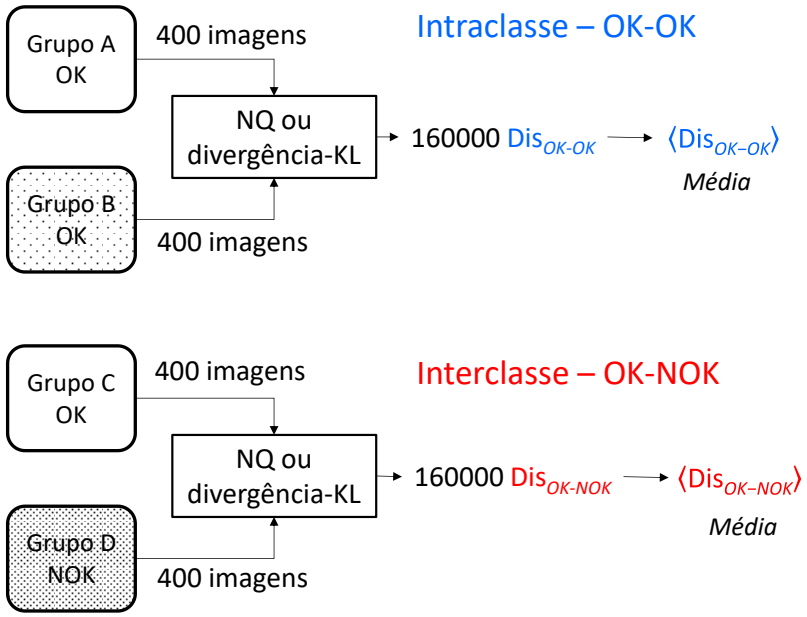
Esta seção apresenta uma visão geral do método proposto. Na Seção 5.2.2, o conceito de pré-processamento dos dados por meio de transformação é empregado para contornar as limitações de se detectarem anomalias usando a assinatura

termográfica do DUT. Especificamente para aquele caso, realizamos a transformação por meio da função $detrend()$. Aqui, testamos transformações adicionais, além expandir o uso da técnica para os casos de detecção de anomalias por assinatura elétrica e a classificação de substâncias. Também definimos o parâmetro “distância estatística” para realizar uma análise quantitativa, que compara os ganhos obtidos com transformações distintas e indica as razões por trás do aumento de desempenho obtido.

7.1.1 Dissimilaridade estatística

Neste trabalho, a dissimilaridade estatística entre duas classes de dados é calculada por meio norma quadrática (Seção 3.6.1) e da divergência-KL (Seção 3.6.2). Com o intuito de inferir a influência das transformações na separação estatística entre as classes OK e NOK, calculamos a dissimilaridade média intraclasse (OK x OK) e interclasse (OK x NOK), conforme ilustrado na Figura 80.

Figura 80 – Diagrama explicativo do método proposto para aperfeiçoamento do modelo por meio de transformação dos dados. Objetivo: aumentar a separabilidade entre as classes OK e NOK



Fonte: O Autor (2022).

As imagens das classes OK e NOK são separadas em quatro grupos de 400 imagens cada. Comparamos 400 imagens de uma classe com 400 imagens da outra classe – ou seja, realizamos 400 x 400 = 160000 comparações para cada distância estatística. Em seguida, calculam-se as médias dos resultados parciais $\langle Dis_{(OK-NOK)} \rangle$

e $\langle Dis_{(OK-OK)} \rangle$. Finalmente, a razão entre essas médias – equação (16) – define a dissimilaridade estatística entre as classes OK e NOK.

$$Dist_{est} = \langle Dis_{(OK-NOK)} \rangle / \langle Dis_{(OK-OK)} \rangle. \quad (16)$$

A separação entre as classes depende da razão (16). Quanto maior que um for $Dist_{est}$, maior é a dissimilaridade entre as classes. A hipótese que consideramos é que uma maior dissimilaridade implica um modelo de detecção mais preciso (Monteiro et al., 2020b).

Em nosso trabalho, tanto a NQ quanto a divergência-KL foram calculados utilizando as bibliotecas do Python e o código executado no *Google Colaboratory* (Nelson, 2020).

7.1.2 Transformação dos dados

Durante a fase preliminar do nosso estudo, testamos vários métodos de pré-processamento, alguns deles propostos em (Abedin et al., 2020), como a função senoidal. Outras transformadas testadas foram: x^3; $sen(x)$; $sen^3(x)$; e $arctan(x)$; No entanto, apenas alguns apresentaram resultados satisfatórios. Assim, em nosso trabalho, consideramos as quatro transformações promissoras descritas abaixo. O objetivo da transformação é aumentar a diferença entre amostras (imagens) pertencentes a duas classes diferentes (espaços vetoriais). Em geral, dados dois vetores $\mathbf{x} \in \mathbb{X}$ e $\mathbf{z} \in \mathbb{Z}$, onde $\mathbb{X}$ e $\mathbb{Z}$ são espaços vetoriais, uma transformada é definida por

$$\mathbf{x} \xrightarrow{\varphi} \mathbf{z}. \quad (17)$$

Ou seja, $\mathbf{z} = \varphi(\mathbf{x})$, onde φ é a transformação aplicada ao vetor $\mathbf{x}$. Nos tópicos I a IV a seguir, descrevemos as transformações usadas neste trabalho.

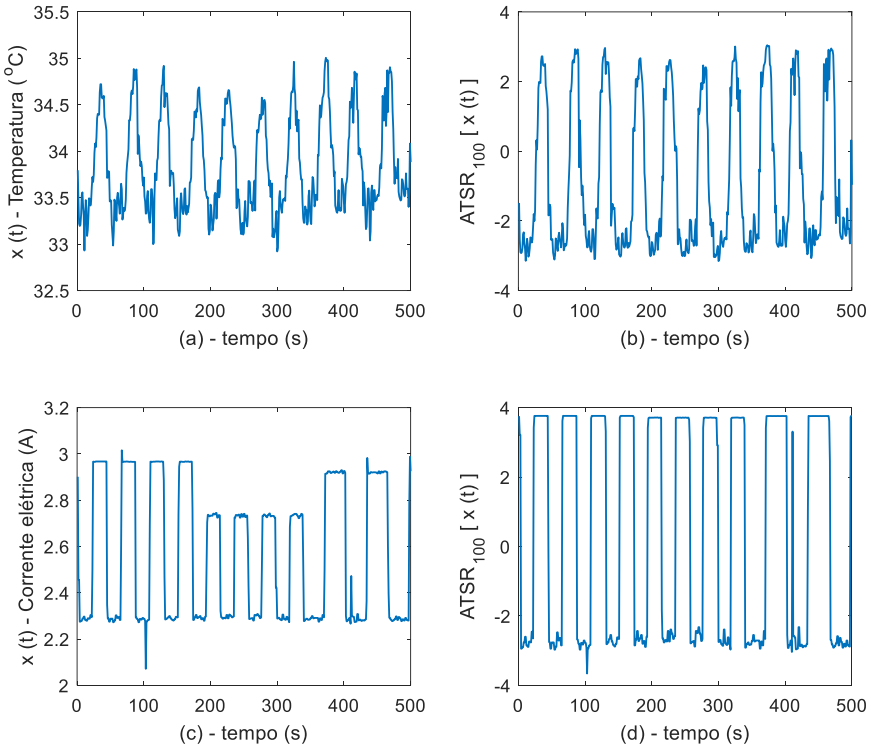
I. *ARC-TANGENT SQUARE ROOT* (ATSR)

ATSR são transformações não lineares usadas no processamento de sinais acústicos (Oo, 2010). Neste trabalho, consideramos a relação (18) para a transformação ATSR:

$$\varphi(\mathbf{x}) = 2{,}5\arctan(\beta\mathbf{x}) + 2{,}5\sqrt{1-(0{,}9\mathbf{x})^2} - 2{,}5, \quad (18)$$

onde, β é um parâmetro que controla a intensidade da não linearidade da transformação. Realizamos testes preliminares com diversos valores de β (1, 10, 100 e 1000), dos quais $\beta = 100$ apresentou os resultados mais promissores. Por isso, nos referimos a essa transformação como $ATSR_{100}$. Na equação (18), devemos assegurar que qualquer valor do vetor $\mathbf{x}$ seja menor $1/0{,}9$ para garantir que $\varphi(\mathbf{x})$ seja real. Portanto, antes da transformação, normalizamos a entrada (i. e., garantimos que $max(\mathbf{x}) = 1, min(\mathbf{x}) = -1$). As curvas na Figura 81 ilustram o comportamento da transformada $ATSR_{100}$ quando aplicada a dois sinais de amostra que adquirimos do DUT em nossos experimentos.

Figura 81 – Exemplos de transformação usando ATSR aplicado a sequências de dados amostrais: (a) e (b) – sinal termográfico; (c) e (d) – sinal elétrico. ATSR modifica o formato do sinal $x(t)$



Fonte: O Autor (2022).

II. DETREND

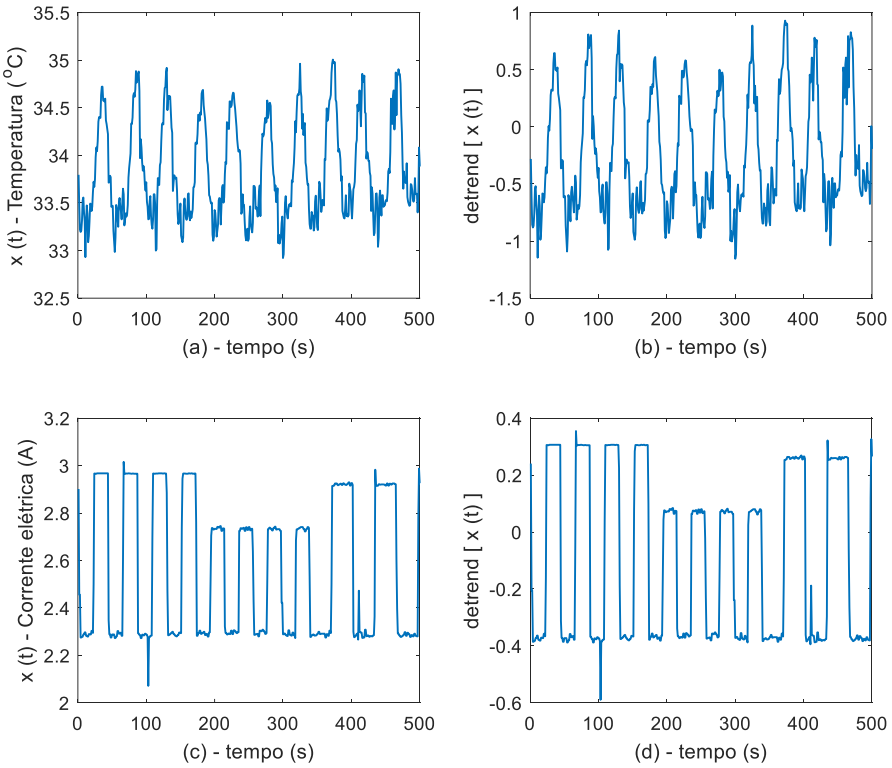
O procedimento $detrend(\mathbf{x})$ (Moore & Sanadhya, 2009) subtrai a média ou uma linha de melhor ajuste (no sentido de mínimos quadrados) do vetor $\mathbf{x}$. Assim, se os dados, apresentarem algum comportamento flutuante, ele pode ser detectado e

removido. Em nosso trabalho, usamos a função $detrend()$ do *Matlab* para realizar essa transformação:

$$\varphi(\mathbf{x}) = detrend(\mathbf{x}). \tag{19}$$

A Figura 82 mostra exemplos da transformada de tendência aplicada a sinais de amostra. Após a transformação, o sinal flutua em torno do eixo horizontal, o que indica que possui valor médio (nível DC) nulo.

Figura 82 – Exemplos de transformação de tendência aplicada a sequências de dados de amostra. (a) e (b) – sinal termográfico; (c) e (d) – sinal elétrico. $Detrend$ suprime o nível DC do sinal



Fonte: O Autor (2022).

III. MODULAÇÃO EM FREQUÊNCIA (FM)

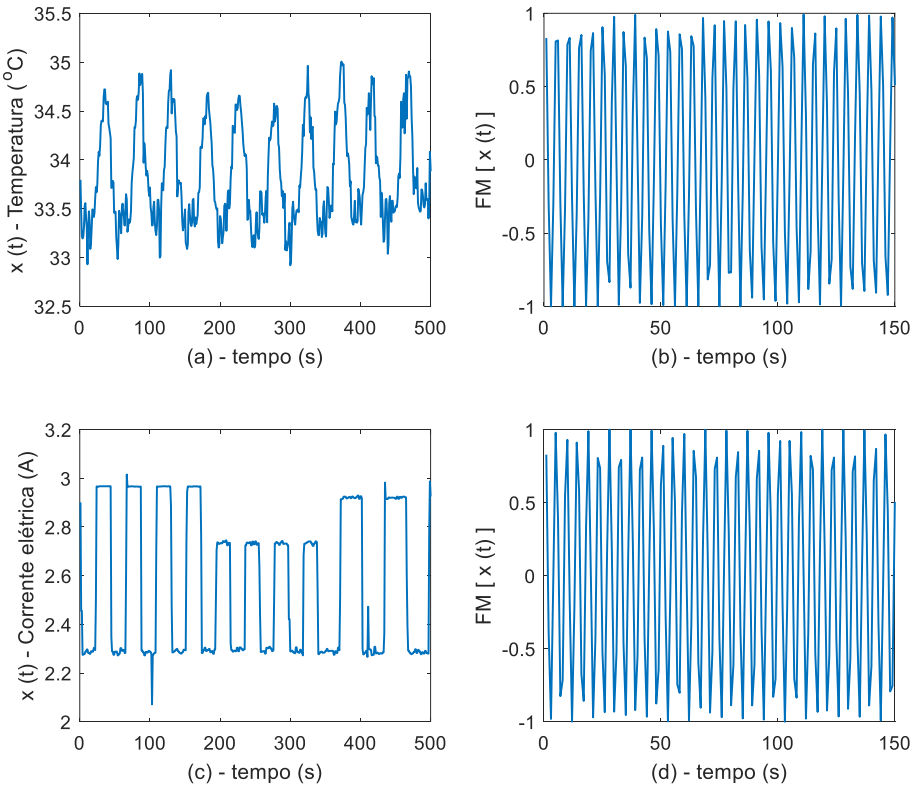
Como estamos usando análise de domínio de tempo e de frequência (ou seja, espectrogramas), uma maneira possível de aumentar a distância entre os dados de duas classes distintas é modificar o conteúdo espectral dos sinais usados para criar os espectrogramas. A modulação de frequência é uma boa maneira de realizar essa modificação de conteúdo espectral. Na FM, a frequência da portadora instantânea é modificada por um sinal de mensagem (Faruque, 2017). A equação (20) define a transformada FM:

$$\varphi(\mathbf{x}) = \cos\left[2\pi f_o t + K_p \int_{-\infty}^{t} \mathbf{x}(t)dt\right]. \quad (20)$$

onde, t é o tempo, f_o é a frequência da portadora e K_p é a constante de modulação, que define o desvio da frequência da portadora. A integral assegura que a frequência instantânea do sinal modulado (portadora) varia com o sinal modulante $x(t)$.

Para obter a modulação em frequência, usamos a função do *Matlab* $fmmod(x, Fc, Fs, freqdev)$, onde x é a mensagem, a portadora tem frequência Fc, Fs é a taxa de amostragem e freqdev é o desvio de frequência. Em nossos experimentos, usamos $Fc = Fs/8$ e $freqdev = 0{,}8Fc$. Na Figura 83, mostramos exemplos de transformação de dados FM usando sequências de sinal de amostra.

Figura 83 – Exemplos de transformação FM aplicada a sequências de dados de amostra. (a) e (b) – sinal termográfico; (c) e (d) – sinal elétrico. A transformada FM aumenta o conteúdo espectral de x(t)



Fonte: O Autor (2022).

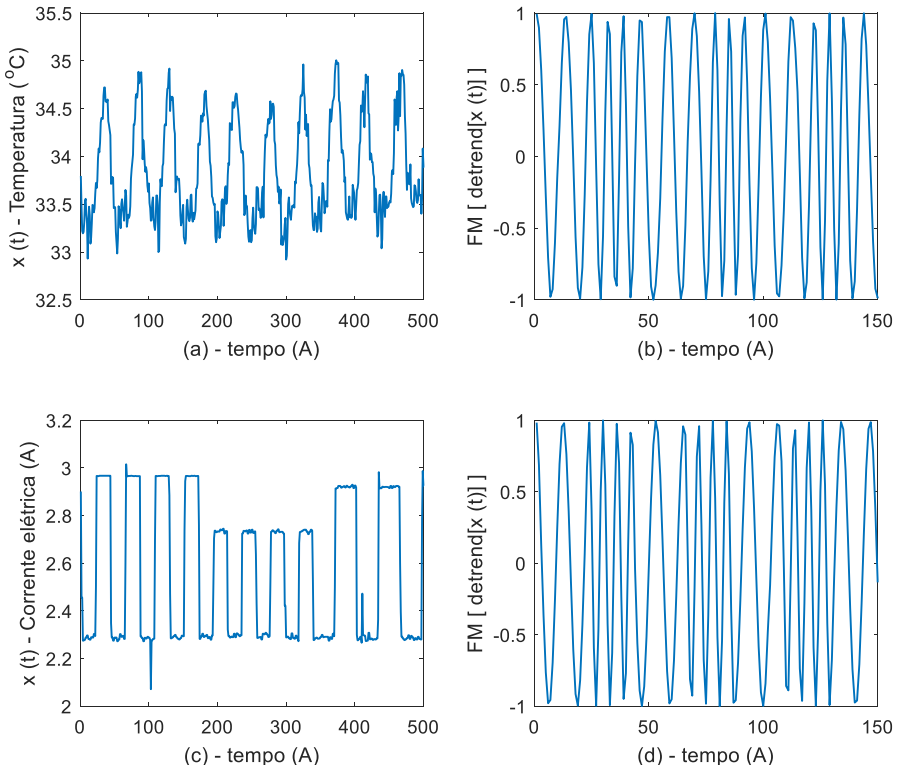
IV. DETREND & FM

Nesse caso, a tendência é removida primeiro e, em seguida, a modulação é realizada usando uma equação similar à (20), i. e.:

$$\varphi(\mathbf{x}) = \cos\left[2\pi f_o t + K_p \int_{-\infty}^{t} detrend[\mathbf{x}(t)]dt\right]. \quad (21)$$

A Figura 84 mostra exemplos de transformada de tendência aplicada a sinais de amostra. O efeito da modulação de frequência nos sinais transformados é mais pronunciado do que na Figura 83.

Figura 84 – Exemplos de transformação Detrend e FM aplicados a sequências de dados de amostra. (a) e (b) – sinal termográfico; (c) e (d) – sinal elétrico. Detrend + transformada FM aumenta a variação espectral ao longo do tempo de x(t)



Fonte: O Autor (2022).

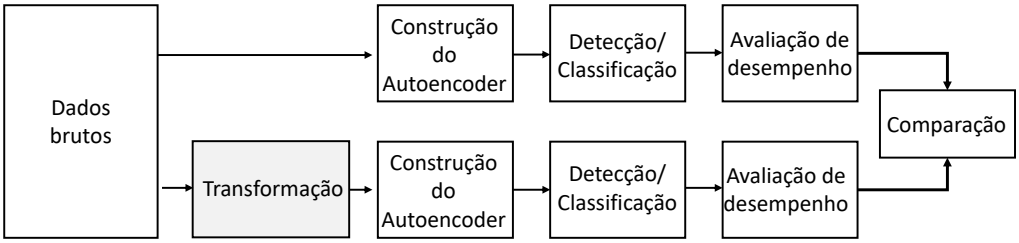
7.2 RESULTADOS

A seguir, são apresentados os resultados obtidos quando se aplica o pré-processamento dos dados. Separamos a apresentação de acordo com o sistema analisado: detecção por assinatura termográfica (Seção 7.2.1); detecção por assinatura de corrente elétrica (Seção 0); e classificação de substâncias (Seção 0).

O diagrama de blocos da Figura 85 ilustra o fluxo de análise que adotamos neste trabalho para processar os dados, construir o autoencoder e avaliar os resultados em termos de desempenho da detecção de anomalias ou da classificação de

substâncias. Os dados brutos correspondem aos sinais obtidos do DUT no caso de detecção de anomalias (Capítulos 4 e 5) ou o sinal óptico detectado, no caso da classificação de substâncias (Capítulo 6). Em uma abordagem tradicional, sem o pré-processamento, essas amostras são convertidas em espectrogramas enviados ao autoencoder, seja para treinamento ou para detecção/classificação. Propomos transformar as amostras – usando as transformadas descritas na Seção 7.1.2 – antes de construir e testar os espectrogramas.

Figura 85 – Fluxo de análise com e sem pré-processamento dos dados



Fonte: O Autor (2022).

Ao final do processo, avaliamos os resultados obtidos comparando o desempenho de detecção em ambos os casos: com e sem pré-processamento.

7.2.1 Detecção de anomalias por assinatura elétrica

Para este caso, vários comprimentos de janela foram testados (Seção 4.3.2). Entretanto, a janela ΔT_1750 se destaca pelo fato de não ser possível treinar adequadamente o autoencoder, como mostra o resultado da Figura 46 para essa janela. Para contornar esses resultados limitados, propomos a aplicação das transformações descritas no Capítulo 7 às amostras listadas na Tabela 15. Cada teste foi repetido 15 vezes. Ou seja, repetimos o treinamento de modo que a variância dos resultados, graficamente observada por meio de gráficos do tipo *boxplot* ou com barra de erro, fosse limitada. Portanto, a variância das taxas de erro e demais parâmetros é tal que, cada resultado pode ser estatisticamente comparado aos demais. Note que aqui, diferentemente do Capítulo 4, renomeamos a janela de largura 1750 ms para $1750\_Original$, uma forma de distinguir essa janela sem transformação das demais.

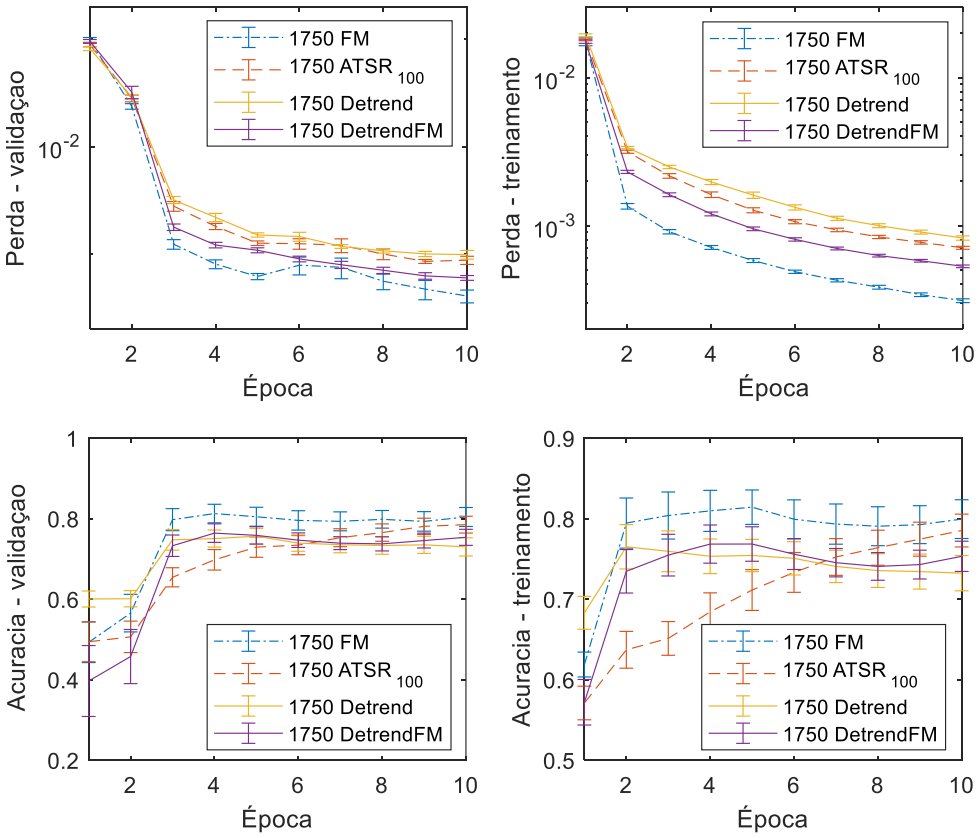
Tabela 15 – Designação e quantidade de amostras de corrente elétrica geradas para cada janela

ΔT (ms)	***Designação***	**Transformação**	**#treinamento**	**#validação**	**#teste**
1750	*1750_Original*	Nenhuma	9600	2400	2000
1750	*1750_Detrend*	Detrend	9600	2400	2000
1750	*1750_ FM*	FM	9600	2400	2000
1750	*1750_ ATSR$_{100}$*	ATSR$_{100}$	9600	2400	2000
1750	*1750_Detrend_FM*	Detrend & FM	9600	2400	2000

Fonte: O Autor (2022).

Diferentemente da janela $1750\_Original$, após a transformação, o treinamento se torna possível. Isso pode ser observado nas curvas da Figura 86, onde são exibidos os resultados de validação e treinamento para as janelas após as transformações. O eixo das perdas está em escala logarítmica para facilitar a visualização.

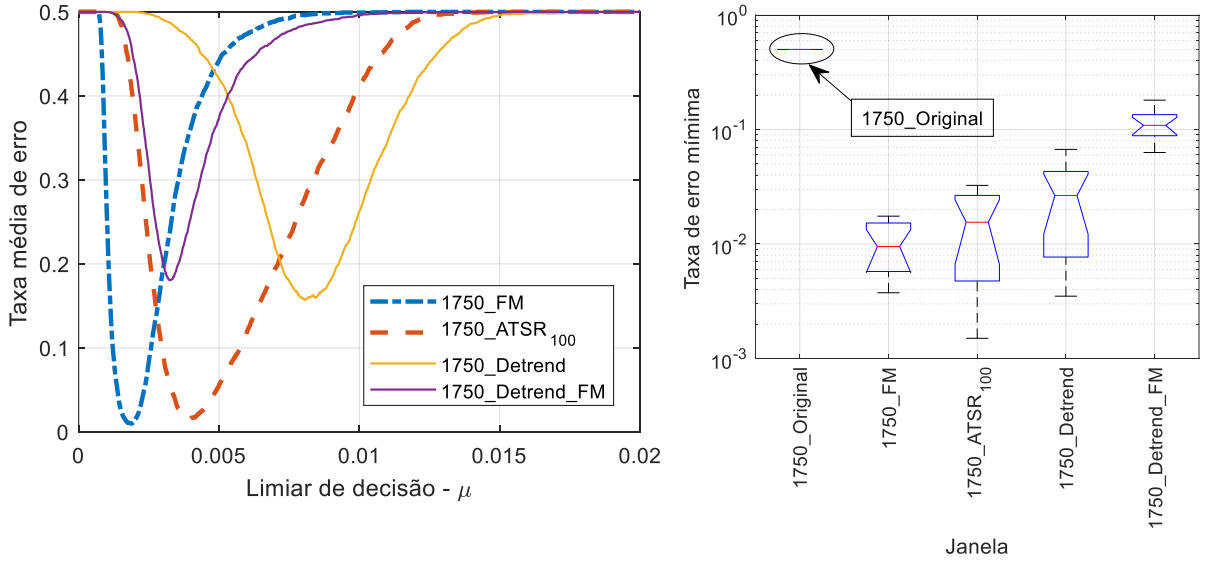
Figura 86 – Validação e treinamento do autoencoder com barras de erro para as janelas testadas - caso corrente elétrica



Fonte: O Autor (2022).

Após o treinamento, testamos a detecção de anomalias para cada transformada. Primeiro, definimos um valor para o limiar de detecção e obtemos os valores de cada elemento da matriz de confusão (Tabela 1 da Seção 3.7.2). A partir desses elementos, calculamos a taxa de erro de detecção, e os parâmetros FPR e TPR. Os resultados são apresentados na Figura 87. Aplicando as transformações propostas, os resultados indicam uma situação diferente daquela encontrada na Seção 4.3.1, onde o modelo sem transformação não pode ser treinado corretamente. Dentre as transformações consideradas, o melhor desempenho é obtido pela modulação em frequência (FM), pois apresenta a menor taxa de erro. Por outro lado, o erro para a janela $1750\_Original$ fica bem acima do padrão geral de valores no gráfico.

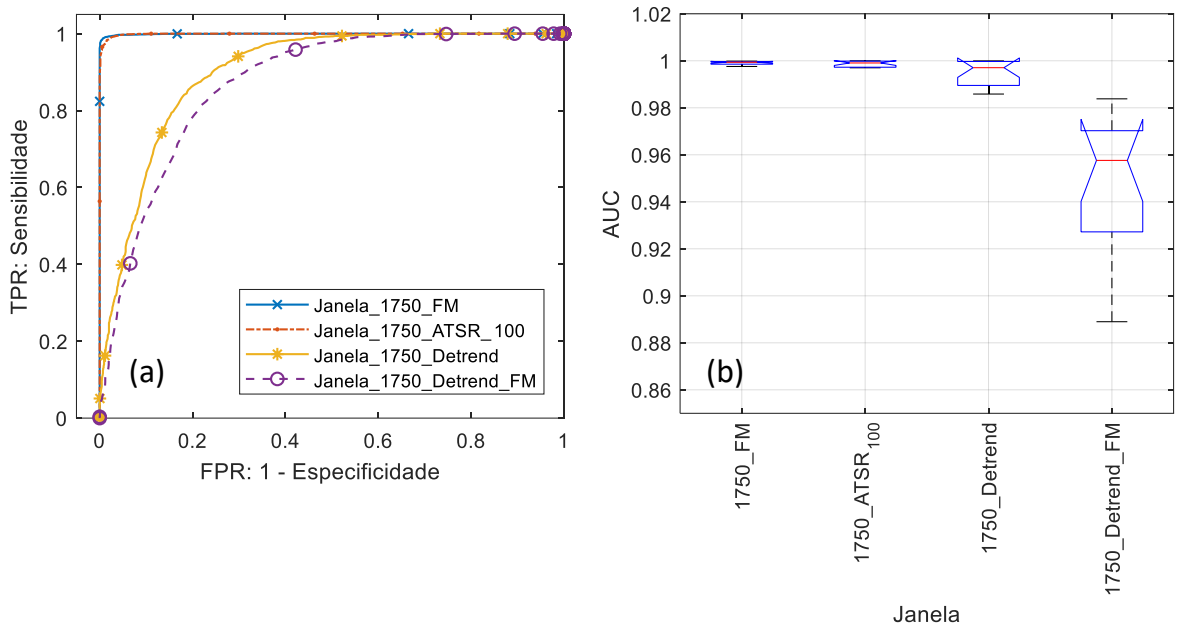
Figura 87 – (a) - Taxa mínima de erro em função do limiar de decisão (média de 15 experimentos para cada uma das janelas de amostras transformadas); (b) - Taxa mínima de erro para as janelas original e transformadas (*boxplot* considerando os 15 testes)



Fonte: O Autor (2022).

Seguindo a metodologia de análise do desempenho para sistemas de detecção de anomalias, podemos comprovar os ganhos obtidos com a aplicação das transformações por meio das curvas ROC e da AUC da Figura 88. A janela $1750\_Original$ não apresenta resultado aceitável, por isso não foi adicionada ao gráfico.

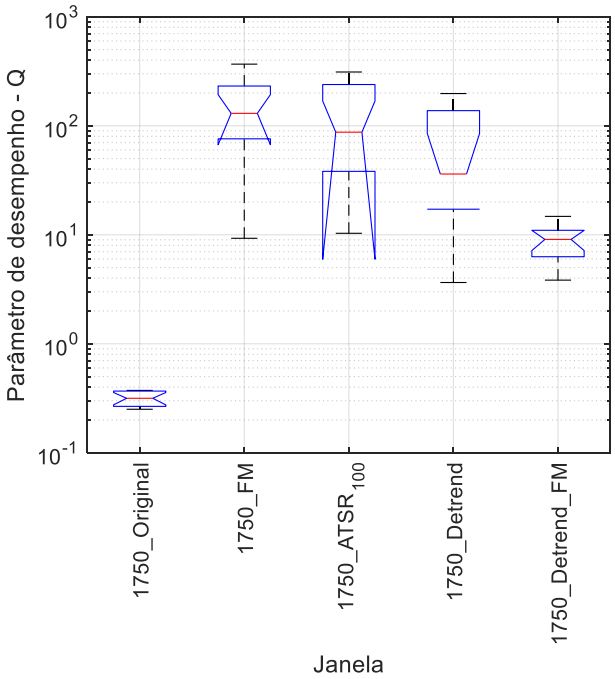
Figura 88 – (a) – Curvas ROC médias para os 15 testes realizados; (b) - AUC (*boxplot* considerando os 15 testes)



Fonte: O Autor (2022).

Para finalizar, usamos o parâmetro de desempenho para averiguar o nível de melhoramento que as transformações trazem ao modelo. Os resultados da Figura 89 confirmam que o uso da janela 1750_FM leva a um modelo com melhor desempenho.

Figura 89 – *Boxplot* do parâmetro de desempenho Q. Caso de corrente elétrica. Média de 15 testes



Fonte: O Autor (2022).

Um resumo dos resultados obtidos com o emprego das transformações está listado na Tabela 16 com o objetivo de facilitar a comparação numérica entre as janelas analisadas nos testes. Em destaque a janela 1750_FM que possui o melhor desempenho, considerando todos os índices de avaliação.

Tabela 16 – Valores médios para os índices de desempenho considerando 15 repetições

	1750_Original	**1750_FM**	**1750_ATSR$_{100}$**	**1750_Detrend**	**1750_Detrend FM**
Taxa de erro mínima	0,5	0,0095	0,0155	0,157	0,1805
Área sob a curva - AUC	0,5	0,99937	0,99906	0,99076	0,95761
Parâmetro de desempenho - Q	0,32	130,0	87,3	36,1	9,07

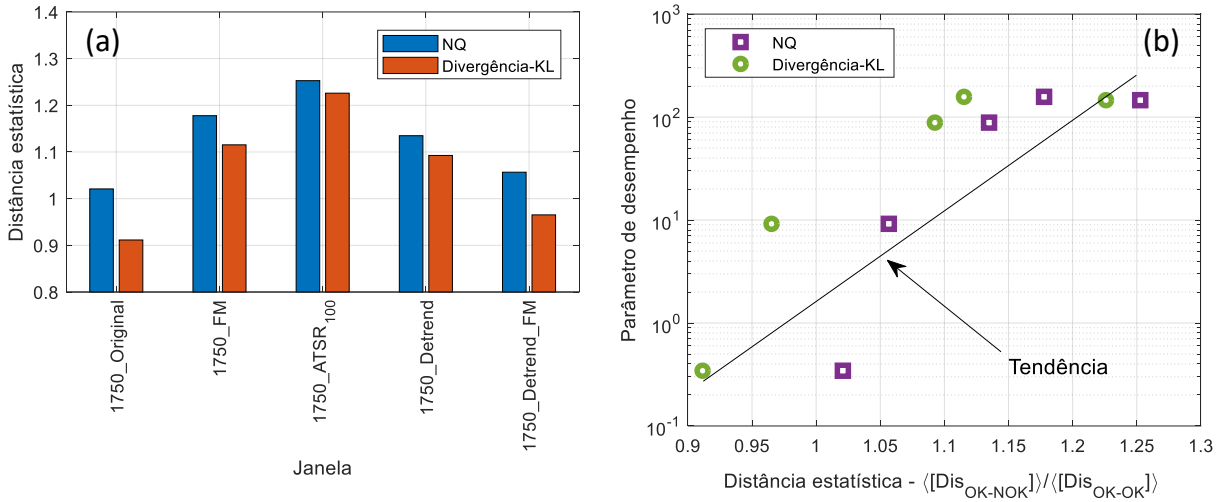
Fonte: O Autor (2022).

Por fim, analisamos a dissimilaridade estatística para janelas com e sem transformação. Os resultados são exibidos na Figura 90. Na Figura 90-a, percebemos que sem transformação, a distância estatística é próxima da unidade ($\langle Dis_{(OK-NOK)}\rangle / \langle Dis_{(OK-OK)}\rangle \cong 1$). Nesse caso, o autoencoder não consegue distinguir imagens de diferentes classes, como é o caso para a janela $1750\_Original$. Por outro lado, as transformações levam a um aumento da distância entre as classes OK e NOK ($\langle Dis_{(OK-NOK)}\rangle / \langle Dis_{(OK-OK)}\rangle > 1$). Isso, por sua vez, leva a uma melhoria da precisão. Dos resultados da Figura 90-b, é possível perceber que há uma relação direta entre dissimilaridade e desempenho do sistema de detecção de anomalias. A linha tracejada representa a tendência observada: Q aumenta com a distância estatística.

Comparando os resultados da Tabela 16 com a Figura 90-a, notamos que a transformada FM tem o melhor desempenho, apesar da maior razão de distância para a ATSR$_{100}$. De fato, a forte não linearidade da transformada ATSR$_{100}$ leva a um aumento mais pronunciado na diferença estatística interclasse do que a transformada FM. No entanto, treinar o autoencoder é um pouco mais eficaz com as imagens obtidas com FM do que com os da transformada ATSR$_{100}$ (ver a Figura 86). Portanto, concluímos que a FM apresenta melhor desempenho na detecção de anomalias devido ao seu melhor desempenho no treinamento do autoencoder quando comparado ao ATSR$_{100}$. Por outro lado, a transformação conjunta Detrend & FM não apresenta melhoria do mesmo nível que as outras. Isso pode ser explicado

pela baixa separação entre as classes OK e NOK para os dados obtidos desta transformação, conforme mostra a Figura 90-a ($\langle Dis_{(OK-NOK)}\rangle/\langle Dis_{(OK-OK)}\rangle \cong 1$).

Figura 90 – (a) - Distâncias estatísticas entre as classes NOK e OK $\langle Dis_{(OK-NOK)}\rangle/\langle Dis_{(OK-OK)}\rangle$ para a janela original e as janelas transformadas. (b) - Parâmetro de desempenho médio de detecção em função das distâncias estatísticas



Fonte: O Autor (2022).

7.2.2 Detecção de anomalias por assinatura termográfica

Para este caso, a janela de aquisição tem um comprimento fixo ($\Delta T = 6000\, ms$). Após adquirir os dados brutos, ou seja, as amostras termográficas, aplicamos a transformação antes de gerar os espectrogramas. Em relação à quantidade de dados, foram gerados um total de 1204 amostras. As amostras obtidas foram separadas em três grupos. Usamos 642 e 160 amostras para treinamento e validação cruzada, respectivamente. As 401 amostras restantes foram usadas para testar o autoencoder. O nome da janela, o número de imagens e o respectivo comprimento da janela para cada caso são indicados na Tabela 17. Como neste caso termográfico usamos apenas amostras de mesma duração, a designação das janelas não contém essa informação, diferentemente do caso da assinatura elétrica.

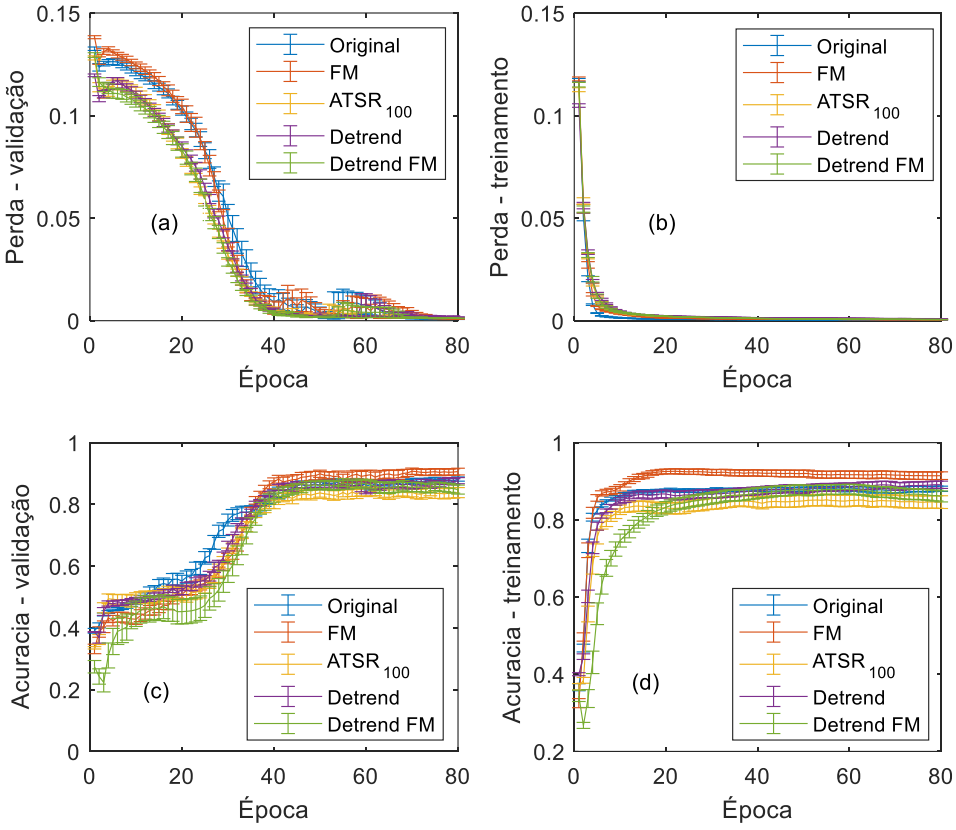
Tabela 17 – Designação e quantidade de amostras de sinais termográficos geradas para cada janela

ΔT (ms)	***Designação***	**Transformação**	**#treinamento**	**#validação**	**#teste**
6000	*Original*	Nenhuma	642	160	401
6000	*Detrend*	Detrend	642	160	401
6000	*FM*	FM	642	160	401
6000	$ATSR_{100}$	$ATSR_{100}$	642	160	401
6000	*Detrend_FM*	Detrend & FM	642	160	401

Fonte: O Autor (2022).

Para cada transformação e para os dados brutos, o procedimento de treinamento/validação foi repetido 50 vezes. Ou seja, repetimos o treinamento de modo que a variância dos resultados, graficamente observada por meio de gráficos do tipo *boxplot* ou com barra de erro, fosse limitada. Portanto, a variância das taxas de erro e demais parâmetros é tal que, cada resultado pode ser estatisticamente comparado aos demais. A Perda e a acurácia médias de treinamento/validação são exibidas com barras de erro na Figura 91. As curvas confirmam que o autoencoder foi treinado adequadamente, independentemente do pequeno número de amostras.

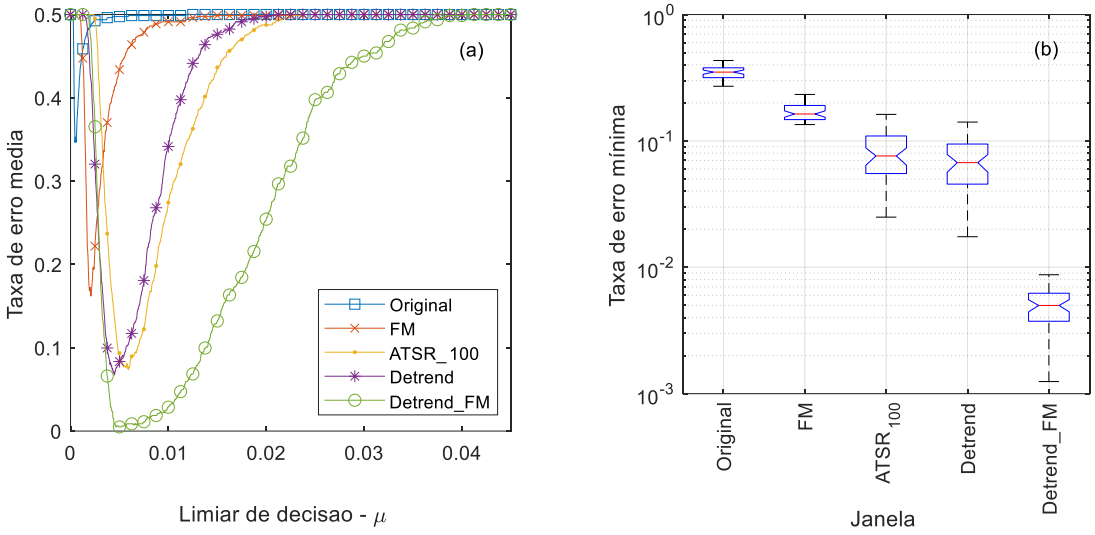
Figura 91 – Resultados de treinamento para caso termográfico com transformações. (a) e (b) - Perda de validação e treinamento; (c) e (d) - Acurácia de validação e treinamento. Média dos 50 testes



Fonte: O Autor (2022).

Após o treinamento do autoencoder, testamos seu desempenho como detector de anomalias. A seguir os principais resultados são discutidos. A Figura 92 ilustra os resultados médios de 50 testes realizados para cada caso. A partir dos resultados, observamos que os dados brutos não oferecem condições adequadas para detecção de anomalias, uma vez que a taxa de erro é bastante elevada. Por outro lado, todos os casos transformados mostram melhoria de desempenho. Conforme explicado na Seção 5.2.2, o uso da transformação Detrend serviu de inspiração em nosso estudo para a busca por outras possibilidades. Após os testes realizados com outras transformadas, encontramos uma que apresentou resultados ainda melhores. A combinação Detrend & FM, por exemplo, apresenta uma taxa de erro média mínima abaixo de 0,5%. Para visualizar as variações obtidas dentro dos 50 testes realizados, plotamos o gráfico *boxplot* da taxa de erro mínima na Figura 92-b. Mesmo com as variações observadas nos testes, é possível concluir que as transformações melhoram a qualidade do sistema de detecção. Confirma-se também que a transformada combinada Detrend & FM supera as demais de maneira estatisticamente relevante.

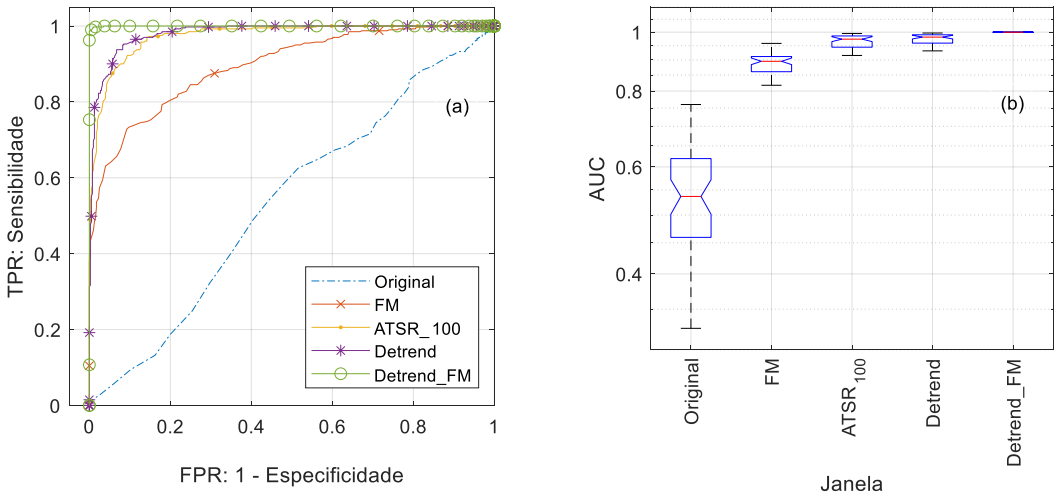
Figura 92 – Taxa mínima de erro em função do limiar de decisão (média de 50 experimentos para cada janela); (b) - Taxa mínima de erro para cada janela (*boxplot* considerando os 50 testes)



Fonte: O Autor (2022).

Outra forma de visualizar os resultados é por meio das curvas ROC e da área sob a curva (AUC). Esses resultados são ilustrados na Figura 93. A linha aproximadamente diagonal da Figura 93-a e a AUC correspondente igual a aproximadamente 0,5 na Figura 93-b confirmam que o uso dos dados brutos não permite a detecção de anomalias. Já o panorama para os dados transformados confirma a efetividade do método, sendo a transformada Detrend & FM a que apresenta o melhor resultado: curva ROC com a maior AUC.

Figura 93 – Curvas ROC médias para os 50 testes realizados; (b) - AUC (*boxplot* considerando os 50 testes)

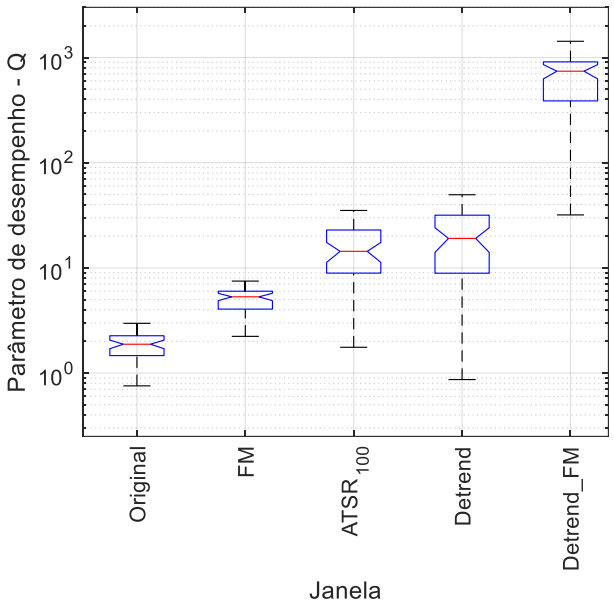


Fonte: O Autor (2022).

Os resultados para taxa de erro mínima e AUC confirmam Detrend & FM como melhor opção. Contudo, além disso, nota-se da Figura 92-a que para a janela

$Detrend\_FM$, o intervalo do limiar de decisão para o qual o erro permanece baixo é mais proeminente do que para todos os outros casos. Então, se considerarmos esse aspecto de análise mais amplo, a transformação combinada Detrend & FM apresenta uma melhoria ainda mais relevante. Quantitativamente, podemos usar o parâmetro Q para confirmar essa vantagem. O parâmetro Q para as janelas consideradas está mostrado na Figura 94 e confirma a superioridade da transformada Detrend & FM.

Figura 94 – *Boxplot* do parâmetro de desempenho Q. Caso termográfico. Média dos 50



Fonte: O Autor (2022).

Um resumo dos resultados obtidos com o emprego das transformações está listado na Tabela 18 com o objetivo de facilitar a comparação numérica entre as janelas analisadas nos testes. Em destaque a janela $Detrend\_FM$ que possui o melhor desempenho, considerando todos os índices de avaliação.

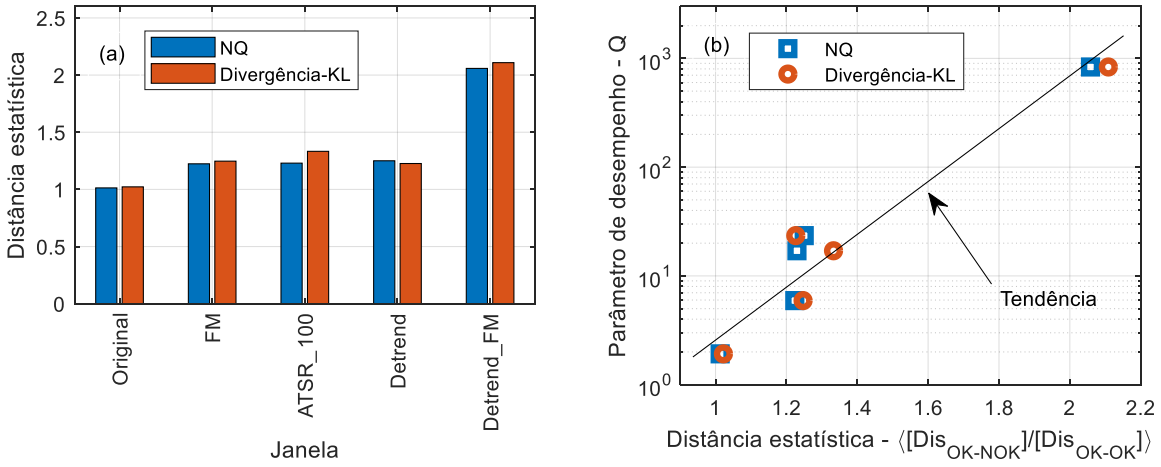
Tabela 18 – Valores médios para os índices de desempenho considerando 50 repetições do teste

	Original	**FM**	**$ATSR_{100}$**	**Detrend**	**Detrend FM**
Taxa de erro mínima	0,3504	0,1633	0,076	0,0673	0,00498
Área sob a curva - AUC	0,536	0,8955	0,974	0,9815	0,9998
Parâmetro de desempenho - Q	1,88	5,3	14,36	19,06	741,3

Fonte: O Autor (2022).

Após a transformação, o aumento no desempenho para os dados transformados pode ser justificado pelo aumento da distância estatística entre as diferentes classes. Usando equação (16) da Seção 7.1.1, podemos visualizar a relação entre o desempenho (Q) e a distância entre as classes OK e NOK para cada transformação. Na Figura 95-a, observamos que um aumento na dissimilaridade é observado para todas as transformadas ($\langle Dis_{(OK-NOK)}\rangle/\langle Dis_{(OK-OK)}\rangle > 1$), quando comparadas com a janela $Original$ ($\langle Dis_{(OK-NOK)}\rangle/\langle Dis_{(OK-OK)}\rangle \cong 1$). Ainda na Figura 95-a, nota-se o destaque para a transformada Detrend & FM, em concordância com os resultados da Tabela 18. Na Figura 95-b, plotamos o parâmetro Q em função das distâncias estatísticas para todos os casos. Aqui, notamos uma tendência de Q aumentar coma dissimilaridade, o que é consistente com o aumento de desempenho observada para cada transformada.

Figura 95 – a - Distância estatística entre as classes NOK e OK $\langle Dis_{(OK-NOK)}\rangle/\langle Dis_{(OK-OK)}\rangle$ antes e depois da transformação. b - Parâmetro de desempenho médio em função das distâncias estatísticas. A linha tracejada representa a tendência observada: há uma tendência de Q aumentar em função da distância estatística



Fonte: O Autor (2022).

7.2.3 Classificação de substâncias

Pelos resultados apresentados na Figura 74, pode-se concluir que os sinais emitidos com número de comprimentos de onda ópticos iguais a dois, três, cinco e seis ($N_\lambda = 2, 3, 5$ e 6) não geram espectrogramas de classes OK e NOK com separação suficiente para atingir desempenho adequada na classificação da substância. Por esta razão, escolhemos esses casos para aplicar as transformações antes de gerar os espectrogramas que, por sua vez, são usados para treinar e testar

o autoencoder classificador. Por uma questão de simplificação e economia de tempo, apenas as transformadas FM e Detrend & FM foram utilizadas neste estudo. Essas janelas apresentaram os melhores resultados em uma análise preliminar.

As designações das janelas analisadas remetem ao número de comprimento de ondas do sinal emitido e à transformada. As janelas estão listadas na Tabela 19 juntamente com sua designação, quantidade e utilização no experimento.

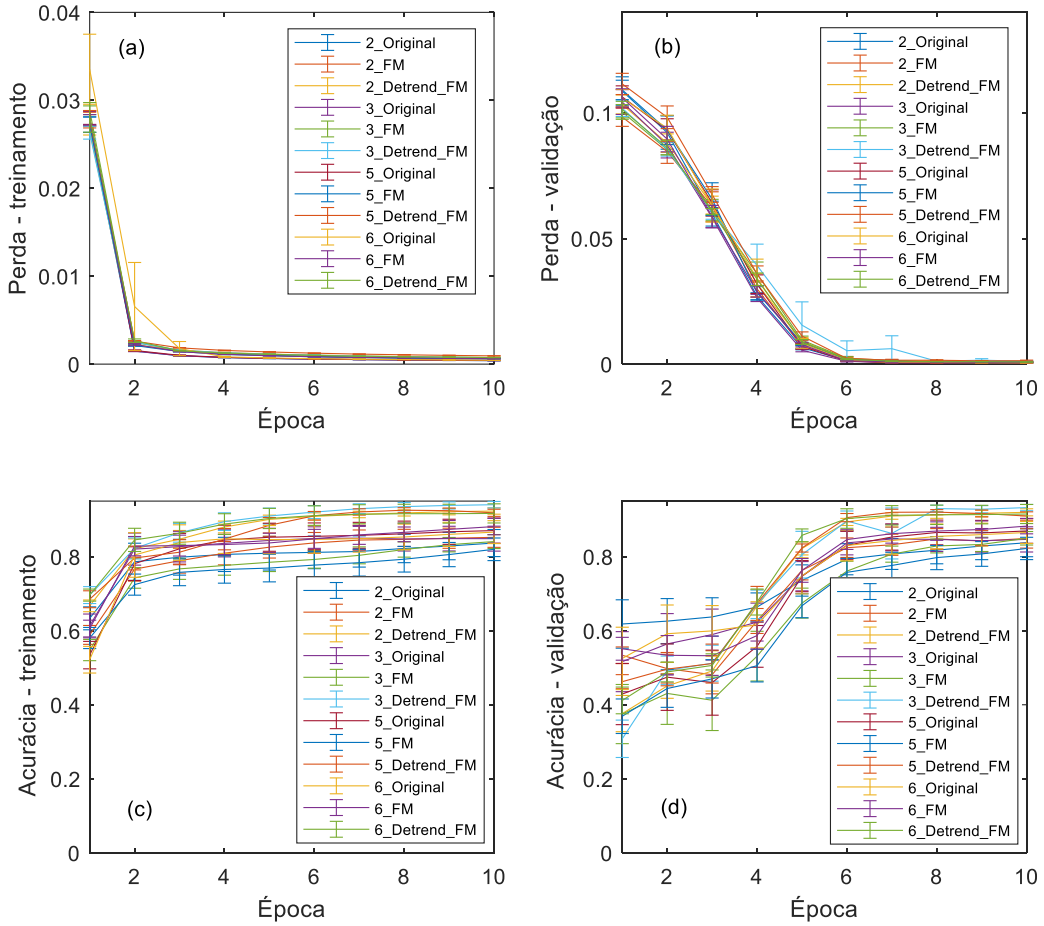
Tabela 19 – Designações, quantidades e uso das amostras geradas para cada janela analisada

Designação	**Transformada**	**N_λ**	**#treinamento (OK)**	**#validação (OK)**	**#teste (OK e NOK)**
2_Original	Nenhuma	2	4800	1200	4000
3_Original		3			
5_Original		5			
6_Original		6			
2_FM	FM	2			
3_FM		3			
5_FM		5			
6_FM		6			
2_Detrend_FM	Detrend & FM	2			
3_Detrend_FM		3			
5_Detrend_FM		5			
6_Detrend_FM		6			

Fonte: O Autor (2022).

Para assegurar que os resultados sejam estatisticamente relevantes, treinamos o e testamos autoencoder 21 vezes. Os resultados do treinamento são exibidos na Figura 96. Ou seja, repetimos o treinamento de modo que a variância dos resultados, graficamente observada por meio de gráficos do tipo *boxplot* ou com barra de erro, fosse limitada. Portanto, a variância das taxas de erro e demais parâmetros é tal que, cada resultado pode ser estatisticamente comparado aos demais. As curvas confirmam a efetividade do processo de treinamento em todos os casos.

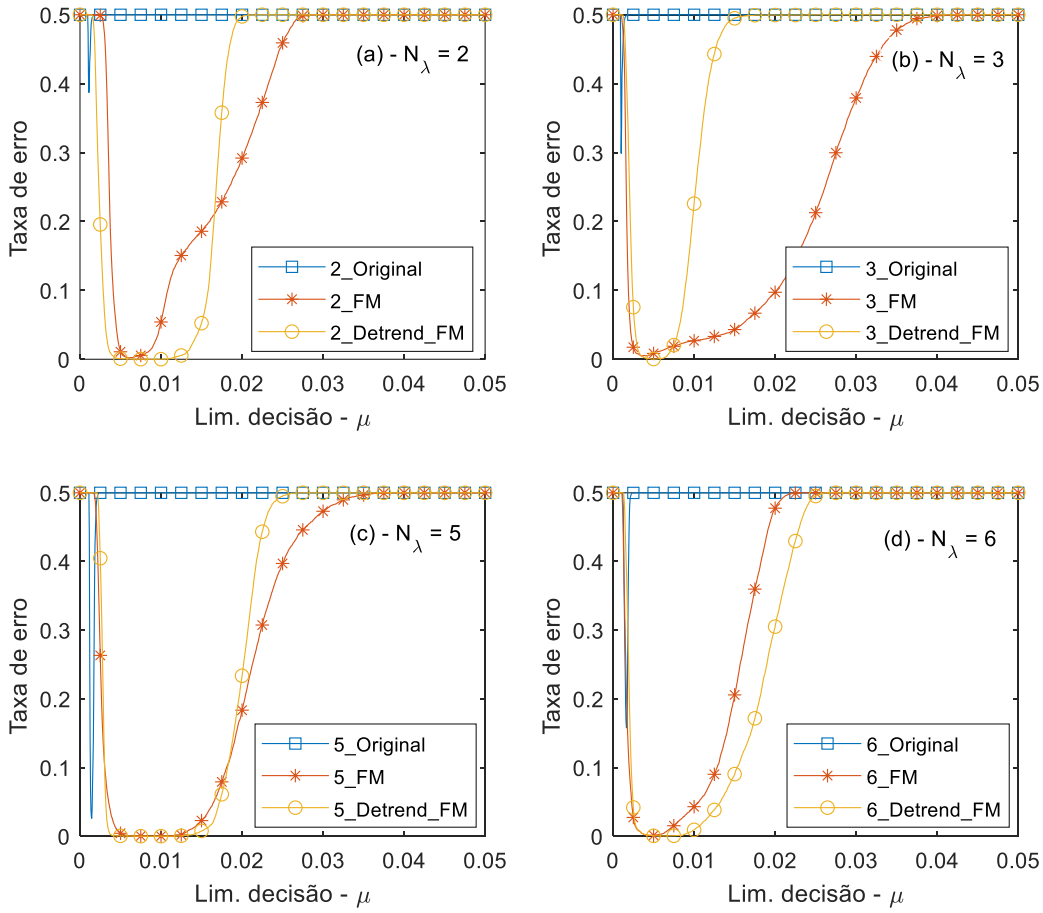
Figura 96 – Resultados de treinamento com e sem transformações. (a) e (b) - Perda de validação e treinamento; (c) e (d) - Acurácia de validação e treinamento. Média dos 21 testes



Fonte: O Autor (2022).

Os testes do autoencoder treinado foram repetidos 21 vezes neste caso. Os resultados estão mostrados na Figura 97.

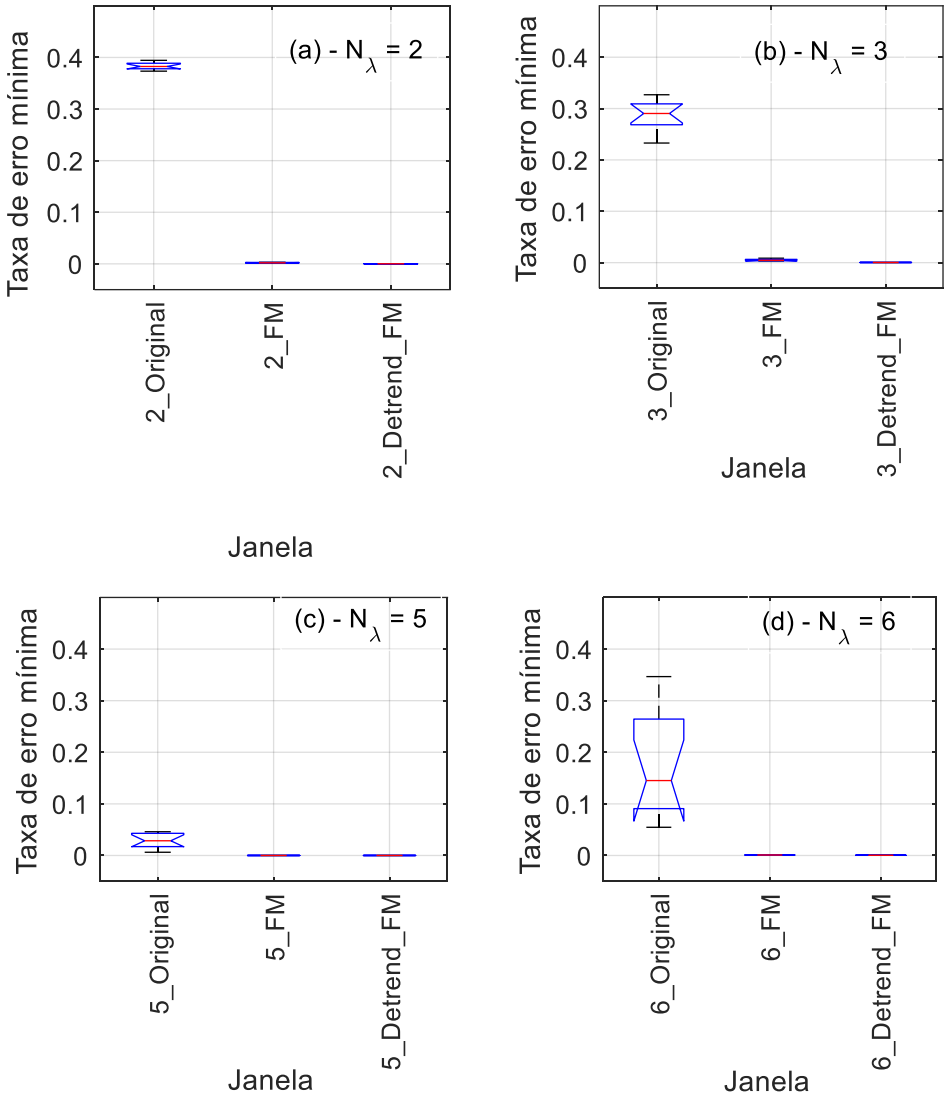
Figura 97 – Taxa de erro média para as janelas original e transformadas em função do limiar de decisão. Média dos 21 testes



Fonte: O Autor (2022).

Considerando os casos transformados (independentemente do valor de N_λ), não é possível distinguir o desempenho entre eles. Pois, a taxa de erro mínima é praticamente nula para todas as janelas. Isso também fica evidente observando-se a taxa mínima de erro média apresentada para cada caso na Figura 98.

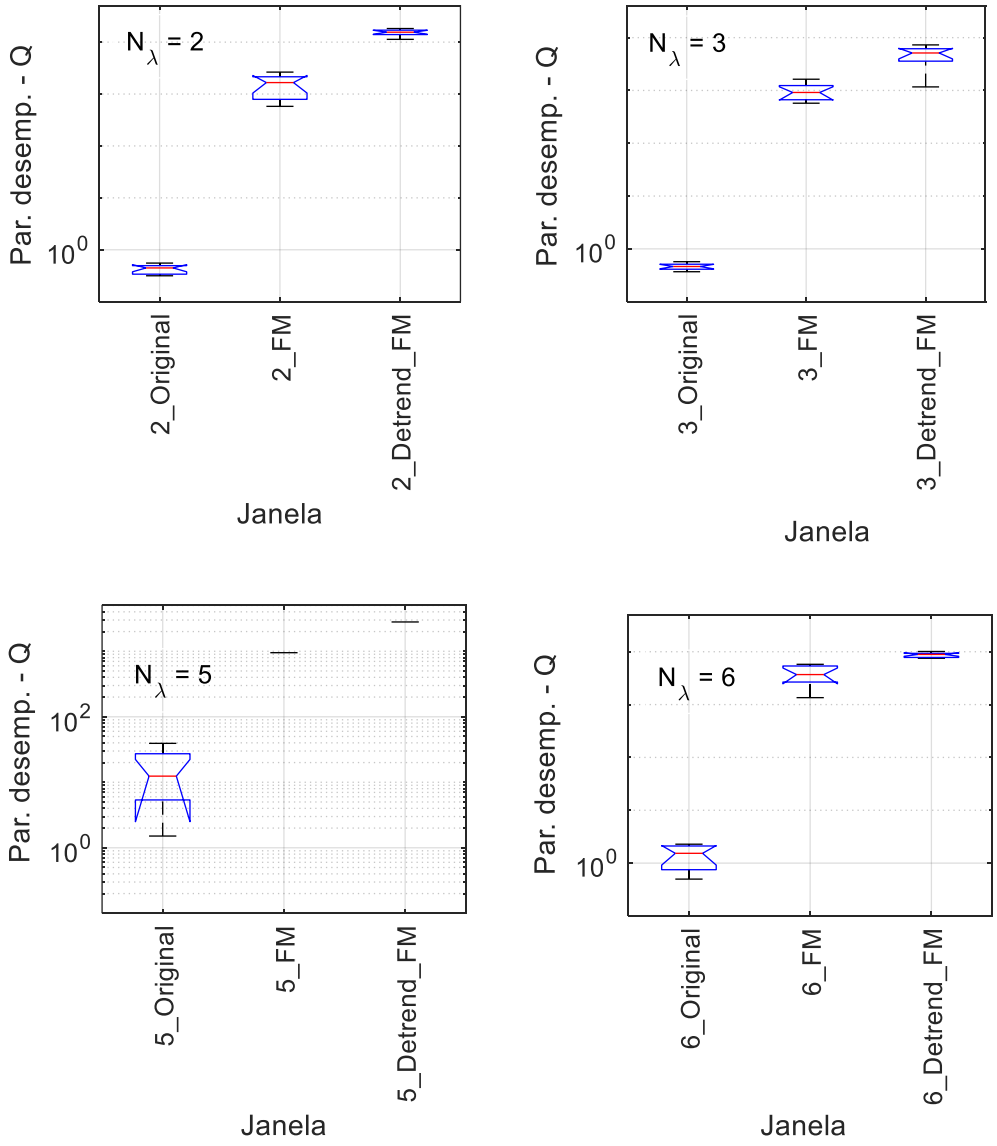
Figura 98 – Taxa de erro mínima para as janelas testadas. *Boxplot* dos 21 testes



Fonte: O Autor (2022).

Entretanto, se analisarmos o parâmetro de qualidade Q traçado na Figura 99, é possível notar uma pequena vantagem em direção à transformada Detrend & FM em todos os casos. De fato, as taxas de erro mínima para as janelas com transformação Detrend & FM são ligeiramente menores. Logo, a equação (14) resulta em um valor mais elevado para Q, apesar da aparente “maior largura” das curvas para os casos $N_\lambda = 3$ e 5 usando a transformada FM (Figura 97-b e c).

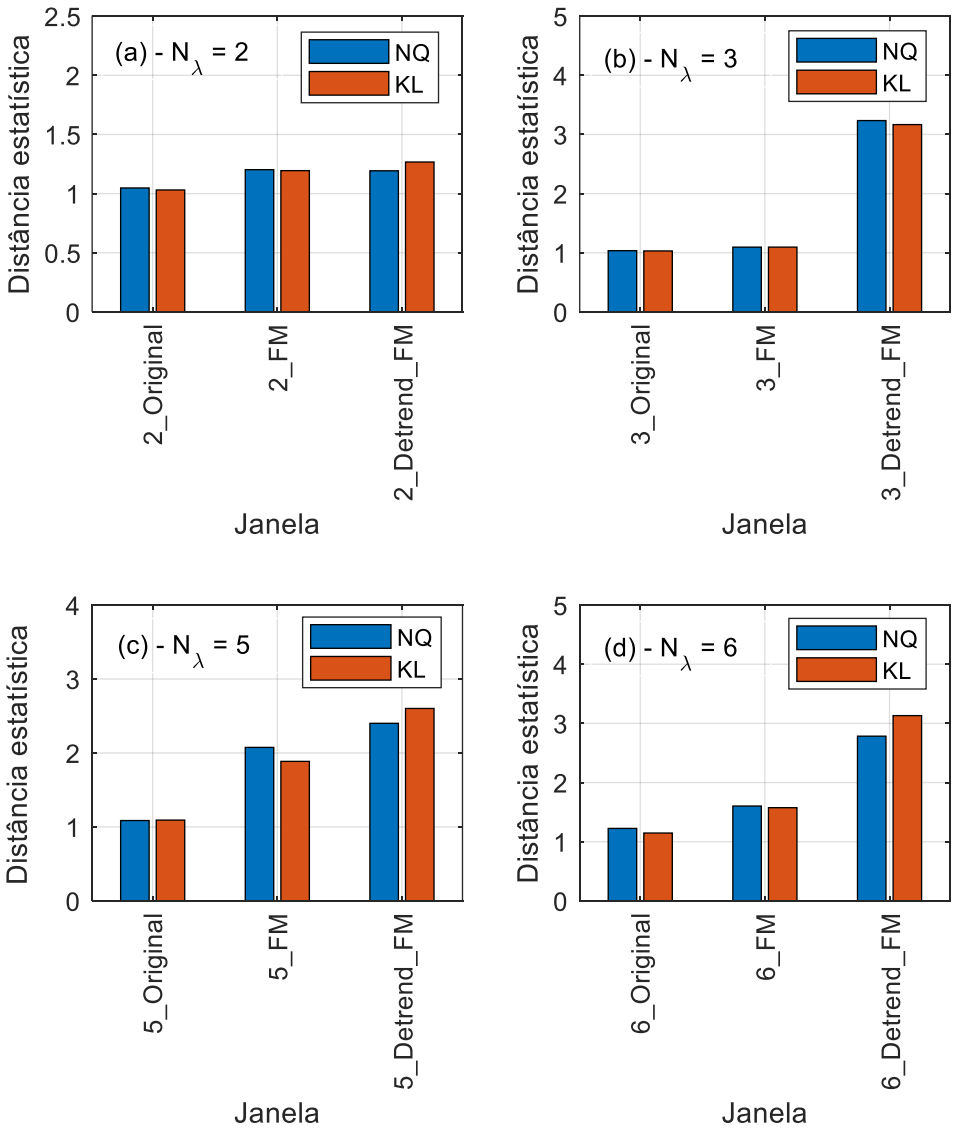
Figura 99 – Parâmetro de desempenho para as janelas testadas. *Boxplot* dos 21 testes



Fonte: O Autor (2022).

Em relação à análise por dissimilaridade estatística para janelas com e sem transformação, os resultados são exibidos na Figura 100. Em todos os casos (a, b, c e d), percebemos que sem transformação (2,3,5,6_$Original$), a distância estatística é próxima da unidade ($\langle Dis_{(OK-NOK)}\rangle/\langle Dis_{(OK-OK)}\rangle \cong 1$). Portanto, o autoencoder não consegue distinguir adequadamente as imagens de classes diferentes. Por outro lado, as transformações levam a um aumento da distância entre as classes OK e NOK ($\langle Dis_{(OK-NOK)}\rangle/\langle Dis_{(OK-OK)}\rangle > 1$). Isso, por sua vez, viabilizam o bom desempenho do autoencoder como classificador.

Figura 100 – Distâncias estatísticas entre as classes NOK e OK $\langle Dis_{(OK-NOK)} \rangle / \langle Dis_{(OK-OK)} \rangle$ antes e depois da transformação para várias configurações do sinal emitido (N_λ). KL denota a divergência-KL



Fonte: O Autor (2022).

Para facilitar a visualização e a comparação, um resumo dos resultados desta seção está indicado na Tabela 20. Em azul, os melhores resultados.

Tabela 20 – Valores médios para os índices de desempenho considerando 21 testes

Designação	**Transformada**	N_λ	**Taxa de erro**	**Par. desempenho**
2_Original	Nenhuma	2	0,3825	0,44
3_Original		3	0,29	0,46
5_Original		5	0,0285	12,41
6_Original		6	0,1452	1,53
2_FM	FM	2	0,002	1661
3_FM		3	0,0046	914
5_FM		5	0,0	950
6_FM		6	0,00093	3698
2_Detrend_FM	Detrend & FM	2	0,000125	15451
3_Detrend_FM		3	0,00031	5121
5_Detrend_FM		5	0,0	2802
6_Detrend_FM		6	0,00037	8890

Fonte: O Autor (2022).

7.3 CONSIDERAÇÕES ESPECIAIS

Nesta seção, tecemos alguns comentários adicionais em relação aos resultados explorados na Seção 7.2.

7.3.1 Detecção de anomalias

Dos resultados apresentados nas Seções 7.2.1 e 0, destacamos duas janelas para as quais o desempenho da detecção de anomalias é muito limitado: $Original$ (termografia) e $\Delta T\_1750\_Original$ (assinatura elétrica). No caso termográfico (janela $Original$), embora o autoencoder esteja adequadamente treinado (Figura 91), o erro de detecção é significativamente alto (Figura 92). Para o caso elétrico (janela $\Delta T\_1750\_Original$), o treinamento do autoencoder não converge (Figura 46), o que inviabiliza seu uso como detector de anomalias (Figura 87). Ambos os resultados podem ser explicados pelos valores médios de NQ e divergência-KL para o teste intraclasse ($\langle Dis_{(OK-OK)} \rangle$) mostrados na Tabela 21, onde se destacam em azul as duas janelas aqui mencionadas.

Tabela 21 – Dissimilaridade estatística intraclasse para dados da classe OK. Resultados médios para 160000 execuções (440 x 400 imagens, ver Seção 7.1.1) de cada janela

Janela	$\langle \mathrm{NQ}_{\mathrm{OK-OK}} \rangle$	$\langle \mathrm{Divergência} - \mathrm{KL}_{\mathrm{OK-OK}} \rangle$
	Assinatura elétrica	
500_Original	0,02933	1125,13
750_Original	0,03323	1279,94
1000_Original	0,03534	1231,77
1250_Original	0,03738	1318,43
1500_Original	0,04046	1294,92
1750_Original	0,17231	8130,62
2000_Original	0,04240	1281,38
2250_Original	0,03968	1282,04
2500_Original	0,03940	1225,44
2750_Original	0,04350	1185,09
3000_Original	0,04032	1135,63
3500_Original	0,04047	1071,24
Média	**0,04948**	**1796,80**
	Assinatura termográfica	
Original	0,02174	1096,74
FM	0,02366	762,59
ATSR1000	0,07710	1845,42
Detrend	0,09729	2359,02
Detrend FM	0,05124	1066,14
Média	**0,0542**	**1425,98**

Fonte: O Autor (2022).

Para a janela $\Delta T\_1750\_Original$, a NQ e a divergência-KL intraclasses ($\langle MSE_{\ OK-OK} \rangle$ e $\langle Divergência - KL_{\ OK-OK} \rangle$) são 3,48 e 4,52 vezes maiores do que a média, respectivamente. Assim, embora uma razão maior entre os valores médios interclasse e intraclasse aumente o desempenho (Figura 90 e Figura 95), um alto valor absoluto da distância estatística entre amostras da mesma classe (de treinamento) limita claramente o treinamento do autoencoder. Uma discussão interessante envolvendo as características dos dados de treinamento e o treinamento de modelos de aprendizado de máquina é encontrada em (LeCun et al.,

2012). Nesse estudo, diversas heurísticas para facilitar o treinamento de redes neurais artificiais são propostas. Afirma-se, entre outras coisas, que os dados de treinamento devem “todos ter aproximadamente a mesma covariância”. Este não é o caso da janela $\Delta T\_1750\_Original$. As imagens dessa janela apresentam covariâncias não equalizadas, já que a dissimilaridade média intraclasse é elevada. Isso prejudica o treinamento do autoencoder.

Por outro lado, a janela termográfica $Original$ apresenta NQ e divergência-KL intraclasse próximos da média. Logo, embora o treinamento do autoencoder seja viável (pois, $\langle Dis_{(OK-OK)}\rangle$ é baixo), há pouca separação entre as classes (i.e., $\langle Dis_{(OK-NOK)}\rangle / \langle Dis_{(OK-OK)}\rangle \cong 1$), e isso limita o desempenho do modelo.

7.3.2 Classificação de substâncias

No caso do uso do autoencoder para classificação de substâncias, ressaltamos que a transformação poderia ter sido aplicada ao próprio sinal transmitido. Pois, esse sinal é “artificialmente” projetado com a finalidade de criar uma assinatura após a propagação através da substância. A transformação, portanto, já aumentaria o conteúdo espectral do sinal emitido. Isso não seria viável no caso da detecção de anomalias, pois o formato do sinal depende do próprio DUT e não é passível de modificação antes de ser medido.

De qualquer forma, neste caso, a transformação aplicada ao sinal detectado também se mostra eficiente para aumentar o desempenho do sistema de classificação, o que serve para reafirmar a validade da técnica proposta.

7.4 SÍNTESE DO CAPÍTULO

Neste capítulo, apresentamos uma forma de aumentar o desempenho dos sistemas de detecção de anomalias e de classificação de substâncias descritos nas Seções 4, 5 e 6. Os melhores desempenhos entre as transformadas investigadas foram obtidos com as transformadas combinadas Detrend & FM (para o caso termográfico) e a transformada FM (para o caso de corrente elétrica). A transformação Detrend é indicada nos casos em que há uma componente DC relativamente alta no sinal bruto. Para os casos em que o sinal possui frequências mais elevadas em torno de um nível DC próximo ao zero, a modulação em

frequência (FM) apresenta bons resultados. Em alguns casos, a combinação Detrend & FM leva a uma melhoria do desempenho ainda mais pronunciada.

As imagens usadas no treinamento e no teste do autoencoder são derivadas de espectrogramas. Assim, a modulação de frequência adiciona conteúdo harmônico à amostra. Especula-se que este conteúdo harmônico seja adicionado em diferentes quantidades, dependendo da classe a que a amostra pertence. Consequentemente, aumenta-se a distância entre as classes.

O aumento da distância estatística média entre as diferentes classes de dados após a transformação é uma explicação factível para a melhoria do desempenho do autoencoder. Pois, quanto maior a distância entre as classes, mais fácil para o autoencoder distinguir imagens de diferentes classes através do erro de reconstrução. E, como consequência, há uma diminuição da taxa de erro mínima do modelo.

Na assinatura termográfica, menos amostras estão disponíveis devido à restrição de tempo de aquisição contínua da câmera infravermelha. Além disso, o valor absoluto da temperatura varia em trono de um nível DC elevado. Para este caso, mostramos que a transformação ajuda a reduzir a taxa de erro e aumentar o parâmetro Q, mesmo com essas adversidades.

Para o caso de corrente elétrica, o grande número de amostras disponíveis possibilita o treinamento do autoencoder para a maioria das situações. No entanto, não foi possível treinar o modelo para um determinado comprimento de janela ($\Delta T = 1750\,ms$). Neste caso, o pré-processamento dos dados viabiliza o treinamento do autoencoder e, consequentemente, a obtenção de um desempenho aceitável na detecção de anomalias.

Para a classificação de substâncias, a aplicação das transformadas FM e Detrend & FM permitem reduzir a complexidade do sinal óptico transmitido, i. e., pode-se reduzir a quantidade de comprimentos de onda desse sinal e ainda assim, obter um nível de desempenho ótimo.

Em todos os casos analisados, com o intuito de realizar uma análise quantitativa mais aprofundada, usamos o parâmetro de desempenho Q, que leva em

consideração tanto a taxa de erro quanto a faixa do limite de decisão, para a qual a taxa de erro permanece aceitável.

Em um comentário final, ressaltamos que o índice de melhoria de desempenho – que pode ser medido por meio do parâmetro Q, por exemplo – apenas deve ser utilizado para comparar os resultados de um mesmo modelo, quando treinado e testado com dados gerados a partir dos mesmos sinais brutos. Por exemplo, comparar numericamente os resultados da Tabela 18 e da Tabela 16 não é razoável, visto que se trata de dois modelos treinados com imagens geradas a partir de sinais diferentes. A razão para essa restrição é o fato de que cada modelo apresenta seu próprio comportamento em relação à distribuição estatística dos erros de reconstrução para as classes OK e NOK (como na Figura 26).

8 CONCLUSÕES

Em nosso trabalho de pesquisa, abordamos soluções de aprendizado de máquina aplicados à Indústria 4.0. Especificamente, os esforços se concentraram em duas grandes áreas: detecção de anomalias em sistemas embarcados e classificação de substâncias químicas ou biológicas. Em ambos os casos, as soluções apresentadas se baseiam no modelo de aprendizado de máquina autoencoder. Para casos práticos, esses modelos apresentam uma vantagem: treinar um autoencoder requer apenas a classe de referência. Isso simplifica o processo de implementação da solução em um ambiente industrial, onde a obtenção de dados da classe anômala (ou diferente da referência) é bastante complexa e limitada.

O autoencoder, ao ser treinado com dados de uma determinada classe, reconstrói cópias de um novo dado apresentado à sua entrada de forma muito similar, caso o dado de entrada pertença à mesma classe de treinamento. Do contrário, a reconstrução ocorre de maneira menos fidedigna. A partir dessa premissa, coletamos dados do sistema ou substância a serem testados e criamos imagens que contém assinaturas de seu comportamento (espectrogramas). Usando espectrogramas da classe de referência, treinamos o autoencoder. Dessa forma, ao serem apresentadas imagens de outra classe que não seja a de treinamento à entrada do autoencoder, é possível detectar a ocorrência de uma anomalia ou indicar se a substância testada pertence à classe de treinamento.

Em relação à detecção de anomalias, duas soluções foram propostas: utilizando assinatura de corrente elétrica consumida ou assinatura termográfica do DUT. No primeiro caso, a corrente consumida ao longo de um período é medida, armazenada e convertida em imagem espectrográfica. No segundo caso, a variação temporal de temperatura é adquirida e convertida em imagem espectrográfica. Essas imagens são, por fim, usadas como entrada para o autoencoder, tanto para treinamento (conjunto específico para treinamento) como para teste de desempenho (conjunto específico para teste).

A classificação de substâncias também é realizada por meio da comparação de imagens espectrográficas. No entanto, emite-se um sinal óptico que se propaga através da substância em teste. Esse sinal tem parte de sua energia absorvida pela

substância. Tal absorção depende do comprimento de onda óptico do sinal e da transmitância da substância que ele atravessa. O sinal óptico capturado por um fotodetector, portanto, contém uma assinatura da própria substância. Pode-se, assim, utilizar as imagens espectrográficas obtidas a partir do sinal detectado para comparar a substância de teste com uma substância de referência utilizada para treinar o autoencoder.

Para validar as propostas, projetamos e construímos arranjos experimentais para cada caso estudado. Nos casos de detecção de anomalias, um protótipo de sistema embarcado foi montado e utilizado para geração de dados experimentais. A classificação de substâncias é apresentada em forma de prova de conceito, já que não haveria tempo suficiente para montar um experimento com amostras de substâncias químicas ou biológicas reais. Assim, simulamos em computador o comportamento espectral de substâncias de teste para validar a solução proposta.

De uma maneira geral, os resultados experimentais comprovam a eficácia das soluções propostas. Para todos os casos estudados, pelo menos uma configuração apresenta desempenho ótimo, ou seja, com taxa de erro nula. Além da taxa de erro, usamos diversos outros parâmetros para analisar o desempenho de cada modelo, por exemplo, curvas ROC e AUC. Entretanto, nem sempre esses índices são capazes de discernir dois modelos com taxa de erro nula. Entendemos que, a depender do comportamento estatístico dos erros de reconstrução das imagens do autoencoder, um modelo pode ser mais vantajoso que outro. Então, propomos um parâmetro de medição de desempenho que leva em consideração não só a taxa de erro mínima, mas também seu comportamento em função do limiar de decisão.

Algumas configurações dos modelos propostos apresentaram um desempenho aquém do aceitável. Por exemplo, para o modelo baseado em assinatura termográfica, apesar de o treinamento do autoencoder convergir, a taxa de erro foi próxima da máxima, ou seja, o pior resultado possível. Isso nos levou a buscar uma forma de contornar essa limitação. Propomos, então uma técnica de pré-processamento dos dados por meio de transformações matemáticas antes de gerar as imagens espectrográficas. Em seguida, testamos as transformações em vários casos, tanto de detecção de anomalias quanto de classificação de substâncias. Uma melhoria no desempenho, geralmente significativa, foi alcançada em todos os casos, quando comparado com o sinal bruto (não transformado). Por exemplo, para a

detecção de anomalia por assinatura elétrica, a taxa de erro de detecção obtida caiu de 50% para menos de 1%. No caso termográfico, a taxa de erro de detecção caiu de 35% aproximadamente para menos de 0,5%. Na classificação de substâncias, o sistema testado mais simples – com apenas 2 comprimentos de onda transmitidos – teve sua taxa de erro de classificação reduzida de quase 40% para 0%, ou seja, uma classificação perfeita.

Ressaltamos que o emprego das transformações propostas não se limita aos casos relatados neste trabalho. Entendemos que transformar sinais temporais gera espectrogramas com características espectrais que geralmente levam a um melhor desempenho de detecção de anomalias. Em princípio, qualquer modelo de aprendizado de máquina baseado em espectrograma pode se beneficiar da transformação de dados. É uma solução simples, eficiente e flexível o suficiente para ser aplicada a uma variedade de modelos de aprendizado de máquina.

Consideração prática: para todos os sistemas (de detecção de anomalia ou classificação de substâncias), dependendo da duração da janela de aquisição utilizada, é possível aumentar a confiabilidade dos resultados por meio da repetição do teste. Por exemplo, em um sistema de detecção de anomalias em que a duração da janela de aquisição projetada seja de cinco segundos, se repetirmos a aquisição dez vezes, o tempo total da aquisição será de cinquenta segundos. Assim, teremos amostras para realizar dez testes parciais. O veredito final sobre a presença ou não de anomalia pode ser definida pela maioria dos dez resultados parciais. Neste caso, o custo adicional seria o maior tempo de testagem.

Do ponto de vista da aplicabilidade prática das soluções apresentadas, testes não invasivos de sistemas embarcados são uma demanda recorrente na indústria. Desenvolver um sistema de detecção de anomalias invasivo para um novo modelo de produto representa uma fração importante dos custos de produção. Um sistema não invasivo e flexível, por outro lado, é bem mais eficiente, pois demanda menos tempo e pode ser facilmente adaptado para novos produtos. Já a técnica proposta de análise de substâncias pode ser explorada de diversas formas. Por exemplo, na indústria alimentícia ou química; no ramo de análise laboratorial, como detecção de patógenos; ou ainda em aplicações de proteção ambiental, como avaliação da qualidade de água.

8.1 TRABALHOS FUTUROS

Apesar dos resultados obtidos, sem dúvidas ainda existem muitos pontos a serem explorados relativos ao que foi desenvolvido. A seguir listamos algumas questões, que demandariam um esforço adicional que não caberia nesta tese, a serem consideradas em trabalhos futuros:

- Entender os processos internos do modelo que levam ao aumento do desempenho após o uso de transformação dos dados;
- Entender porque a janela 1750 não consegue ser treinada;
- Estudar os detalhes que levam certas transformações a gerar imagens espectrográficas com separação estatística maior do que outras;
- Aprofundar a análise da definição da ROI no caso termográfico;
- Tratar casos mais gerais considerando existência de possíveis *outliers* nos sinais capturados do sistema embarcado devido a interferência do usuário em sua operação;
- Validar a técnica de classificação de substâncias com amostras químicas e/biológicas reais;
- Expandir a utilização da assinatura elétrica ou termográfica para ataque criptográfico de *hardware*. Experimentos podem ser conduzidos usando FPGA (circuitos digitais reconfiguráveis), por exemplo;
- Expandir a funcionalidade dos modelos de detecção de anomalias para, além de detectar, classificar os defeitos;
- Realizar implantação piloto em uma linha de produção real para validar a técnica em ambiente real. Este caso demanda a parceria com o setor produtivo.

8.2 ARTIGOS RELACIONADOS À TESE

Nesta seção, listamos artigos submetidos ao longo do desenvolvimento deste trabalho e relacionados ao assunto abordado.

Trabalhos aceitos em periódicos

DE OLIVEIRA, José Paulo G.; BASTOS-FILHO, Carmelo JA; OLIVEIRA, Sergio Campello. Non-invasive embedded system hardware/firmware anomaly detection based on the electric current signature. Advanced Engineering Informatics, 2022, 51. Jg., S. 101519.

Trabalhos submetidos em periódicos

DE OLIVEIRA, José Paulo G.; BASTOS-FILHO, Carmelo JA; OLIVEIRA, Sergio Campello. Improving Autoencoder-Based Anomaly Detection in Embedded Systems using Data Transformation. Soft Computing, 2022.

Trabalhos aceitos em conferências

DE OLIVEIRA, José Paulo G.; BASTOS-FILHO, Carmelo JA; OLIVEIRA, Sergio Campello. Non-intrusive Embedded Systems Anomaly Detection using Thermography and Machine Learning. Congresso Brasileiro de Inteligência Computacional, 2021, Joinville. Anais do 15. Congresso Brasileiro de Inteligência Computacional, 2021. p. 1.

DE OLIVEIRA, José Paulo G.; BASTOS-FILHO, Carmelo JA; OLIVEIRA, Sergio Campello. Chemical sample classification using autoencoder-based spectroscopy. SBFOTON IOPC 2022 - International Optics and Photonics Conference.

Orientação de Trabalho de Conclusão de Curso

CABRAL, T. F.; José Paulo G. Sistema De Detecção De Anomalias Em Sistemas Embarcados. 2021. Trabalho de Conclusão de Curso (Graduação em Engenharia da Computação) - Universidade de Pernambuco.

Aprovação em editais de inovação

Edital catalisa ICT para seleção de pesquisas com potencial de inovação. SEBRAE, 2021.

8.3 ARTIGOS NÃO RELACIONADOS À TESE

Nesta seção, listamos artigos submetidos ao longo do desenvolvimento deste trabalho, mas que não são relacionados ao assunto abordado.

Trabalhos aceitos em periódicos

DE SOUZA JERONIMO, Bruno, et al. Comparing Social Robot Embodiment for Child Musical Education. Journal of Intelligent & Robotic Systems, 2022, 105. Jg., Nr. 2, S. 1-16.

MONTEIRO, Rodrigo P., et al. Improving Adaptive Filters for Active Noise Control Using Particle Swarm Optimization. International Journal of Swarm Intelligence Research (IJSIR), 2018, 9. Jg., Nr. 4, S. 47-64.

Trabalhos aceitos em conferências

MELO, Rodrigo, et al. Guitar tuner and song performance evaluation using a nao robot. In: 2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE). IEEE, 2020. S. 1-6.

MONTEIRO, Rodrigo P., et al. Accelerating the convergence of adaptive filters for active noise control using particle swarm optimization. In: 2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI). IEEE, 2017. S. 1-6.

REFERÊNCIAS

ABEDIN, Mohammad Zoynul, et al. Tax default prediction using feature transformation-based machine learning. **IEEE Access**, 2020, 9. Jg., S. 19864-19881.

ACCIANI, G.; BRUNETTI, G.; FORNARELLI, G. A multiple neural network system to classify solder joints on integrated circuits. **International Journal of computational intelligence Research**, 2006, 2. Jg., Nr. 4, S. 337-348.

ALAOUI, Nabil El Belghiti, et al. New defect detection approach using near electromagnetic field probing of high density PCBAs. **Microelectronics Reliability**, 2018, 88. Jg., S. 288-293.

ALDRICH, Eric. **A Package of Functions for Computing Wavelet Filters, Wavelet Transforms and Multiresolution Analyses**. 2013.

ALFEO, Antonio L., et al. Using an autoencoder in the design of an anomaly detector for smart manufacturing**. Pattern Recognition Letters**, 2020, 136. Jg., S. 272-278.

BACH, Francis. Breaking the curse of dimensionality with convex neural networks. **The Journal of Machine Learning Research**, 2017, 18. Jg., Nr. 1, S. 629-681.

BAUDAT, Gaston; ANOUAR, Fatiha. Feature vector selection and projection using kernels. Neurocomputing, 2003, 55. Jg., Nr. 1-2, S. 21-38.

BAYGIN, Mehmet, et al. Machine vision based defect detection approach using image processing. In: **2017 international artificial intelligence and data processing symposium (IDAP)**. Ieee, 2017. S. 1-5.

BERMAN, Paul R.; MALINOVSKY, Vladimir S. **Principles of laser spectroscopy and quantum optics**. Princeton University Press, 2011.

BISHARA, Anthony J.; HITTNER, James B. Reducing bias and error in the correlation coefficient due to nonnormality. **Educational and psychological measurement**, 2015, 75. Jg., Nr. 5, S. 785-804.

BOYER, Stuart A. **Supervisory control and data acquisition**. Isa, 1999.

BRADLEY, Andrew P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. **Pattern recognition**, 1997, 30. Jg., Nr. 7, S. 1145-1159.

BREIMAN, Leo, et al. randomForest: **Breiman and Cutler's random forests for classification and regression**. R package version, 2018, 4. Jg., S. 6-14.

CAI, Yu; HUANG, Yanjin; ZHANG, Shugong. Research of defect inspection and processing in PCB automatic optical inspection. In: **Proceedings of the 2012 International Conference on Electronics, Communications and Control**. 2012. S. 803-806.

CALDERS, Toon; ŽLIOBAITĖ, Indrė. Why unbiased computational processes can lead to discriminative decision procedures. In: **Discrimination and privacy in the information society**. Springer, Berlin, Heidelberg, 2013. S. 43-57.

CELEBI, M. Emre; AYDIN, Kemal (Hg.). **Unsupervised learning algorithms**. Berlin: Springer International Publishing, 2016.

CHALAPATHY, Raghavendra; CHAWLA, Sanjay. Deep learning for anomaly detection: A survey. **arXiv preprint arXiv:1901.03407**, 2019.

CHO, Il-Hoon, et al. Nano/micro and spectroscopic approaches to food pathogen detection. **Annu. Rev. Anal. Chem**, 2014, 7. Jg., Nr. 1, S. 65-88.

CHOLLET, Francois. Building autoencoders in keras. **The Keras Blog**, 2016, 14. Jg.

CHOLLET, Francois. **Deep learning with Python**. Simon and Schuster, 2021.

CHUANG, Shui-Fa, et al. Misalignment inspection of multilayer PCBs with an automated X-ray machine vision system. **The International Journal of Advanced Manufacturing Technology**, 2010, 51. Jg., Nr. 9, S. 995-1008.

CIOS, Krzysztof J., et al. Unsupervised learning: clustering. In: **Data Mining**. Springer, Boston, MA, 2007. S. 257-288.

COHEN, Leon. Time-frequency distributions-a review. **Proceedings of the IEEE**, 1989, 77. Jg., Nr. 7, S. 941-981.

COWPERTWAIT, Paul SP; METCALFE, Andrew V. **Introductory time series with R**. Springer Science & Business Media, 2009.

CUNNINGHAM, Pádraig; CORD, Matthieu; DELANY, Sarah Jane. Supervised learning. In: **Machine learning techniques for multimedia**. Springer, Berlin, Heidelberg, 2008. S. 21-49.

DAI, Wenting, et al. Soldering defect detection in automatic optical inspection. **Advanced Engineering Informatics**, 2020, 43. Jg., S. 101004.

DEBNATH, Lokenath; SHAH, Firdous Ahmad. **Wavelet transforms and their applications**. Boston: Birkhäuser, 2002.

DEVEZAS, Tessaleno; SARYGULOV, Askar. **Industry 4.0**. Basel: Springer, 2017.

DONG, Ronglu, et al. Detection and direct readout of drugs in human urine using dynamic surface-enhanced Raman spectroscopy and support vector machines. **Analytical chemistry**, 2015, 87. Jg., Nr. 5, S. 2937-2944.

DU PIN CALMON, Flavio, et al. Data pre-processing for discrimination prevention: Information-theoretic optimization and analysis. **IEEE Journal of Selected Topics in Signal Processing**, 2018, 12. Jg., Nr. 5, S. 1106-1119.

DWORK, Cynthia, et al. The reusable holdout: Preserving validity in adaptive data analysis. **Science**, 2015, 349. Jg., Nr. 6248, S. 636-638.

EL BELGHITI ALAOUI, Nabil, et al. Upgrading In-Circuit Test of High Density PCBAs Using Electromagnetic Measurement and Principal Component Analysis. **Journal of Electronic Testing**, 2018, 34. Jg., Nr. 6, S. 749-762.

ELLIOTT, Douglas F. **Handbook of digital signal processing: engineering applications**. Elsevier, 2013.

ESSER, Michael; STRUSS, Peter. Fault-Model-Based Test Generation for Embedded Software. In: **IJCAI**. 2007. S. 342-347.

FAN, Jerome; UPADHYE, Suneel; WORSTER, Andrew. Understanding receiver operating characteristic (ROC) curves. **Canadian Journal of Emergency Medicine**, 2006, 8. Jg., Nr. 1, S. 19-20.

FONG, Ruth C.; SCHEIRER, Walter J.; COX, David D. Using human brain activity to guide machine learning. **Scientific reports**, 2018, 8. Jg., Nr. 1, S. 1-10.

FRANCOIS, Ch. **Deep Learning with Python**. Manning Publications. 2017.

FRATELLO, Michele; TAGLIAFERRI, Roberto. Decision trees and random forests. **Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics**, 2018, 1. Jg., S. 3.

GALATRO, Daniela, et al. Battery Health Diagnosis Approach Integrating Physics-Based Modeling with Electrochemical Impedance Spectroscopy. **Energy Technology**, 2022, 10. Jg., Nr. 4, S. 2100942.

GEMAN, Stuart; BIENENSTOCK, Elie; DOURSAT, René. Neural networks and the bias/variance dilemma. **Neural computation**, 1992, 4. Jg., Nr. 1, S. 1-58.

GENUER, Robin; POGGI, Jean-Michel; TULEAU-MALOT, Christine. Variable selection using random forests. **Pattern recognition letters**, 2010, 31. Jg., Nr. 14, S. 2225-2236.

GIANNAKOPOULOS, Theodoros; PIKRAKIS, Aggelos. **Introduction to audio analysis: a MATLAB® approach**. Academic Press, 2014.

GÓMEZ, J., et al. A robotic system for PCBs inspection based on computer vision and mobile probes. **IFAC Proceedings Volumes**, 2007, 40. Jg., Nr. 3, S. 171-176.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep learning**. MIT press, 2016.

GROUT, Ian A. Test Pattern Generation and Fault Simulation. **Integrated Circuit Test Engineering: Modern Techniques**, 2006, S. 235-255.

GUO, Fenglin; GUAN, Shu-an. Research of the machine vision based PCB defect inspection system. In: **2011 International Conference on Intelligence Science and Information Engineering**. IEEE, 2011. S. 472-475.

HAN, Jiawei; PEI, Jian; TONG, Hanghang. **Data mining: concepts and techniques**. Morgan kaufmann, 2022.

HASTIE, Trevor, et al. **The elements of statistical learning: data mining, inference, and prediction**. New York: springer, 2009.

HAYKIN, Simon; NETWORK, N. **A comprehensive foundation. Neural networks**, 2004, 2. Jg., Nr. 2004, S. 41.

HEIBATI, Masoumeh, et al. Assessment of drinking water quality at the tap using fluorescence spectroscopy**. Water research**, 2017, 125. Jg., S. 1-10.

HOTHORN, Torsten; HORNIK, Kurt; ZEILEIS, Achim. ctree: Conditional inference trees. **The comprehensive R archive network**, 2015, 8. Jg.

HUNT, Earl B. Artificial intelligence. Academic Press, 2014.

HUNTER, J. Stuart. The exponentially weighted moving average. **Journal of quality technology**, 1986, 18. Jg., Nr. 4, S. 203-210.

JAMALUDIN, Juliza; ROHANI, Jemmy Mohd. Cyber-physical system (cps): State of the art. In: **2018 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)**. IEEE, 2018. S. 1-5.

KAMEI, Yasutaka; SHIHAB, Emad. Defect prediction: Accomplishments and future challenges. In: **2016 IEEE 23rd international conference on software analysis, evolution, and reengineering (SANER)**. IEEE, 2016. S. 33-45.

KETKAR, Nikhil. Introduction to keras. In: **Deep learning with Python**. Apress, Berkeley, CA, 2017. S. 97-111.

KHANDPUR, Raghbir Singh. **Printed circuit boards: design, fabrication, assembly and testing**. Tata McGraw-Hill Education, 2006.

KINGMA, Diederik P.; BA, Jimmy. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

KRA, Yehuda. A cross-debugging method for hardware/software co-design environments. In: 30th **ACM/IEEE Design Automation Conference**. IEEE, 1993. S. 673-677.

LAMPEN, Peter, et al. JCAMP-DX for mass spectrometry. **Applied spectroscopy**, 1994, 48. Jg., Nr. 12, S. 1545-1552.

LECUN, Yann A., et al. **Efficient backprop. In: Neural networks: Tricks of the trade**. Springer, Berlin, Heidelberg, 2012. S. 9-48.

LEE, Edward Ashford; SESHIA, Sanjit Arunkumar. **Introduction to embedded systems: A cyber-physical systems approach**. Mit Press, 2016.

LIU, Fei Tony; TING, Kai Ming; ZHOU, Zhi-Hua. Isolation forest. In: **2008 eighth ieee international conference on data mining**. IEEE, 2008. S. 413-422.

LOWE, David G. Distinctive image features from scale-invariant keypoints. **International journal of computer vision**, 2004, 60. Jg., Nr. 2, S. 91-110.

LU, Shui-Shong; CHU, Ju-Lih; JANG, Huey-Chin. Development of a novel coordinate transposing fixture system. **The International Journal of Advanced Manufacturing Technology**, 1997, 13. Jg., Nr. 5, S. 350-358.

LU, Yang. Cyber physical system (CPS)-based industry 4.0: A survey. **Journal of Industrial Integration and Management**, 2017, 2. Jg., Nr. 03, S. 1750014.

MALACARA, Daniel; SERVIN, Manuel; MALACARA, Zacarias. **Interferogram analysis for optical testing**. CRC press, 2018.

MAMCHUR, Yaryna, et al. Thermography investigation of soldered joints for LED mounting. In: **2020 IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO)**. IEEE, 2020. S. 143-147.

MANJULA, C.; FLORENCE, Lilly. Deep neural network based hybrid approach for software defect prediction using software metrics. **Cluster Computing**, 2019, 22. Jg., Nr. 4, S. 9847-9863.

MARTINEZ-CUAZITL, Adriana, et al. ATR-FTIR spectrum analysis of saliva samples from COVID-19 positive patients. **Scientific Reports**, 2021, 11. Jg., Nr. 1, S. 1-14.

MARWEDEL, Peter. **Embedded system design: embedded systems foundations of cyber-physical systems, and the internet of things**. Springer Nature, 2021.

MAZIDI, Muhammad Ali, et al. **TI Tiva ARM Programming For Embedded Systems: Programming ARM Cortex-M4 TM4C123G with C** (Volume 2). 2017.

MEHTA, Pankaj, et al. A high-bias, low-variance introduction to machine learning for physicists. **Physics reports**, 2019, 810. Jg., S. 1-124.

MILALI, Masabho P., et al. An autoencoder and artificial neural network-based method to estimate parity status of wild mosquitoes from near-infrared spectra. **PLoS One**, 2020, 15. Jg., Nr. 6, S. e0234557.

MITCHELL, Tom M.; MITCHELL, Tom M. **Machine learning**. New York: McGraw-hill, 1997.

MONTEIRO, R. P.; BASTOS-FILHO, C. J. A. Using Kullback-Leibler Divergence to Identify Prominent Sensor Data for Fault Diagnosis. In: **International Conference on Intelligent Data Engineering and Automated Learning**. Springer, Cham, 2020a. p. 136-147.

MONTEIRO, R. P., BASTOS-FILHO, C. J., CERRADA, M., CABRERA, D. R., & SÃNCHEZ, R. V. Using The Kullback-Leibler Divergence And Kolmogorov-Smirnov Test To Select Input Sizes To The Fault Diagnosis Problem Based On A Cnn Model 2020b. **Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC)**, Vol. 18, Iss. 2, pp. 16-26.

MOSER, Raimund; PEDRYCZ, Witold; SUCCI, Giancarlo. A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction. In: **Proceedings of the 30th international conference on Software engineering**. 2008. S. 181-190.

MUJEEB, Abdul, et al. One class based feature learning approach for defect detection using deep autoencoders. **Advanced Engineering Informatics**, 2019, 42. Jg., S. 100933.

MUKHERJEE, Rajdeep, et al. Formal techniques for effective co-verification of hardware/software co-designs. In: **2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)**. IEEE, 2017. S. 1-6.

NANDA, Umakanta; PATTNAIK, Sushant Kumar. Universal asynchronous receiver and transmitter (uart). In: **2016 3rd international conference on advanced computing and communication systems (ICACCS)**. IEEE, 2016. S. 1-5.

NELSON, Mark J.; HOOVER, Amy K. Notes on using Google Colaboratory in AI education. In: **Proceedings of the 2020 ACM conference on innovation and Technology in Computer Science Education**. 2020. S. 533-534.

NIST 2-HEPTANONE, National Istitute of Standards and Technology. Livro de Química na Web, SRD 69. Available from: https://webbook.nist.gov/cgi/cbook.cgi?ID=C110430&Units=SI&Type=IR-SPEC&Index=1#IR-SPEC

NIST 3-HEPTANONE, National Istitute of Standards and Technology. Livro de Química na Web, SRD 69. Available from: https://webbook.nist.gov/cgi/cbook.cgi?ID=C106354&Units=SI&Mask=80#IR-Spec

NOGUEIRA, João Marcelo, et al. Backpropagation neural network for analysis and classification of fluorescence spectroscopy of squamous cell carcinoma in animal model. In: **2021 SBFoton International Optics and Photonics Conference (SBFoton IOPC)**. IEEE, 2021. S. 1-4.

OJEDA-AGUIRRE, Noe Alejandro, et al. Reassigned Short Time Fourier Transform and K-means Method for Diagnosis of Broken Rotor Bar Detection in VSD-fed Induction Motors. **Advances in Electrical and Computer Engineering**, 2019, 19. Jg., Nr. 2, S. 61-68.

PANG, Bo; NIJKAMP, Erik; WU, Ying Nian. Deep learning with tensorflow: A review. **Journal of Educational and Behavioral Statistics**, 2020, 45. Jg., Nr. 2, S. 227-248.

PARVAT, Aniruddha, et al. A survey of deep-learning frameworks. In: **2017 International Conference on Inventive Systems and Control (ICISC)**. IEEE, 2017. S. 1-7.

PAVIA, Donald L., et al. **Introduction to spectroscopy**. Cengage learning, 2014.

PIETRUSZKA, Wolf Dieter. **MATLAB® und Simulink® in der Ingenieurpraxis**. Wiesbaden: Vieweg+ Teubner Verlag, 2012.

PLEVRIDIS, S. E.; BIRBAS, A. N.; DERVENIS, J. Hand held spectrometer with networking capability for use in chemical industry. In: **1995 Proceedings of the IEEE International Symposium on Industrial Electronics**. IEEE, 1995. S. 777-780.

RAJALAKSHMI, G., et al. Identification of moisture, glucose, sucrose, fructose region in honey sample using NIR spectroscopy. In: **2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS).** IEEE, 2017. S. 389-391.

RENSTRÖM, Niklas; BANGALORE, Pramod; HIGHCOCK, Edmund. System-wide anomaly detection in wind turbines using deep autoencoders. **Renewable Energy**, 2020, 157. Jg., S. 647-659.

RICHTER, Johannes; STREITFERDT, Detlef; ROZOVA, Elena. On the development of intelligent optical inspections. In: **2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)**. IEEE, 2017. S. 1-6.

RIMAL, Yagyanath. Machine Learning Random Forest Cluster Analysis for Large Overfitting Data: using R Programming. In: **2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)**. IEEE, 2019. S. 1265-1271.

ROKACH, Lior; MAIMON, Oded. Decision trees. In: **Data mining and knowledge discovery handbook**. Springer, Boston, MA, 2005. S. 165-192.

SAKURADA, Mayu; YAIRI, Takehisa. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: **Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis**. 2014. S. 4-11.

SANA, Joydeb Kumar, et al. Data transformation based optimized customer churn prediction model for the telecommunication industry. **arXiv preprint arXiv:2201.04088**, 2022.

SARAVANAN, N.; RAMACHANDRAN, K. I. Incipient gear box fault diagnosis using discrete wavelet transform (DWT) for feature extraction and classification using artificial neural network (ANN). **Expert systems with applications**, 2010, 37. Jg., Nr. 6, S. 4168-4181.

SHORTEN, Connor; KHOSHGOFTAAR, Taghi M. A survey on image data augmentation for deep learning. **Journal of big data**, 2019, 6. Jg., Nr. 1, S. 1-48.

SIDHIK, Siraj; ITTIARAH, Jijo V.; GANGOPADHYAY, Tarun Kumar. A fiber loop based chemical sensor for industrial application. In: **2015 1st Conference on Power, Dielectric and Energy Management at NERIST (ICPDEN)**. IEEE, 2015. S. 1-3.

SIEGEL, Hardo; EGGERSDORFER, Manfred. Ketones. **Ullmann's Encyclopedia of Industrial Chemistry,** 2000.

SILVA, Leandro H. de S., et al. Automatic optical inspection for defective PCB detection using transfer learning. In: **2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)**. IEEE, 2019. S. 1-6.

SINGH, Praman Deep; CHUG, Anuradha. Software defect prediction analysis using machine learning algorithms. In: **2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence**. IEEE, 2017. S. 775-781.

SPEAKMAN, John R.; WARD, S. **Infrared thermography: principles and applications**. Zoology-Jena-, 1998, 101. Jg., S. 224-232.

SUTHAR, Madhuri; ASGHARI, Hossein; JALALI, Bahram. Feature enhancement in visually impaired images. **IEEE Access**, 2017, 6. Jg., S. 1407-1415.

TAHA, E. Mahmoud; EMARY, Eid; MOUSTAFA, Khalid. Automatic optical inspection for PCB manufacturing: A survey. **International Journal of Scientific and Engineering Research**, 2014, 5. Jg., Nr. 7, S. 1095-1102.

TATIBANA, Mauro Hiromu; LOTUFO, R. de A. Novel automatic PCB inspection technique based on connectivity. In: **Proceedings X Brazilian Symposium on Computer Graphics and Image Processing**. IEEE, 1997. S. 187-194.

TIAN, Zhen, et al. Applications of Raman Spectroscopy as an Non-Invasive Method in Pharmaceutical Analysis. In: **2010 Symposium on Photonics and Optoelectronics. IEEE**, 2010. S. 1-5.

TOOLS, F. L. I. R.; GUIDE, Users. Flir Systems. Inc.: Wilsonville, OR, USA, 2016. http://support.flir.com/answers/A1568/FLIR%20Tools%20User%20Guide%20v2.1.1.pdf (last access 05 August 2021).

TRES, A., et al. Authentication of organic feed by near-infrared spectroscopy combined with chemometrics: A feasibility study. **Journal of agricultural and food chemistry**, 2012, 60. Jg., Nr. 33, S. 8129-8133.

TSAKANIKAS, Panagiotis, et al. A machine learning workflow for raw food spectroscopic classification in a future industry. **Scientific Reports**, 2020, 10. Jg., Nr. 1, S. 1-11.

VAN DER SCHAFT, Peter H., et al. Fed-batch production of 2-heptanone by Fusarium poae. **Applied microbiology and biotechnology**, 1992, 36. Jg., Nr. 6, S. 709-711.

WAGH, Chaitali R.; BARU, Vijay B. Detection of faulty region on printed circuit board with IR thermography. **International Journal of Scientific & Engineering Research**, 2013, 4. Jg., Nr. 11, S. 1-4.

WATERS, R. Pink Floyd. (1973). The dark side of the moon [Album]. Harvest Records.

XIE, H., et al. Miniature fourier transform spectrometers based on electrothermal MEMS mirrors with large piston scan range. In: **2015 IEEE SENSORS. IEEE**, 2015. S. 1-4.

XU, Li Da; XU, Eric L.; LI, Ling. Industry 4.0: state of the art and future trends. **International journal of production research**, 2018, 56. Jg., Nr. 8, S. 2941-2962.

XU, Wen, et al. Improving performance of autoencoder-based network anomaly detection on nsl-kdd dataset. **IEEE Access**, 2021, 9. Jg., S. 140136-140146.

YANG, Peng; YANG, Guowei. Feature extraction using dual-tree complex wavelet transform and gray level co-occurrence matrix. **Neurocomputing**, 2016, 197. Jg., S. 212-220.

YOSINSKI, Jason, et al. How transferable are features in deep neural networks?. **Advances in neural information processing systems**, 2014, 27. Jg.

ZEBARI, Rizgar, et al. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. **Journal of Applied Science and Technology Trends**, 2020, 1. Jg., Nr. 2, S. 56-70.

ZHANG, Feng; KEIVANLOO, Iman; ZOU, Ying. Data transformation in cross-project defect prediction. **Empirical Software Engineering**, 2017, 22. Jg., Nr. 6, S. 3186-3218.

ZHOU, Chong; PAFFENROTH, Randy C. Anomaly detection with robust deep autoencoders. In: **Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining**. 2017. S. 665-674.

ZHU, Xiaojin Jerry. Semi-supervised learning literature survey. 2005.

ZONG, Pengyang; WANG, Yichen; XIE, Feng. Embedded software fault prediction based on back propagation neural network. In: **2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)**. IEEE, 2018. S. 553-558.