



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA
PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA

EMANUEL PINHEIRO SANTOS

**APRENDIZADO DE MÁQUINA PARA AUMENTAR PRECISÃO EM
TERMÔMETROS FOTOLUMINESCENTES**

Recife

2023

EMANUEL PINHEIRO SANTOS

**APRENDIZADO DE MÁQUINA PARA AUMENTAR PRECISÃO EM
TERMÔMETROS FOTOLUMINESCENTES**

Dissertação apresentada ao Programa de Pós-Graduação em Física da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de mestre em Física.

Área de concentração: Óptica.

Orientador (a): Anderson Stevens Leonidas Gomes

Coorientador (a): André de Lima Moura

Recife

2023

Catálogo na fonte
Bibliotecária Nataly Soares Leite Moro, CRB4-1722

S237a Santos, Emanuel Pinheiro
Aprendizado de máquina para aumentar precisão em termômetros
fotoluminescentes / Emanuel Pinheiro Santos – 2023.
89 f.: il., fig.

Orientador: Anderson Stevens Leonidas Gomes.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CCEN,
Física, Recife, 2023.
Inclui referências e apêndices.

1. Óptica. 2. Termometria. 3. Aprendizado de máquina. 4. Laser. I. Gomes,
Anderson Stevens Leonidas (orientador). II. Título.

535.2 CDD (23. ed.) UFPE- CCEN 2023 - 116

EMANUEL PINHEIRO SANTOS

**APRENDIZADO DE MÁQUINA PARA AUMENTAR PRECISÃO EM
TERMÔMETROS FOTOLUMINESCENTES**

Dissertação apresentada ao Programa de Pós-Graduação em Física da Universidade Federal de Pernambuco, no Departamento de Física, como requisito para a obtenção do título de Mestre em Física.

Aprovado em: 28/07/2023.

BANCA EXAMINADORA

Prof. Dr. Anderson Stevens Leonidas Gomes (Orientador)
Universidade Federal de Pernambuco - UFPE

Prof. Dr. Anderson Monteiro Amaral
Examinador Interno
Universidade Federal de Pernambuco - UFPE

Prof. Dr. Erving Clayton Ximendes
Examinador Externo
Universidad Autónoma de Madrid – UAM, Espanha

Em memória de Noel Barbosa Santos (06/11/1949 – 10/02/2023), meu pai.

AGRADECIMENTOS

Ao Criador do(s) universo(s), por nos dar a principal motivação de nossos estudos, a natureza.

À minha esposa Thayna, por estar sempre ao meu lado mesmo nos momentos mais difíceis.

Aos meus pais (Elza e Noel) e meus irmãos (Nael, Natanalle e Natanael) por fazerem parte em tudo na minha vida.

À minha segunda família (Alexandre, Lucidalva, Lucas e Letícia) por todo apoio.

Aos amigos construídos desde minha graduação até o mestrado: Evilásio, Jório, Vitor, Ithowyth, Hérika, Daiane, Ana, Regiane, Samuel, Célio, Rodrigo, Daniel, Juciane, Luan, Vinicius, Fernando, Cláudio, Yuri, Fahel, Stenio Jessica, Arthur, Rosana, Cecília, Denise, Igor, Jefferson, Anderson, Jailson e todos os outros que não tenho muita proximidade, mas fizeram parte direta ou indiretamente. Também quero agradecer a Dra. Roberta Pugina e ao Dr. Alyson Carvalho, pelo acompanhamento e grande contribuição na parte experimental dos resultados desta dissertação.

Aos professores Samuel Albuquerque e Askery Canabarro, ambos da UFAL, por contribuições diretas ao longo da minha carreira.

Ao professor e amigo recifense (e junqueirense), Paulo de Jesus, e sua família, por todo apoio durante esta jornada de meu mestrado.

Ao professor Anderson Gomes, por me aceitar prazerosamente em sua equipe de pesquisadores na UFPE e por ser um grande cientista que traz inspiração e motivação para nós.

A todos do Labfoton e do INFO, pelo amadurecimento científico que foi me passado nesta etapa.

Ao professor André de Lima Moura, por me acompanhar academicamente desde a Iniciação Científica e que, além de pesquisador e professor, em nível mais pessoal, tem se tornando um amigo e exemplo de profissional.

À banca examinadora, por todas as sugestões e correções que foram assertivas e necessárias para a versão final desse trabalho.

À CAPES, pelo investimento financeiro.

À Universidade Federal de Pernambuco.

RESUMO

Termometria Óptica (TO) é, essencialmente, um campo da física que estuda a temperatura e seus parâmetros usando ferramentas do domínio da óptica, como lasers, por exemplo. De forma objetiva, a TO consiste em observar qualquer característica óptica (emissão, tempo de vida, polarização, etc) de uma amostra e sua relação com a temperatura. Na maioria dos trabalhos da literatura, o domínio de estudo da TO são nas regiões espectrais ultravioleta-visível-infravermelho, sem ir muito longe no espectro eletromagnético. Portanto, também chamamos esses estudos de “Termometria de Luminescência” ou “Termometria Luminescente”. Os métodos empregados na literatura para realizar o estudo da termometria óptica (ou luminescente) consiste em observar como a temperatura afeta as características espectroscópicas (largura de linha, deslocamento espectral, tempo de vida, etc.) da amostra. Com isso, é possível estudar a sensibilidade relativa e a resolução térmica (e outros parâmetros) para propor uma configuração em que se possa construir um termômetro óptico. As vantagens desse tipo de termômetro em relação aos termômetros convencionais são, principalmente, a resolução espacial na escala micro- e nanoscópica e a possibilidade de medir a temperatura em regiões mais profundas do que a superfície em tecidos biológicos, materiais orgânicos e inorgânicos. O estudo da TO possui diversos exemplos e aplicações demonstrados na literatura, no entanto, os métodos utilizados (que chamaremos de métodos convencionais) possuem um grau de incerteza considerável em diversos casos. Com isso, neste trabalho usamos uma técnica moderna bem estabelecida computacionalmente para fazer as análises dos dados – *Machine Learning* (ML). O Aprendizado de Máquina (*Machine Learning*) é uma técnica computacional de Inteligência Artificial que tem, dentre seus objetivos, a possibilidade de proporcionar à máquina (computador) o aprendizado de padrões ou não-padrões, a partir de um conjunto de dados de treino, para conseguir fazer previsões ou classificações. Nessa pesquisa, realizamos o estudo da termometria óptica usando microesferas de Nd³⁺:YAG e comparamos os resultados obtidos por métodos convencionais (razão de intensidade e regressão linear múltipla) e pelo aprendizado de máquina. Mostramos que usando o ML é possível reduzir a incerteza na previsão da temperatura na termometria óptica, quando comparado aos métodos convencionais. O algoritmo construído está disponível na seção de apêndices.

Palavras-chave: termometria; optica; aprendizado de máquina; laser.

ABSTRACT

Optical Thermometry (OT) is essentially a field of physics that studies temperature and its parameters using tools from the optics domain, such as lasers, for example. Succinctly, OT consists of observing any optical characteristic (emission, lifetime, polarization, etc.) of a sample and its relationship with temperature. In most works in the literature, the OT field of study are in the ultraviolet-visible-infrared spectral regions, without going too far into the electromagnetic spectrum. Therefore, we also call these studies “Luminescence Thermometry” or “Luminescent Thermometry”. The methods used in the literature to study optical (or luminescent) thermometry consist of observing how temperature affects the spectroscopic characteristics (line width, spectral shift, lifetime, etc.) of the sample. With this, it is possible to study the relative sensitivity and the thermal resolution (and other parameters) to propose a configuration in which an optical thermometer can be built. The advantages of this type of thermometer in relation to conventional thermometers are, mainly, the spatial resolution micro- nano- scale and the possibility of measuring the temperature in regions deeper than the surface in biological tissues, organic and inorganic materials. The study of OT already contains several examples and applications shown in the literature, however, the methods used (which we will call conventional methods) have a considerable degree of uncertainty in several cases. Therefore, in this work we use a well-established modern computational tool to analyze the data – Machine Learning (ML). Machine Learning is a computational technique of Artificial Intelligence that has among its goals the possibility to enable the machine (computer) with the learning of patterns or non-patterns, from a set of training data, to be able to make predictions or ratings. In this research, we performed the study of optical thermometry using Nd³⁺:YAG microspheres and compared the results obtained by conventional methods and by machine learning. We show that using ML it is possible to reduce uncertainty in temperature prediction in optical thermometry, when compared to conventional methods. The constructed algorithm is available in the appendices section.

Keywords: thermometry; optica; machine learning; laser.

LISTA DE FIGURAS

Figura 1 –	Estatística de Busca usando o <i>Google Trends</i>	14
Figura 2 –	Aparelhos para aferição de temperatura	15
Figura 3 –	Principais indicadores utilizados em termometria óptica	20
Figura 4 –	Exemplo de intensidades de emissão dependentes da temperatura	24
Figura 5 –	Busca na <i>Web Of Science</i> para razão de intensidades	26
Figura 6 –	Representação de níveis acoplados e não acoplados termicamente	27
Figura 7 –	Exemplo meramente ilustrativo da influência da temperatura no (a) deslocamento espectral e na (b) largura da banda.	28
Figura 8 –	Algumas subdivisões da Inteligência Artificial	33
Figura 9 –	Busca na <i>Web Of Science</i> pelo tópico: <i>Machine Learning</i>	35
Figura 10 –	<i>Underfitting</i> e <i>Overfitting</i>	38
Figura 11 –	Esquema do aprendizado supervisionado	40
Figura 12 –	Exemplo simples de um modelo de árvore de decisão	42
Figura 13 –	Exemplo simples de uma rede neural	43
Figura 14 –	Rede Neural com duas camadas ocultas	44
Figura 15 –	Busca na <i>Web Of Science</i> para TO com ML	46
Figura 16 –	Esquematização do experimento, espectrômetro e algoritmo	50
Figura 17 –	Aprendizado automatizado e rede neural de múltiplas camadas.	52
Figura 18 –	Árvore de decisão para termometria óptica.	53
Figura 19 –	Distribuição de tamanho das partículas de Nd ³⁺ :YAG (3.5%)	54
Figura 20 –	Amostra usada: Nd ³⁺ :YAG (3.5%)	55
Figura 21 –	Espectros de emissão das partículas das partículas de Nd ³⁺ :YAG (3.5%).	55
Figura 22 -	Níveis de Energia e emissões para a amostra Nd ³⁺ : YAG (3.5%).	56
Figura 23 –	Picos usados para o LIR e MLR.	57
Figura 24 –	Relação entre 2 intensidades	58

Figura 25 –	(a) Intensidades das emissões de 879 nm; (b) Intensidades das emissões de 946 nm; (c) LIR; (d) Temperatura determinada pela LIR (azul) e medida pela câmera térmica (vermelha). Todas as curvas em função do tempo.	59
Figura 26 –	Histograma	60
Figura 27 –	Temperatura calculada x temperatura alvo.	62
Figura 28 –	Temperatura calculada x temperatura alvo no espectro	64
Figura 29 –	Treino e Teste entre os ciclos.	65
Figura 30 –	Razões de intensidades para todas as possíveis combinações.	83
Figura 31 –	Comparação da previsão para diferentes quantidades de combinações no MLR.	87
Figura 32 –	Treino com diferentes arquiteturas da MLP.	87
Figura 33 –	Treino com diferentes arquiteturas da MLP.	88
Figura 34 –	Esboço da MLP usada.	88
Figura 35 –	Resultado usando a CNN.	89

LISTA DE TERMOS (INGLÊS-PORTUGUÊS)

1. Machine Learning – Aprendizado de Máquina
2. Deep Learning - Aprendizado Profundo
3. Artificial Intelligence – Inteligência artificial
4. Bias – Viés
5. Accuracy – Acurácia (o quão correto são as previsões ou resultados em relação ao que se esperava alcançar)
6. Precision – Precisão (o quão próximo os resultados estão entre si)
7. Overfitting – Sobreajuste ou Superajuste
8. Underfitting – Subajuste ou Ajuste Insuficiente
9. Feature – Característica ou Atributo
10. Label – Rótulo ou Etiqueta
11. Training Set – Conjunto de Treinamento
12. Validation Set – Conjunto de Validação
13. Test Set – Conjunto de Teste
14. Model – Modelo
15. Algorithm – Algoritmo
16. Supervised Learning – Aprendizado Supervisionado
17. Unsupervised Learning – Aprendizado Não Supervisionado
18. Neural Network – Rede Neural
19. Convolutional Neural Network – Rede Neural Convolutacional
20. Loss Function - Função de Perda
21. Activation Function - Função de Ativação
22. Batch Size - Tamanho do Lote
23. Epoch - Época (referente a uma passagem completa pelo conjunto de treinamento)
24. Cross-Validation - Validação Cruzada
25. Ensemble Learning - Aprendizado em Conjunto

SUMÁRIO

1 INTRODUÇÃO: TERMOMETRIA E TERMÔMETROS	13
1.1 TERMÔMETROS CONVENCIONAIS	16
1.2 PARÂMETROS QUE DEFINEM/QUALIFICAM UM TERMÔMETRO	18
2 TERMOMETRIA ÓPTICA	19
2.1 PARÂMETROS QUE DEFINEM UM TERMÔMETRO ÓPTICO	19
2.2 MÉTODOS USADOS NA TERMOMETRIA ÓPTICA	22
2.2.1 Única Banda / Única intensidade	24
2.2.2 Razão de Intensidades	25
2.2.3 Largura da Banda e Deslocamento Espectral	27
2.2.3 Tempo de Vida	28
2.2.4 Tempo de subida da emissão	29
2.3 REGRESSÃO LINEAR MÚLTIPLA	29
2.4 APRENDIZADO DE MÁQUINA COMO PROPOSTA PROMISSORA PARA TERMOMETRIA ÓPTICA.....	31
2.5 CONFIABILIDADE E ESTATÍSTICA SOBRE TERMÔMETROS ÓPTICOS	31
3 FUNDAMENTOS DO APRENDIZADO DE MÁQUINA (<i>MACHINE LEARNING</i>) ..	32
3.1 O QUE É O APRENDIZADO DE MÁQUINA	32
3.2 BREVE HISTÓRIA DO MACHINE LEARNING	33
3.2.1 Século XXI	34
3.3 CONCEITOS E PROCEDIMENTOS GERAIS.....	36
3.3.1 Dados de treino e Dados de teste	36
3.3.2 <i>Overfitting</i> e <i>Underfitting</i>	37
3.3.3 Aprendizado Supervisionado x Aprendizado não-supervisionado	38
3.3.3.1 Aprendizado não-supervisionado	38
3.3.3.2 Aprendizado supervisionado	39
3.4 ÁRVORE DE DECISÃO (DECISION TREE)	41
3.5 REDES NEURAS (NEURAL NETWORKS).....	42
4 APRENDIZADO DE MÁQUINA PARA TERMOMETRIA FOTOLUMINESCENTE	46
4.1 TRABALHOS ANTERIORES	47

4.1.1 Nosso trabalho (2023)	49
4.2 O EXPERIMENTO E A INTERPRETAÇÃO COMPUTACIONAL	50
4.3 A AMOSTRA	54
4.4 RESULTADOS E DISCUSSÕES	55
5 PERSPECTIVAS	66
6 CONSIDERAÇÕES FINAIS	67
REFERÊNCIAS	68
APÊNDICE A – CÓDIGOS	73
APÊNDICE B – MATERIAL SUPLEMENTAR	83

1 INTRODUÇÃO: TERMOMETRIA E TERMÔMETROS

Há muito tempo a humanidade observa curiosos fenômenos, como o “aquecer” e o “resfriar”, buscando entender como e através de quais processos os corpos mudam sua sensação de “quente” e “frio”. Para explicar esses fenômenos, nasce o termo “Temperatura” e a ciência chamada *Therme dynamis*, ou, em português, Termodinâmica. No entanto, com mais ênfase nos comportamentos térmicos e nas medidas numéricas da agitação cinética das partículas, entramos no campo da “Termometria” que é apenas um braço da grande área chamada Termodinâmica. De modo didático, o termo “Temperatura” é concebido como sendo o grau de agitação das moléculas ou átomos de um sistema, em que, através disso, pode-se definir as sensações de “quente” e “frio” a partir da energia cinética média que eles possuem. No entanto, a definição mais abrangente é que, de acordo com a física estatística, a temperatura é uma grandeza intensiva no qual seu inverso é a igual a variação da entropia com respeito a energia interna (REICHL, 1999):

$$\frac{1}{T} = \frac{\partial S}{\partial U}. \quad (1)$$

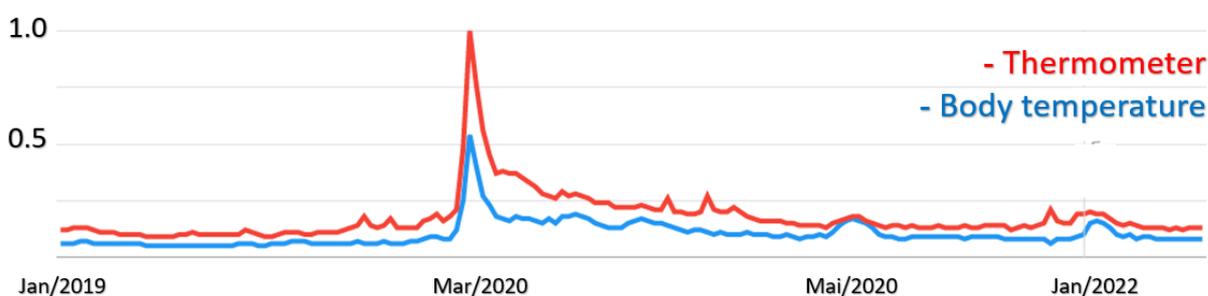
Não é certa a data das primeiras observações técnicas ou o início da termodinâmica, mas nesse contexto se pode citar físicos como Hippokrates (460-370 a.C.), por pensamentos primeiros acerca do temperamento corporal, Klaudios Galenos (133-200 d.C.), por considerações primeiras sobre graus de calor a partir de mistura de água e gelo. Contudo, há um grande consenso na física em dizer que a Termodinâmica, Termologia e Termometria começaram a se solidificar a partir de Galileu Galilei (1564-1642), por conceber o primeiro termoscópio. Esses campos se desenvolveram significativamente com o passar dos anos, e nomes como Celsius, Kelvin, Carnot e outros se tornaram bastante conhecidos por suas contribuições fundamentais. Nesse contexto, a referência (MÜLLER, 2007) propicia uma apresentação rica sobre o desenvolvimento dessa área da física.

Nos dias atuais, medir a temperatura é algo que se tornou indispensável em inúmeras aplicações científicas e tecnológicas. A enorme demanda nas mais diversas situações e o avanço tecnológico permitiu um significativo aperfeiçoamento

dos aparelhos de medição de temperatura. Por exemplo, podem ser encontrados desde termômetros simples (através de contato) até aparelhos de detecção a distância e sem contato (pirômetros).

Um exemplo de como mensurar a temperatura é importante para a humanidade é quando estamos preocupados em alguma doença viral. Em 2020, houve uma grande busca por esses temas devido a pandêmica crise sanitária do Coronavírus iniciada e descoberta no final de 2019 (COVID-19). Realizando uma busca no *Google Trends* para os termos “*thermometer*” e “*body temperature*” (em todos os países), constata-se um crescimento abrupto em meados de março de 2020 (figura 1), época em que a mídia do mundo todo começou a enfatizar sobre o vírus e seus sintomas. Isso mostra a importância que os termômetros possuem para a vida em determinadas situações.

Figura 1 – Estatística de Busca por temas específicos (“*thermometer*” e “*body temperature*”) usando o *Google Trends*. No eixo Y é representada a quantidade de buscas normalizada e o eixo X representa o tempo.



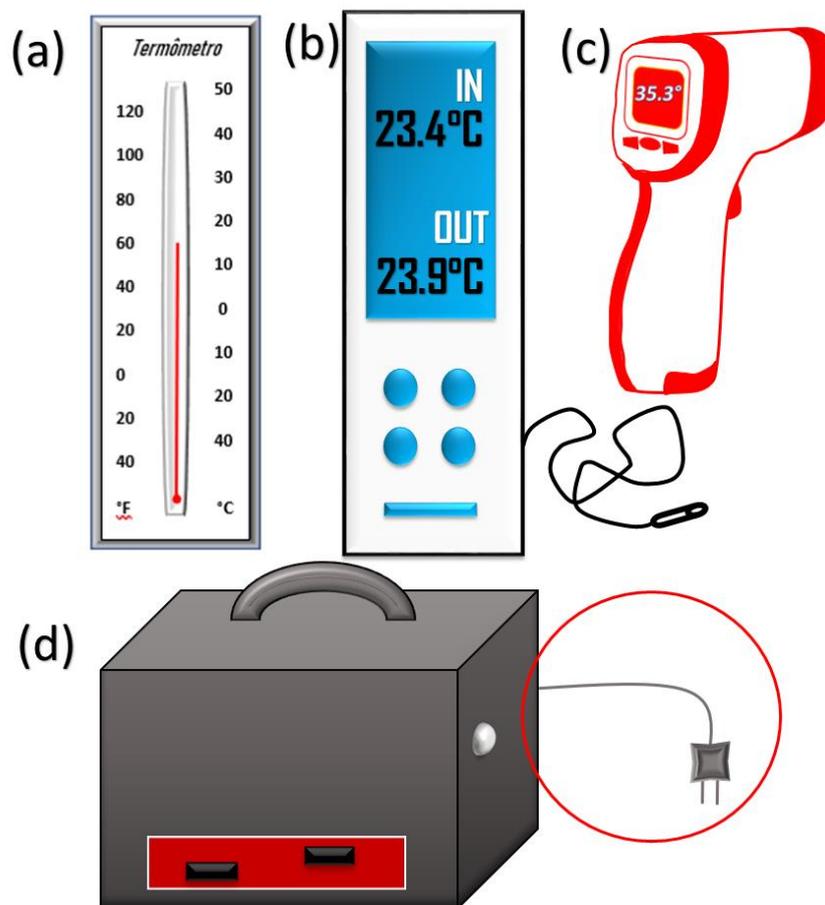
Fonte: Autor (2023).

O uso de termômetros é muito amplo e possui interesse em muitas áreas, levando ao mercado financeiro uma expectativa de giro de 6,86 bilhões de dólares para 2023 (BRITES; BALABHADRA; CARLOS, 2019). De modo geral, os termômetros são amplamente usados na saúde, meteorologia, em laboratórios de pesquisa, laboratórios de ensino, indústrias, monitoramento do clima e, os mais simples, em casa e em sala de aula de ensino básico ou superior.

Na figura 2, ilustramos os mais conhecidos aparelhos de aferição de temperatura. Na figura 2(a) é mostrado o termômetro mais comum de líquido em vidro, em 2(d) é representado um tipo de termopar, em 2(b) tem-se um termômetro digital com termistor e o quarto, em 2(c), é representado um termômetro

infravermelho que atua sem contato. Descreveremos brevemente sobre estes aparelhos mais adiante. Outros aparelhos para medição de temperatura são apresentados na seguinte referência (“Types of Thermometer - Uses of Thermometer - Thermometer Types”, 2019). Todos os termômetros têm 2 elementos em comum: 1) sensor de temperatura, 2) escala numérica de temperatura. Além disso, eles são calibrados e qualificados de acordo com algumas grandezas como sensibilidade relativa e resolução térmica, que trataremos adiante com mais detalhes.

Figura 2 – Exemplos de aparelhos para aferição de temperatura.



Fonte: Autor (2023).

1.1 TERMÔMETROS CONVENCIONAIS

A seguir descreveremos de forma mais detalhada algumas características dos termômetros apresentados na figura 2. Desta forma podemos compreender melhor o funcionamento desses termômetros convencionais, que atuam em objetos e corpos macroscópicos, para depois entender a importância de um micro/nano termômetro óptico, que é tema deste trabalho. O autor também produziu um vídeo discutindo tais termômetros (“PRINCIPAIS TERMOMETROS E COMO FUNCIONAM”, 2022).

- TERMÔMETRO DE LÍQUIDO EM VIDRO

Representado na figura 2(a), é o termômetro mais comum e mais antigo dentre os apresentados. Este termômetro é calibrado a partir de comparação com padrões de temperatura já estabelecidos. O ideal é construí-lo usando líquidos que mantenham constante o coeficiente de dilatação volumétrica mesmo sob grandes variações de temperatura, como o mercúrio, por exemplo. Vale mencionar que a comercialização de termômetros de mercúrio já é proibida em muitos lugares devido aos riscos que o mesmo pode causar a saúde.

O princípio de funcionamento consiste na expansão ou contração do volume do líquido, devido a trocas de calor entre o aparelho e o que se quer medir. Dependendo das características do termômetro, estes podem detectar grandes variações de temperaturas. No entanto, os mais comuns, que são os termômetros clínicos (analógicos ou digitais), medem a temperatura entre 34°C à 43°C, que é o alcance de interesse para monitorar febre e hipotermia, por exemplo.

- TERMÔMETRO DE TERMOPAR

O termopar é um dispositivo em que é gerada uma tensão proporcional à temperatura que ocorre na junção entre dois metais distintos devido ao efeito Seebeck. O efeito Seebeck consiste na observação de que calor gera diferença de potencial elétrico, ou seja, converte-se energia térmica em energia elétrica. Essencialmente, esse efeito ocorre quando se varia a temperatura de uma junção

bimetálica de modo que surge uma diferença de potencial (ddp). Essa ddp ocorre devido a densidade de portadores de carga depender do material usado.

Nessa discussão, existem ainda mais dois efeitos básicos na física que relacionam quantidades elétricas e térmicas: efeito Peltier e efeito Joule. Basicamente são efeitos que explicam outras formas de conversão de energia elétrica em energia térmica. No entanto, enquanto no efeito Joule existe devido ao aquecimento em um condutor, o efeito Peltier ocorre como o inverso do efeito Seebeck, isto é, com a geração de calor devido a passagem de corrente em materiais diferentes. Uma excelente referência nesses temas é (MACIA, 2015).

Ao usar um termopar, essencialmente o que se faz é correlacionar a diferença de potencial elétrico com a temperatura pelo contato da junção bimetálica com o que se deseja medir. Atualmente é muito comum o uso de termopares em pesquisas, laboratórios de ensino e mesmo em ambientes industriais.

- TEMÔMETRO DIGITAL COM TERMISTOR

Esse é semelhante ao Termopar, mas aqui ao invés da relação calor-ddp (como descrito anteriormente), estamos avaliando a relação calor-resistência. O termistor é um componente que pode variar sua resistência quando sofre mudanças de temperatura.

Na ponta deste termômetro (figura 2c) existe um sensor de calor (termistor) que ao receber ou ceder calor, varia sua resistência. O circuito eletrônico então é responsável por receber essa variação na resistência e fornecer o valor (já calibrado) em forma de temperatura no visor do aparelho.

- CÂMERA TÉRMICA (termovisor)

O termovisor de leitura em infravermelho possui um grande diferencial em relação aos outros: funcionamento sem contato. Este tipo de termômetro recebe radiação infravermelha emitida pelos corpos e usando as relações do estudo do Corpo Negro, é possível fornecer a temperatura do corpo visto por esse leitor. Em sua composição, o termovisor possui um termômetro interno calibrado para fornecer uma referência de temperatura conhecida. Ao apontar a câmera para um objeto, um detector de microbolômetro (que faz parte do equipamento) converte a energia da

radiação infravermelha recebida em sinal elétrico. Com base na lei de Stefan-Boltzmann é então possível calcular a temperatura do objeto em mira.

1.2 PARÂMETROS QUE DEFINEM/QUALIFICAM UM TERMÔMETRO

Termômetro é qualquer aparelho capaz de medir a temperatura através de algum sistema termicamente sensível. Nesse momento é oportuno definir que um sensor térmico é qualquer objeto capaz de perceber variações de temperatura. Nesse sentido, alguém perspicaz poderia perguntar “então os humanos são termômetros, já que conseguimos perceber variações de temperatura?” – e a resposta é “sim”. Além disso, somos sensores e capazes de notar a variação de temperatura. Embora nós consigamos perceber variações de temperatura, a qualidade dessa percepção é muito baixa, isso significa que não conseguimos notar variações pequenas de temperatura. Ou seja, ninguém de repente fala “estou sentindo que mudou 1°C de temperatura”, mas é muito comum em determinadas regiões alguém falar “que frio, deve estar fazendo uns 15°C agora” e certamente estará muito próximo do valor real. No entanto, um bom termômetro pode ser capaz de medir variações muito menores que 1°C. Portanto, se vamos falar de Termometria, vale destacar que existem alguns parâmetros que definem e/ou qualificam um termômetro.

De modo abrangente, um termômetro pode ser qualificado por duas grandezas: Sensibilidade e Resolução Térmica. A sensibilidade de um termômetro, como o próprio nome induz, caracteriza o quão sensível é o termômetro para as variações de temperatura. A resolução térmica refere-se a medida de incerteza da temperatura medida. No geral, o objetivo de um termômetro é fornecer a temperatura com o menor grau de incerteza possível. Algumas considerações estatísticas podem ser feitas para calcular o erro dessas medidas (DRAMICANIN, 2018). Porém, nosso trabalho aqui é focar nos termômetros ópticos e por conta disso, os parâmetros que os definem serão detalhados a seguir, no capítulo 2.

2 TERMOMETRIA ÓPTICA

Todos os termômetros apresentados na seção anterior, conhecidos como termômetros convencionais, não são capazes ou não são eficientes para medir temperaturas bem localizadas em objetos de tamanho micro/nanoscópicos. Também não são eficientes ou não são capazes de medir temperatura em regiões profundas (não-superficiais) pois exigem o contato térmico do sistema de medida com a região que se quer medir, ou que a região a se medir seja visível diretamente. Diante dessa dificuldade, há alguns anos que a comunidade científica vem dando uma grande atenção à termometria óptica sem contato. Diferentemente da Termometria convencional, que se usa os termômetros abordados na seção anterior para medir a temperatura, a termometria óptica usa radiação emitida por átomos ou sistemas sensores para monitorar a variação de temperatura. Desse modo, podemos pensar em Termometria Óptica como qualquer processo/experimento do domínio da Óptica capaz de monitorar grandezas relacionadas à Termometria, em especial, a Temperatura.

Um dos braços da Termometria Óptica é a Termometria baseada em Luminescência (ALLISON; GILLIES, 1997; DRAMICANIN, 2018). Essencialmente a Luminescência é a emissão de radiação dentro da região ultravioleta – visível – infravermelho. Nesse trabalho usamos a Luminescência como fator investigativo dos resultados, mas os métodos discutidos e usados são gerais e por isso servem para análise de qualquer âmbito da termometria óptica em qualquer região do espectro eletromagnético.

2.1 PARÂMETROS QUE DEFINEM UM TERMÔMETRO ÓPTICO

Qualquer termômetro que entre no mundo da pesquisa ou no mercado, deve primeiro passar por uma qualificação. Nesse sentido existem alguns parâmetros que definem e qualificam um termômetro óptico.

- Sensibilidade Térmica

O primeiro parâmetro utilizado para definir e qualificar um termômetro óptico é a Sensibilidade Térmica (S). A sensibilidade térmica, ou sensibilidade térmica absoluta,

é a taxa de variação de um indicador termométrico (genericamente designado por Δ) em relação à variação de temperatura:

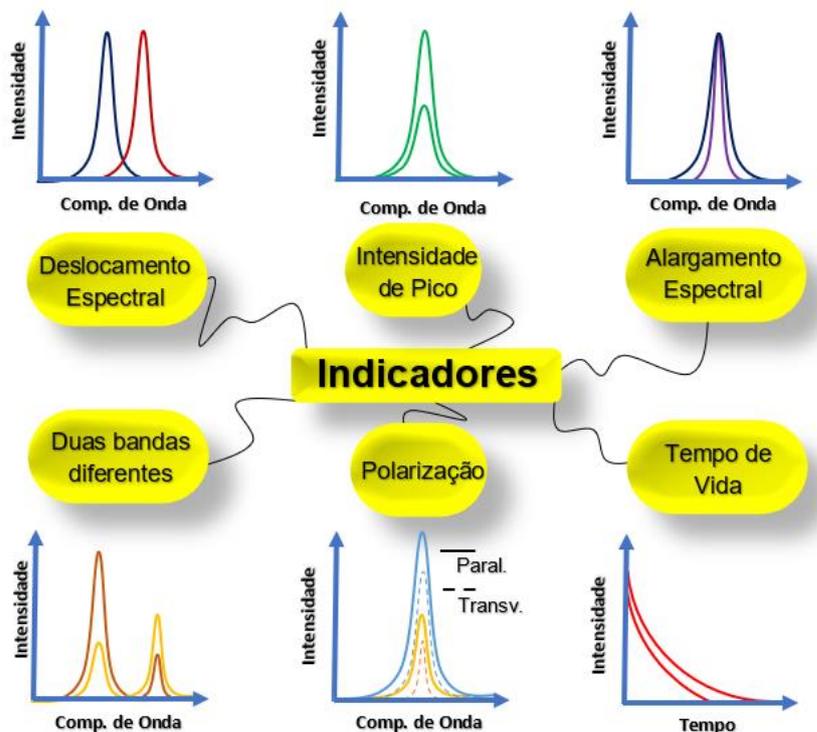
$$S = \frac{\partial \Delta}{\partial T}. \quad (2)$$

Para lidar com diferentes naturezas de termômetros, incluindo aqueles que Δ pode ser descrito como uma função contendo uma constante global multiplicativa, usamos a sensibilidade relativa:

$$S_r = \frac{1}{\Delta} \left| \frac{\partial \Delta}{\partial T} \right|. \quad (3)$$

O indicador termométrico Δ pode ser qualquer grandeza que varie de acordo com a mudança de temperatura. Na espectroscopia, os principais indicadores (BRITES; BALABHADRA; CARLOS, 2019) são mostrados na Figura 3.

Figura 3 – Principais indicadores usados em termometria óptica por luminescência.



Fonte: Autor (2023). Adaptado da referência (BRITES; BALABHADRA; CARLOS, 2019).

Na literatura podemos encontrar diversos artigos que calculam a sensibilidade usando tais indicadores, ou ainda, uma combinação deles. Podemos verificar alguns exemplos na tabela 1 da referência (MATURI et al., 2021).

- Resolução Térmica

Esta grandeza refere-se a medida de incerteza na temperatura. Isso significa que estamos olhando o tamanho do menor passo na variação de temperatura que um termômetro pode medir. Então quanto melhor a resolução térmica de um termômetro, menor é a variação de temperatura que ele pode medir. Essa medida de incerteza depende de muitos fatores, como o sistema experimental de detecção, condições de aquisição dos dados e relação sinal-ruído. Uma forma de definir a resolução térmica (ou incerteza da temperatura) δT é através do desvio padrão de um histograma da diferença entre a temperatura determinada e a temperatura de referência. No entanto, assumindo que não há nenhum fator externo que atrapalhe as medidas e que a incerteza na temperatura só dependa das variações dos indicadores (figura 3), δT é dado por uma expansão em série com respeito ao indicador Δ (BRITES; MILLÁN; CARLOS, 2016):

$$\delta T = \frac{\partial T}{\partial \Delta} \delta \Delta + \frac{1}{2!} \frac{\partial^2 T}{\partial \Delta^2} (\delta \Delta)^2 + \dots + \frac{1}{n!} \frac{\partial^n T}{\partial \Delta^n} (\delta \Delta)^n, \quad (4)$$

em que $\delta \Delta$ é a incerteza relacionado ao indicador Δ . Se os termos de maiores ordens puderem ser suprimidos (BAKER; MCCLESKEY; BAKER, 2005; BRITES; MILLÁN; CARLOS, 2016), então a equação 4 se torna:

$$\delta T = \frac{\partial T}{\partial \Delta} \delta \Delta. \quad (5)$$

Usando (3) em (5):

$$\delta T = \frac{1}{S_r} \frac{\delta \Delta}{\Delta}. \quad (6)$$

As medidas de S_r e δT são as mais comumente reportadas na literatura.

- Resolução espacial e temporal

Um outro ponto importante para definir um termômetro é a questão da resolução espacial e temporal. A resolução espacial refere-se a menor dimensão (ou escala) em que a temperatura pode ser determinada unicamente. O conceito é análogo para a resolução temporal, mas obviamente, aqui é no domínio do tempo.

- Reprodutibilidade e Repetibilidade

Claro que, para o mundo real, todo medidor deve poder ser reproduzível em qualquer lugar, mesmo que mude as condições ambientais. Então, essencialmente, o conceito de repetibilidade é a capacidade de poder reproduzir o mesmo resultado usando o mesmo termômetro. Enquanto que o conceito de reprodutibilidade se refere à capacidade de obter os mesmos resultados para determinado termômetro mesmo que empregado diferentes métodos, ou ainda que os resultados sejam idênticos quando se fabrica dois sistemas idênticos.

- Custo

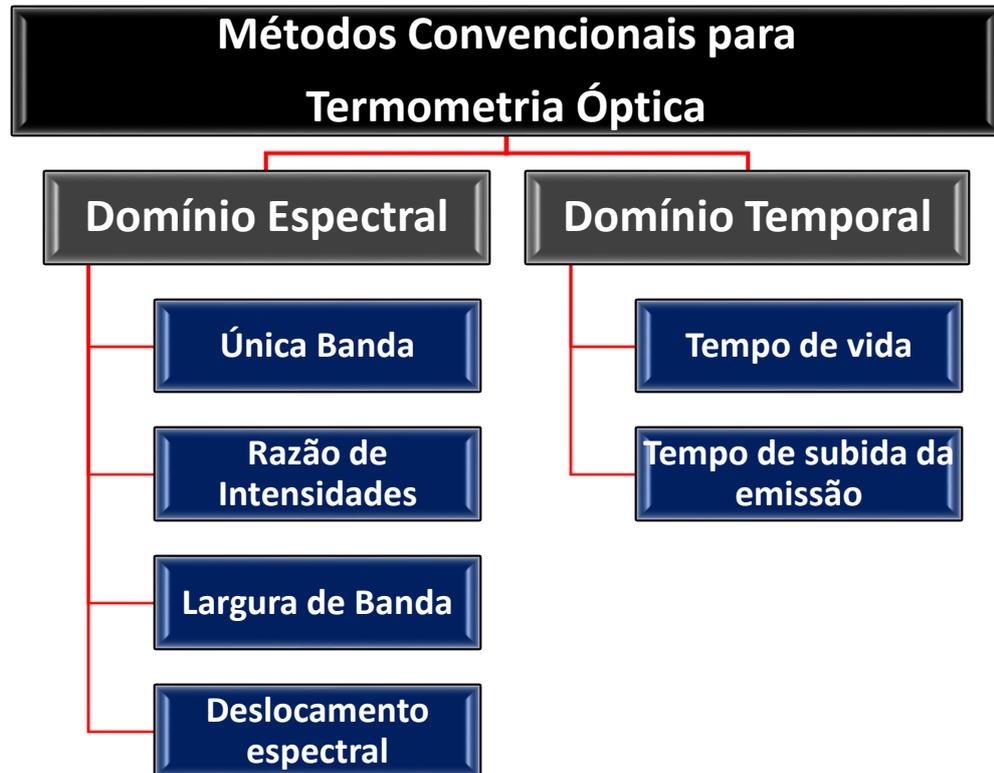
Por último, se estivermos interessados em ingressar no mundo dos negócios, devemos estar atentos as diferentes aplicações. Um sensor pode ser muito caro em algum caso e extremamente barato para outros casos. Isso se torna algo relativo.

2.2 MÉTODOS USADOS NA TERMOMETRIA ÓPTICA

O objetivo da termometria óptica é encontrar materiais e métodos que sirvam para calibrar, propor ou construir um termômetro óptico. Diante da intensa busca por candidatos para um termômetro óptico, na literatura o que se faz, em geral, é avaliar os parâmetros definidos na seção 2.1. Para tal missão, muitos métodos são usados e aqui vamos descrever brevemente sobre eles.

Os métodos empregados na termometria óptica (ou luminescente) podem ser classificados em 2 tipos (fluxograma 1) em relação ao caráter espectral e temporal da observação da luminescência:

Fluxograma 1 – Métodos mais usados para o estudo da TO.



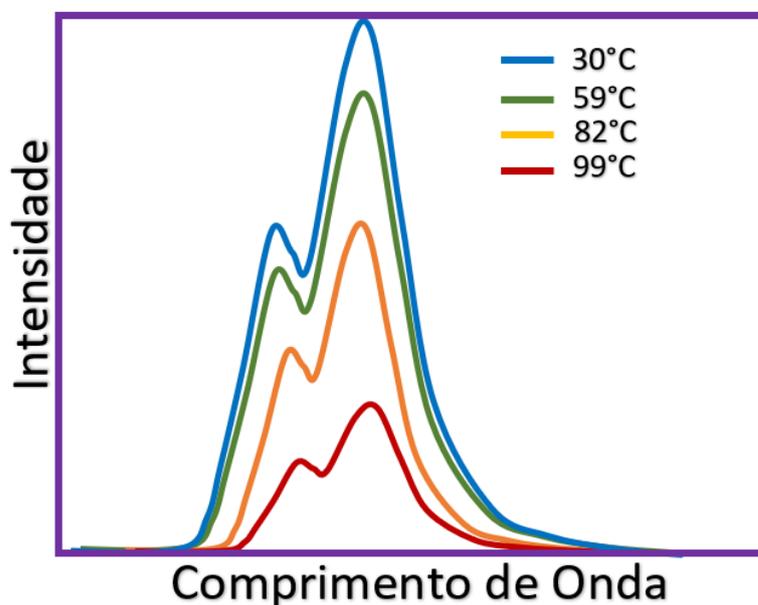
Fonte: Autor (2023).

- Domínio Espectral (na literatura é chamado *Time-Integrated Scheme*) – Nos métodos de domínio espectral, a fonte de excitação é contínua. Estamos interessados em observar a influência direta da temperatura para uma característica do espectro, emissão e/ou intensidades.
- Domínio Temporal (na literatura é chamado *Time-Resolved Scheme*) – Nos métodos de domínio temporal são usadas fontes de excitação pulsadas ou ainda, pode-se utilizar algum tipo de modulador em uma fonte contínua. Estamos interessados em observar a influência direta da temperatura para uma característica temporal da amostra.

2.2.1 Única Banda / Única intensidade

As intensidades de emissão são dependentes da temperatura. De acordo com o material utilizado e a configuração experimental, as intensidades podem aumentar com a elevação da temperatura – ex. (SANTOS et al., 2021) – ou diminuir – ex. (DA SILVA; JACINTO; MOURA, 2020; GALINDO et al., 2021). Na figura 4 (exemplo meramente ilustrativo) vemos como pode ser dada a influência da temperatura para uma banda.

Figura 4 – Exemplo de intensidades de emissão dependentes da temperatura.



Fonte: Autor (2023).

A sensibilidade das intensidades com a temperatura é dependente de vários fenômenos relacionados a redistribuição de população nos níveis de energia. Alguns deles são: relaxação cruzada, decaimentos não-radiativos, processos de conversão de energia assistida por fônons e aparição de defeitos estruturais na rede.

Por conta da sensibilidade das intensidades com a temperatura, trabalhos anteriores mostram que este parâmetro pode ser usado para a termometria óptica (GÁLICO; MAZALI; SIGOLI, 2017; KAMEI et al., 2009). Embora aparente ser interessante usar uma única intensidade (ou única banda) como parâmetro indicador, algumas questões podem tornar esse indicador pouco confiável. Termômetros baseados na intensidade de única transição são muito dependentes

da fonte de excitação e carregam consigo toda a informação do bombeamento de entrada, além de eventuais flutuações ou oscilações experimentais no aparato de detecção. Dessa forma, um método muito usado para compensar tais efeitos consiste em tomar a razão entre duas intensidades ou duas bandas diferentes, assim elimina-se a dependência com a fonte de excitação.

2.2.2 Razão de Intensidades

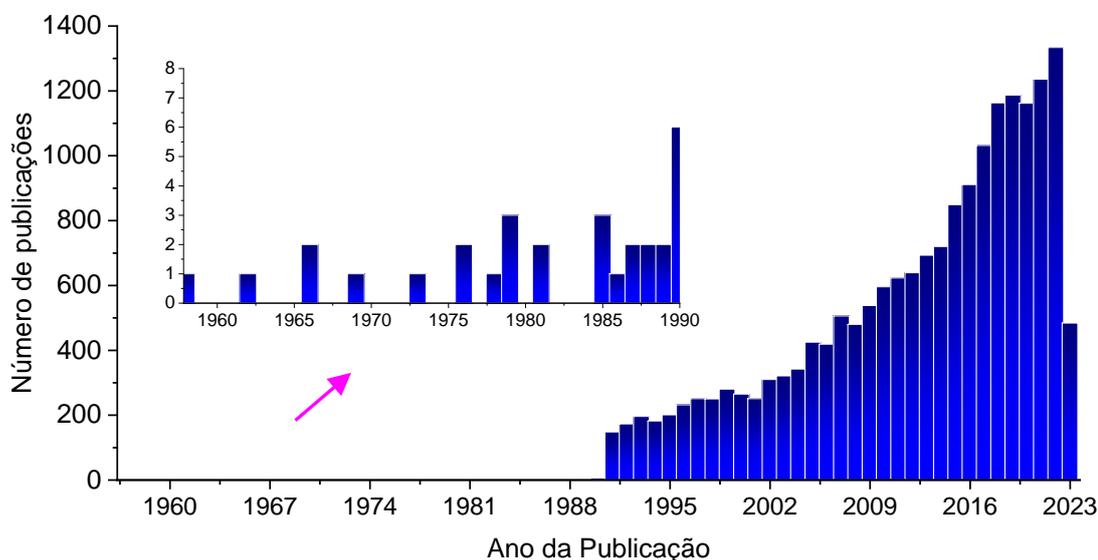
O método da razão de intensidades é conhecido na literatura por diferentes nomenclaturas alternativas como *Fluorescence Intensity Ratio* (FIR), *Luminescence Intensity Ratio* (LIR) ou *Ratiometric Method*. Para simplificar, chamaremos por Razão de Intensidades nessa dissertação. Como o próprio nome intui, o indicador usado agora é dado pela razão entre duas intensidades (bandas de emissão), ou seja:

$$\Delta = \frac{I_1}{I_2}. \quad (7)$$

Com esse método se elimina interferências externas ou de flutuações associadas à fonte de excitação, uma vez que essa mesma informação contida no I_1 e no I_2 se torna unidade na equação (7). É o mais usado na literatura (DRAMICANIN, 2018; WANG et al., 2022). Embora fazer razão de intensidades seja usado há muitas décadas, como por exemplo em (HANSON, 1957), o primeiro trabalho usando este método para análise de termometria foi em publicado 1976 (KUSAMA; SOVERS; YOSHIOKA, 1976).

Realizando uma busca na base de dados da *Web Of Science* usando a busca por tópicos conforme a entrada (*fluorescence intensity ratio OR luminescence intensity ratio OR (ratiometric AND thermometer)*), usando palavras chaves relacionadas, obtemos a Figura 5. Notavelmente ocorre um grande aumento do interesse nesses temas a partir de 1990 e com o passar dos anos consegue-se verificar que a atenção nesses temas tende à um crescimento exponencial.

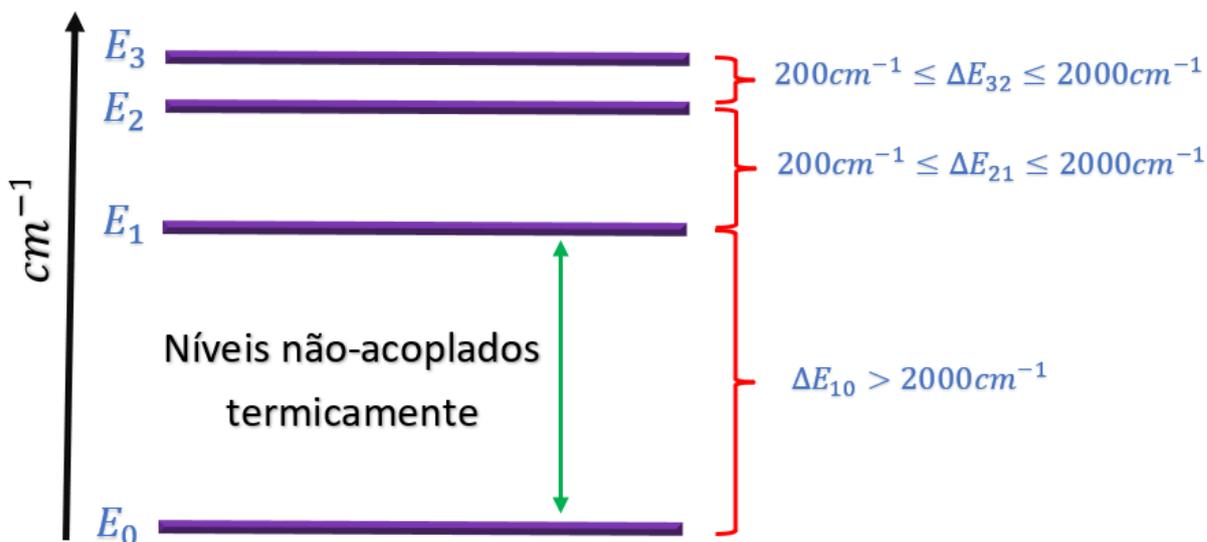
Figura 5 – Busca na *Web Of Science* para razão de intensidades com o tópic: (*fluorescence intensity ratio OR luminescence intensity ratio OR (ratiometric AND thermometer)*).



Fonte: Autor (2023).

O que torna esse método interessante e amplamente usado é que o indicador da equação (7) pode ser interpretado de acordo com a termodinâmica estatística. Isso é, se o par de níveis de energia associados às emissões escolhido for termicamente acoplado, pode ser usada a distribuição de Boltzmann para descrever a curva experimental da razão de intensidades. Caso não sejam acoplados, busca-se um melhor ajuste a curva experimental (LOJPUR; NIKOLIĆ; DRAMIĆANIN, 2014; WANG et al., 2022). Níveis termicamente acoplados são aqueles que possuem uma separação energética no intervalo $200\text{ cm}^{-1} \leq \Delta E \leq 2000\text{ cm}^{-1}$ (BRITES; BALABHADRA; CARLOS, 2019), como representado na Figura 6. Cabe notar que expressar energias em cm^{-1} é bem comum na espectroscopia e basicamente consiste em expressar a energia em termo de $1/\lambda$ com λ expresso em cm . Para converter essa energia para Joules, basta usarmos a relação de Planck $E = hc/\lambda$, em que $1/\lambda$ é a energia. A Figura 6 é uma representação simples de níveis termicamente acoplados.

Figura 6 – Representação de níveis acoplados e não acoplados termicamente.



Fonte: Autor (2023).

Um trabalho pioneiro no Brasil usando este método é (ALENCAR et al., 2004). Outros trabalhos, incluindo brasileiros, usando esse método da razão de intensidades mostraram alguns candidatos a termômetros e nano-termômetros ópticos com altas sensibilidades relativas (DA SILVA; JACINTO; MOURA, 2020; WANG et al., 2022; ZHAO et al., 2020).

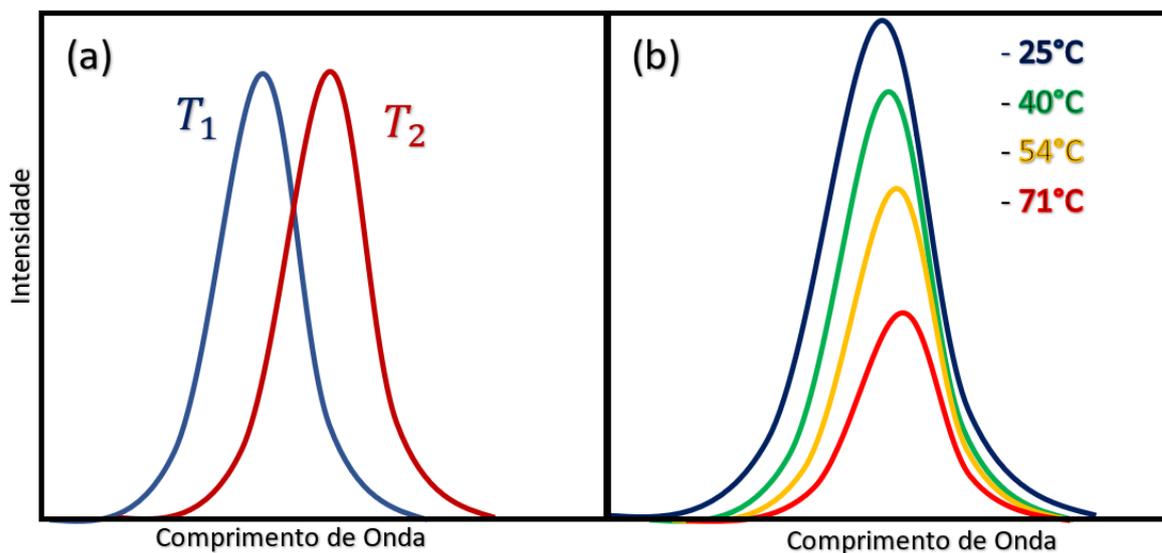
2.2.3 Largura da Banda e Deslocamento Espectral

Uma outra possibilidade para estudar termometria óptica é monitorar a largura da banda ou o deslocamento espectral de uma emissão e verificar sua dependência com a temperatura. Esse caso consiste, essencialmente, em fazer todos os cálculos da termometria óptica usando como parâmetro indicador a largura da banda de emissão ou o deslocamento espectral, como exemplo da Figura 7.

Quando o espectro tem um deslocamento considerável devido a variação de temperatura, então esse é um bom indicador. Por exemplo, considere que na Figura 7(a) seja $T_1 = 2T_2$, em que T_1 possui qualquer valor medido no experimento. Nesse caso, temos distinção razoável entre tais temperaturas usando o deslocamento espectral e isso pode ser usado para propor um termômetro óptico. No entanto, se o deslocamento for pequeno em relação a variação de temperatura, vale a pena

verificar se a variação da largura espectral da banda não se torna um bom candidato para o termômetro óptico – Figura 7(b).

Figura 7 – Exemplo meramente ilustrativo da influência da temperatura no (a) deslocamento espectral e na (b) largura espectral da banda.



Fonte: Autor (2023).

Alguns trabalhos na literatura têm proposto usar essas quantidades como indicadores no campo da termometria óptica (ALLISON; GILLIES, 1997; HERNÁNDEZ-RODRÍGUEZ et al., 2018).

2.2.3 Tempo de Vida

Os métodos relacionados nas subseções 2.2.1 a 2.2.3 são todos do domínio espectral – de acordo com a definição dada no início deste capítulo. No entanto, podemos também ter alguns indicadores no domínio temporal, como por exemplo o tempo de vida ou tempo de decaimento partindo de um estado excitado. Após receber uma excitação pulsada ou modulada (o caso modulado é com fonte de bombeio contínua usando, por exemplo, um *chopper*), os átomos vão para o estado excitado e, em seguida, a população decai num período de tempo chamado tempo de vida e a intensidade de luminescência gerada pelos decaimentos radiativos ocorrem durante um período de tempo. Esse parâmetro possui dependência com a temperatura e também pode ser usado como indicador para propor candidatos a

termômetros ópticos (DRAMICANIN, 2018; GÁLICO; MAZALI; SIGOLI, 2017; YAP et al., 2008).

2.2.4 Tempo de subida da emissão

Ainda pode-se considerar também como indicador o tempo de subida da emissão – período que vem antes do tempo de decaimento e é necessário para a população do estado emissor – embora geralmente seja mais curto do que o tempo de decaimento (DRAMICANIN, 2018; KHALID; KONTIS, 2009).

2.3 REGRESSÃO LINEAR MÚLTIPLA

Recentemente, alguns trabalhos mostraram que, usando o método de regressão linear múltipla (RLM) na Termometria Óptica, pode-se obter um melhor resultado para a sensibilidade relativa e a resolução térmica (ALREBDI et al., 2023; MATURI et al., 2021), em comparação aos métodos descritos anteriormente.

O método da RLM consiste em expandir a temperatura como uma série de todos os indicadores (Figura 3) que variem linearmente com a temperatura:

$$T = c_0 + c_1\Delta_1 + c_2\Delta_2 + \dots + c_n\Delta_n = c_0 + \sum_p c_p\Delta_p, \quad (8)$$

em que c_0 é a soma dos coeficientes lineares relacionados a cada indicador e c_p são os coeficientes angulares. Ou seja, se um termômetro óptico mostra um ou mais indicadores que variam linearmente com a temperatura, podemos usar os coeficientes linear e angular de cada parâmetro e obter a temperatura de cada ponto através da Equação (8).

Considere que a temperatura dependa de um único parâmetro indicador, ou seja, da equação (8) temos $T = c_p\Delta_p$, logo a diferencial é:

$$dT = c_p d\Delta_p \rightarrow c_p = \frac{dT}{d\Delta_p}; \quad (9)$$

usando a relação (3) com (9) obtemos:

$$S_r = \frac{1}{\Delta_p} \frac{d\Delta_p}{dT} = \frac{1}{c_p \Delta_p}. \quad (10)$$

A Equação (10) pode ser expandida vetorialmente para mais parâmetros podendo ser reescrita como (MATURI et al., 2021):

$$S_r = \sqrt{\sum_p \left(\frac{1}{\Delta_p} \left| \frac{\partial \Delta_p}{\partial T} \right| \right)^2}, \quad (11)$$

ou, ainda:

$$S_r = \sqrt{\sum_p (c_p \Delta_p)^{-2}}. \quad (12)$$

Observa-se que, embora esse método tenha melhorado a performance dos parâmetros ópticos em até 10 vezes (MATURI et al., 2021), isso só é válido se os parâmetros variarem linearmente com a temperatura. Em princípio o método pode ser estendido facilmente à parâmetros que variam de forma não-linear com a temperatura. Por ser uma soma de vários parâmetros que variam com a temperatura, esse método apresenta melhor sensibilidade e precisão. Experimentalmente é difícil lidar com ruídos e flutuações experimentais independentemente do método utilizado. No entanto, ainda é possível diminuir os efeitos indesejados e aprimorar os resultados da termometria óptica. Essa forma é o tema deste trabalho, que começaremos a discutir na sequência.

2.4 APRENDIZADO DE MÁQUINA COMO PROPOSTA PROMISSORA PARA TERMOMETRIA ÓPTICA

O objetivo geral deste trabalho é usar o Aprendizado de Máquina (AM) – *Machine Learning* (ML) – no âmbito da termometria óptica. Dentre os objetivos específicos, pretendemos, principalmente, mostrar que através desse método é possível diminuir o erro associado à previsão da temperatura, comparado aos métodos convencionais descritos nos itens anteriores. Uma descrição mais detalhada sobre o aprendizado de máquina está no capítulo 4 desta dissertação.

Uma das motivações em usar o ML é evitar erros associados a “escolha do melhor ajuste” da curva experimental, algo que é feito em outros métodos. Nos capítulos 4 e 5 faremos uma explanação sobre isso.

2.5 CONFIABILIDADE E ESTATÍSTICA SOBRE TERMÔMETROS ÓPTICOS

Um termômetro é considerado confiável se atender aos critérios de repetibilidade e reprodutibilidade, estabelecidos na seção 2.1. Em especial, atender o critério de repetibilidade é o teste preliminar para qualquer proposta experimental de termômetros ópticos. A repetibilidade pode ser determinada através dos dados experimentais (BRITES; MILLÁN; CARLOS, 2016). Além disso, deve-se garantir uma boa reprodutibilidade também, como por exemplo nas seguintes referências (BRITES; MILLÁN; CARLOS, 2016; LAIA et al., 2022).

Além de ser importante atender os critérios mencionados, todo experimento deve ser acompanhado de uma preocupação com as flutuações, o que torna relevante estudar como a precisão da medida de temperatura de uma amostra luminescente depende das intensidades das emissões comparado com medidas de ruído e sinal de fundo (VAN SWIETEN; MEIJERINK; RABOUW, 2022).

Por fim, usar o ML é uma ótima ferramenta de inteligência artificial que podemos aplicar para diminuir incertezas ligadas a flutuações ou perturbações inerentes aos equipamentos ou experimentos, além de ser possível “driblar” a busca pelo o melhor ajuste teórico ou pela expressão matemática que deveria se ajustar aos pontos experimentais. Então, antes de descrever como usamos o ML para tal propósito, vamos usar o próximo capítulo para introduzi-lo.

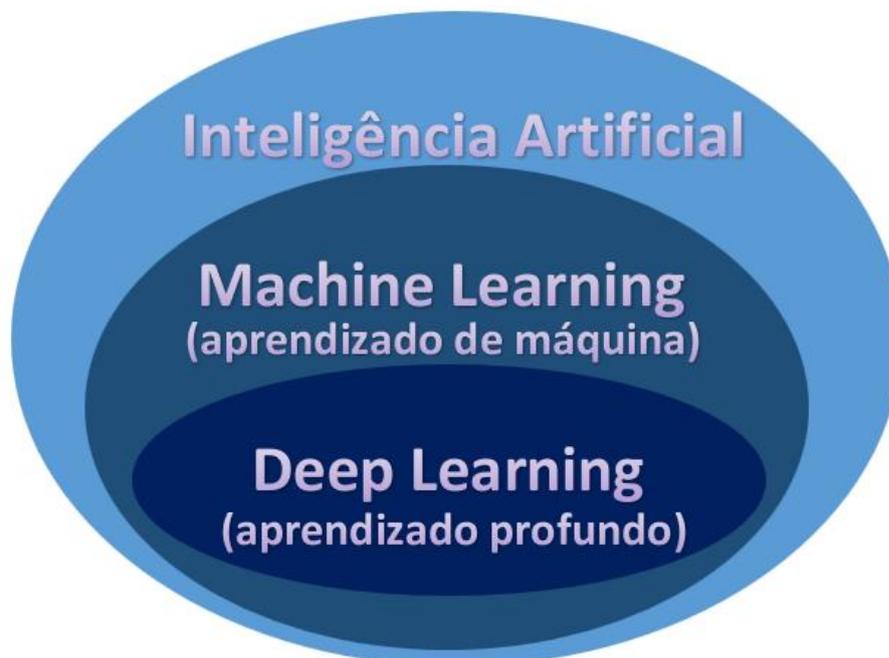
3 FUNDAMENTOS DO APRENDIZADO DE MÁQUINA (*MACHINE LEARNING*)

3.1 O QUE É O APRENDIZADO DE MÁQUINA

O Aprendizado de Máquina – *Machine Learning* (ML) – é uma metodologia computacional que usa algoritmos matemáticos e estatísticos para analisar dados e aprender padrões e relações diretamente, isto é, similar a um humano que aprende através de observação. Com o ML é possível programar a máquina para aprender com uma base de dados e torná-la uma ferramenta capaz de fazer previsões e/ou tomar decisões para o futuro.

No campo da ciência de dados, podemos destacar que o ML é uma subárea da Inteligência Artificial (IA) – Figura 8. A Inteligência artificial é qualquer construção técnica, programa ou método que deixa a máquina capaz de imitar qualquer tarefa humana (exemplos simples: cálculos, preenchimento de planilhas, assistentes virtuais, chatbots). Já o ML é algo mais específico dentro da IA, da forma como descrevemos no primeiro parágrafo desta seção. Alguns exemplos conhecidos de ML são o reconhecimento facial, recomendação de produtos e suporte ao cliente. Por último, o *Deep Learning* é uma metodologia mais recente da IA que se baseia em redes neurais, propondo imitar o cérebro humano. Para este trabalho, aplicamos ML e *Deep Learning* como proposta para calcular a resolução térmica em termômetros ópticos. Portanto, é importante apresentar uma breve introdução sobre tais conceitos.

Figura 8 – Algumas subdivisões da Inteligência Artificial.



Fonte: Autor (2023).

3.2 BREVE HISTÓRIA DO MACHINE LEARNING

Em 1950, o pioneiro cientista da computação Alan M. Turing, publicou um artigo (TURING, 1950) em que fazia a seguinte indagação: “*Can machines think?*”. Nesse artigo, Turing levanta uma hipótese sobre a ideia da máquina “aprender”, semelhante aos humanos. Esse artigo é considerado o marco inicial da era do ML. Turing propôs um jogo, “o jogo da imitação”, que considera a hipótese de que se as máquinas conseguissem imitar ou convencer os humanos de que não eram máquinas, então estas teriam alcançado a chamada inteligência artificial.

Em 1957, Frank Rosenblatt desenvolveu a primeira rede neural (*neural network*) para computadores, chamada de “Perceptron”. O algoritmo do perceptron foi desenvolvido para classificar dados de entrada em grupos. A demonstração desse algoritmo ocorreu pela primeira vez em 1958, quando um IBM 704 – computador desenvolvido para cálculos de larga escala e que pesava em torno de 5 toneladas – que tinha sido alimentado por vários cartões perfurados, após mais de 50 tentativas, conseguiu diferenciar as cartas com furos à esquerda, daquelas com furo à direita (“*Professor’s perceptron paved the way for AI – 60 years too soon*”, 2019).

Em 1959, Bernard Widrow e Marcian Hoff criaram 2 modelos de redes neurais chamados “Adeline”, feito para detectar padrões binários, e “Madeline”, feito para eliminar eco em linhas de ligações telefônicas. As diferenças desses modelos de redes neurais pode ser encontradas com uma análise mais detalhada em (RASCHKA, 2022).

Alguns anos depois, em 1967, foi desenvolvido um outro algoritmo de ML chamado “algoritmo do vizinho mais próximo”, que essencialmente permite computadores usar padrões básicos reconhecidos. Este algoritmo pode ser usado para problemas de regressão ou classificação, descritos mais adiante.

Avançando um pouco mais no tempo, em 1981, Gerald Dejong introduziu o conceito de aprendizado de máquina baseado em explicação (DEJONG, 1981) – *explanation based learning* (EBL) – no qual o computador analisa dados e cria regras gerais para descartar informações irrelevantes para eles.

Durante a última década do século XX muitos algoritmos de ML surgiram junto aos avanços tecnológicos da época, além de aperfeiçoar os algoritmos inventados até então (rede neural, árvore de decisão, EBL, etc.). Cientistas começaram a criar programas de computadores para analisar grandes quantidades de dados, descrever conclusões e “aprender” com os resultados. Com tudo isso, a nova era da tecnologia vinha se solidificando cada vez mais.

3.2.1 Século XXI

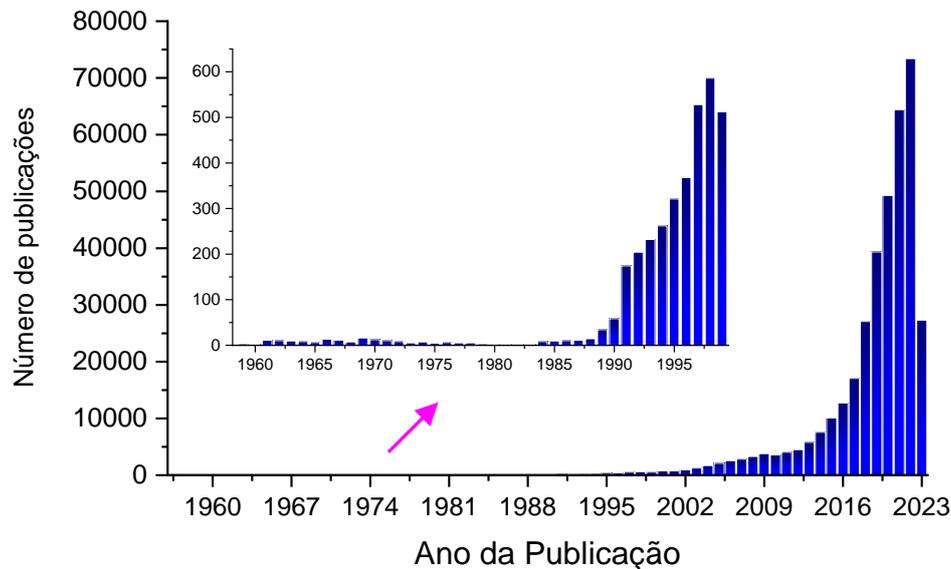
Em 2002, usando uma combinação de ML, processamento de idiomas e técnicas de recuperação de informações, a *IBM Watson* venceu 2 campeões mundiais no jogo do *Jeopardy*.

O interesse nesse campo da inteligência artificial veio crescendo e muitos cursos em diversas universidades começaram a colocar conteúdos de IA e aprendizado de máquina dentro dos componentes curriculares, projetos de pesquisa ou atividades extracurriculares. Nesse sentido, a quantidade de pesquisas e publicações veio crescendo exponencialmente, como representada na Figura 9.

Em 2016, o programa *AlphaGo* do *Google* se tornou o primeiro programa de computador a vencer o campeão mundial em jogos de tabuleiro, usando uma técnica de árvore de decisão no algoritmo de ML.

Atualmente o ML é usado de forma ampla para diversas finalidades, tais como análise e coleta de dados, detecção de *spam* nos serviços de mensagens eletrônicas, detecção de fraudes, leitura biométrica, reconhecimento de voz, reconhecimento facial, organização de informações, recomendação de produtos, sugestão de atrativos em redes sociais, análise de imagens, análise de textos, automatização de sistemas, tradução de idioma e proteção de dados. A eficiência do ML tem se revelado positiva para empresas, humanidade e comunidade científica, de tal forma que o número de publicações tem crescido ao longo dos últimos anos.

Figura 9 – Busca na *Web Of Science* pelo tópico *Machine Learning*.



Fonte: Autor (2023).

Para mais informações sobre fundamentos e aplicações do ML em geral, o leitor pode consultar as referências (ANGRA; AHUJA, 2017; JOHRI; VERMA; PAUL, 2020). Em especial, foi publicado em 2019 um artigo de revisão sobre o aprendizado de máquina para físicos (MEHTA et al., 2019). Nesse artigo os autores, além de descrever conceitos fundamentais de ML, exploram diversos modelos e algoritmos que podem ser implementados em rotinas de *machine* e *deep learning*.

E no campo da termometria óptica, qual o estado atual da aplicação do ML? – A resposta está no próximo capítulo. Por enquanto, vamos abordar alguns pontos importantes sobre os fundamentos do Aprendizado de Máquina.

3.3 CONCEITOS E PROCEDIMENTOS GERAIS

Não é o objetivo desta dissertação discutir extensivamente aspectos da ciência de dados. No entanto, parece adequado ao menos introduzir conceitos básicos sobre o ML, tendo em vista que esta é uma ferramenta importante para esta dissertação. Então, vamos abordar aqui alguns conceitos chaves e o procedimento geral de programação do ML adotado neste trabalho.

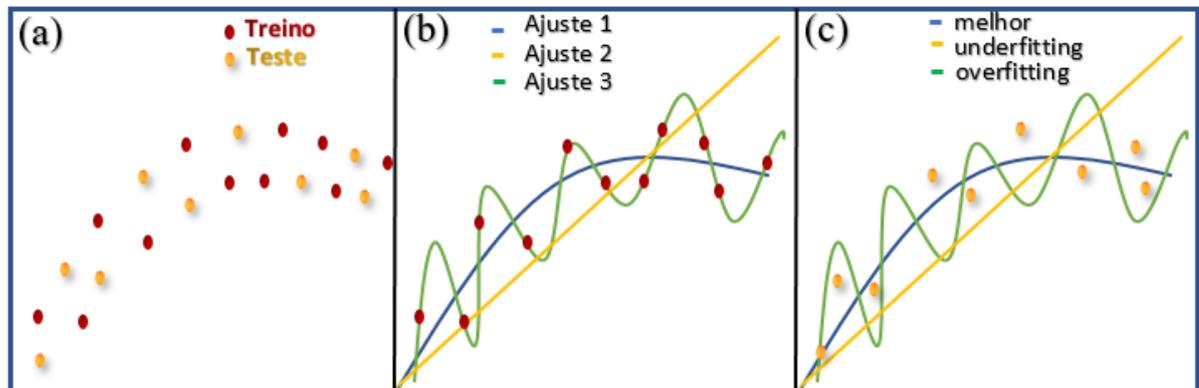
3.3.1 Dados de treino e Dados de teste

O objetivo de um programa de aprendizado de máquina é, como o próprio nome sugere, fazer a máquina aprender. Para isso, é necessário ter um conjunto de dados para treinar a máquina. Com esse conjunto de dados, deve-se então separar uma fração dos dados para treinar a máquina e o restante para testar o modelo. O aprendizado ocorre com os dados de treino e a verificação de quão bom ele está ocorre com os dados de teste. Por exemplo, vamos supor que o objetivo é classificar vinhos em vinho branco ou vinho tinto. Nesse caso, pode-se fornecer ao computador uma série de informações tais como acidez, quantidade de açúcar, densidade, pH, quantidade de álcool e qualquer outra informação relevante que sirva para diferenciar um vinho branco do vinho tinto. Forneceremos um conjunto de dados (dados de treino) para o computador com essas características e nesses dados iremos revelar para o computador quais são vinhos brancos e quais são vinhos tintos. Com isso, o melhor algoritmo do aprendizado de máquina deve descobrir traços, características ou padrões que o ajudem a acertar o máximo de vezes possível para dados futuros (dados de teste). Com o algoritmo pronto, usamos agora os dados que ele nunca viu – os dados de teste – sem revelar se é vinho tinto ou branco, fornecendo apenas as características, e assim vemos qual é a precisão do ML em acertar o resultado exato. Isso se chama acurácia e teste do código. Por exemplo, após o treino, digamos que mostramos ao algoritmo 100 vinhos que tínhamos escondido do treino (ou seja, dados de teste), então se a máquina analisar as características desses vinhos, acertar 91 deles e errar 9, significa que a *accuracy* (acurácia) do nosso algoritmo foi de 91%. Dessa forma podemos medir a performance de um programa de aprendizado de máquina.

É importante destacar aqui que não é uma boa prática usar o mesmo dado do treino, durante o teste; obviamente o programa já conhece aquela estrutura e seria muito alta a probabilidade de acertar e isso não estaria simulando uma situação real. Na prática, queremos treinar a máquina para que ela faça previsões de dados jamais vistos, de dados futuros. Além disso, quando se trata desse tipo de algoritmo, queremos sempre que o programa preveja situações gerais, não limitando apenas as situações vista no treinamento. Para isso, devemos tomar cuidado com o *overfitting* (sobreajuste) e o *underfitting* (subajuste).

3.3.2 *Overfitting* e *Underfitting*

Suponha que tenhamos um conjunto de pontos e queremos encontrar o melhor ajuste, ou seja, a melhor curva que sirva para prever pontos desconhecidos. Por hipótese, vamos considerar todos os pontos da Figura 10 (a) como sendo dados reais de alguma medida ou observação. No processo de ML queremos separar os dados em conjuntos de treinamento (vermelho) e teste (amarelo), conforme dito anteriormente. Inicialmente, vamos usar apenas os pontos vermelhos (dados de treino) e digamos que encontramos 3 ajustes possíveis mostrados na Figura 10 (b). À primeira vista poderíamos pensar “o ajuste 3 está perfeito” e então o modelo já poderia ser aplicado para prever novos pontos. No entanto, o processo de aprendizagem de máquina requer a etapa de teste, ou seja, queremos saber se a curva de ajuste proposta também consegue descrever os dados de teste, que não foram utilizados na etapa de treino. Observa-se que para os dados de teste a curva verde – ajuste 3 – já não é boa como foi antes. Isso significa que caímos num problema de *overfitting*, ou seja, os dados se ajustam demais aos dados do treino, mas descrevem mal para dados futuros.

Figura 10 – *Underfitting e Overfitting*.

Fonte: Autor (2023).

Vamos observar agora que o ajuste 2, uma reta, sempre foi ruim, tanto para os dados de treino quanto para os dados de teste. Nesse caso, chamamos isso de *underfitting*, que significa que o ajuste proposto ele não se ajusta bem nem nos dados de treino e nem nos de teste. Mas há ainda o caso do ajuste 1 em que o ajuste pode não ter sido perfeito para os dados, mas se aproxima bem de dados nunca observados diretamente pelo modelo, indicando que deve ser possível generalizar tais resultados para dados futuros.

3.3.3 Aprendizado Supervisionado x Aprendizado não-supervisionado

Em algoritmos de aprendizado de máquina, existem 2 maneiras de se trabalhar com os dados: aprendizado supervisionado e aprendizado não-supervisionado. Nesta dissertação estaremos trabalhando com o caso de aprendizagem supervisionada, que, portanto, será discutida com mais ênfase.

3.3.3.1 Aprendizado não-supervisionado

O aprendizado não-supervisionado é aquele em que fornecemos os dados sem rótulo, ou seja, temos as características, mas não sabemos do que se trata exatamente. Por exemplo, poderíamos ter um conjunto de dados de características de várias plantas diferentes, mas não fornecemos as espécies. Poderíamos

fornecer, por exemplo, tamanho do caule, característica das folhas, frutos, cores, mas não indicamos qual é de fato a espécie correspondente de cada conjunto de características. Nesse sentido, a máquina terá que encontrar as melhores características e padrões que possa agrupar as plantas de forma a distinguir as espécies. Uma matéria interessante sobre aprendizado não-supervisionado é (“*Unsupervised Machine learning - Javatpoint*”, [s.d.]).

3.3.3.2 Aprendizado supervisionado

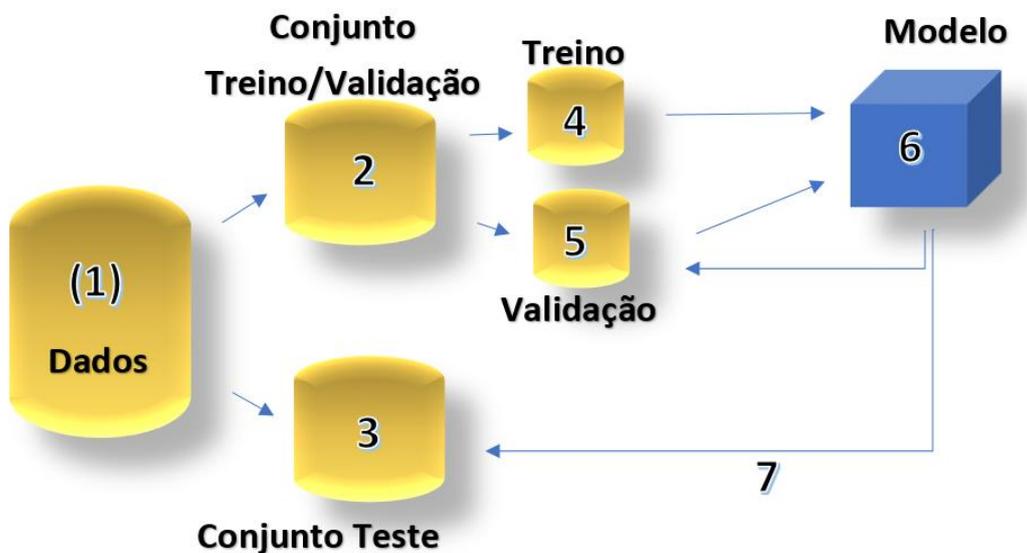
Nesse caso, a máquina é informada sobre a resposta desejada para cada conjunto de dados fornecido. Ou seja, o computador vai receber um conjunto de características ou *inputs* quaisquer e, além disso, irá receber também a que se refere esses *inputs*. Por exemplo, se temos várias imagens de objetos diferentes rotuladas em “geladeira”, “celular”, “computador” ... a máquina vai receber não só a imagem, mas também vai saber do que se trata, pois elas são rotuladas. Por outro lado, no aprendizado não-supervisionado, a máquina receberia apenas a imagem sem saber se é uma geladeira, celular... Em resumo, sempre que a resposta for previamente conhecida, o treinamento da máquina será chamado de aprendizado supervisionado.

No aprendizado supervisionado, o processo de aprendizagem de máquina pode ser esquematizado conforme a Figura 11:

- (1) O conjunto total de dados disponíveis é dividido em 2 partes.
- (2) Dados de treino – são os dados que a máquina terá acesso para modelar o algoritmo de ML. Podemos ainda dividir os dados de treino em 2 partes e usar uma delas para o treino (parte 4) e a outra para atualizar o modelo (parte 5), o que chamamos de validação. É importante notar que a etapa de validação permite verificar a adequação do modelo aos dados apresentados enquanto se varia os hiperparâmetros, evitando problemas de *overfitting*.
- (3) Dados de teste – são os dados que a máquina não teve contato durante a formação do seu algoritmo e é com ele que vamos testar o algoritmo produzido na etapa 6.
- (4) Dados do primeiro treino.

- (5) Dados do segundo treino (validação) após a modelagem feita com a parte 4.
- (6) O modelo que o ML vai desenvolver, que depende de cada problema específico.
- (7) Com o modelo pronto e atualizado, a etapa final é verificar a precisão do programa realizando testes com os dados que tínhamos separado antes.

Figura 11 – Esquema do aprendizado supervisionado.



Fonte: Autor (2023).

Dentro da classe desses algoritmos de ML, existem dois tipos de análise dos problemas:

- Regressão – Nesses problemas o interesse é prever um valor numérico, ou seja, estamos querendo saber uma quantidade específica que pode ser representado por um número (real ou complexo). Exemplos: preço de um produto, força elétrica de um conjunto de cargas, temperatura de um ponto, idade das pessoas.
- Classificação – Nesse tipo de problema, não queremos saber valores numéricos específicos, mas sim a classe de cada item. Ou seja, o interesse aqui é organizar e classificar as coisas de acordo com características, grupos e padrões. Exemplos: tipos de vinho, modelos de carro, tipo de átomo.

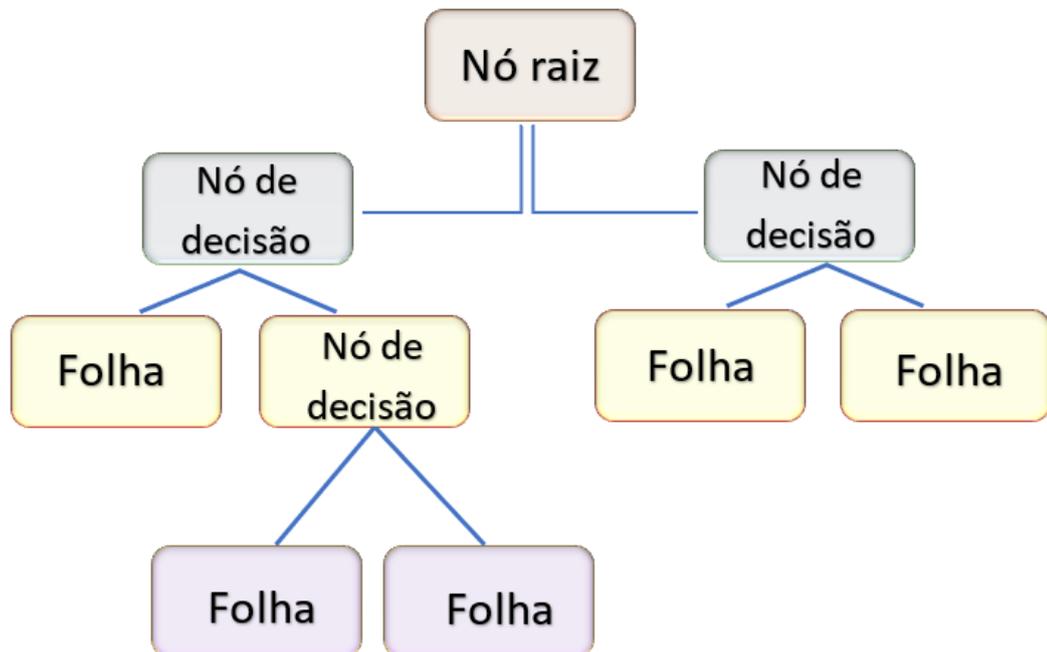
Existem muitas técnicas diferentes para desenvolver um bom algoritmo de ML. Dentre elas, algumas das mais conhecidas são: árvore de decisão, vizinhos próximos, máquina de vetores de suporte, análise de componentes principais e método dos *ensembles*. Neste trabalho usamos especialmente o algoritmo de *deep learning*, que é a rede neural *perceptron*. Mas, colocamos também a possibilidade de realizar com a árvore de decisão (ver apêndice A). Portanto, vamos introduzir brevemente esse algoritmo e em seguida falaremos da rede neural que é o principal algoritmo aqui.

3.4 ÁRVORE DE DECISÃO (DECISION TREE)

O algoritmo de árvore de decisão estabelece uma construção de tomadas de decisão pela máquina e constrói nós, raízes e folhas (figura 12). Em cada nó é gerado uma tomada de decisão de modo a levar cada conjunto de escolhas para caminhos diferentes e resultados finais (folhas) diferentes.

Na árvore de decisão, o algoritmo tem o trabalho de encontrar os nós que serão encaixados em cada posição. Quem será colocado como nó raiz? Depois, quais serão as regras para definir os subsequentes? Dependendo do algoritmo, pode ser considerado cálculos matemáticos de entropia, índice Gini ou até mesmo de forma aleatória para tomar essas decisões. Árvores de decisão são algoritmos já bem estabelecidos no aprendizado de máquina (KINGSFORD; SALZBERG, 2008).

Figura 12 – Exemplo simples de um modelo de árvore de decisão.



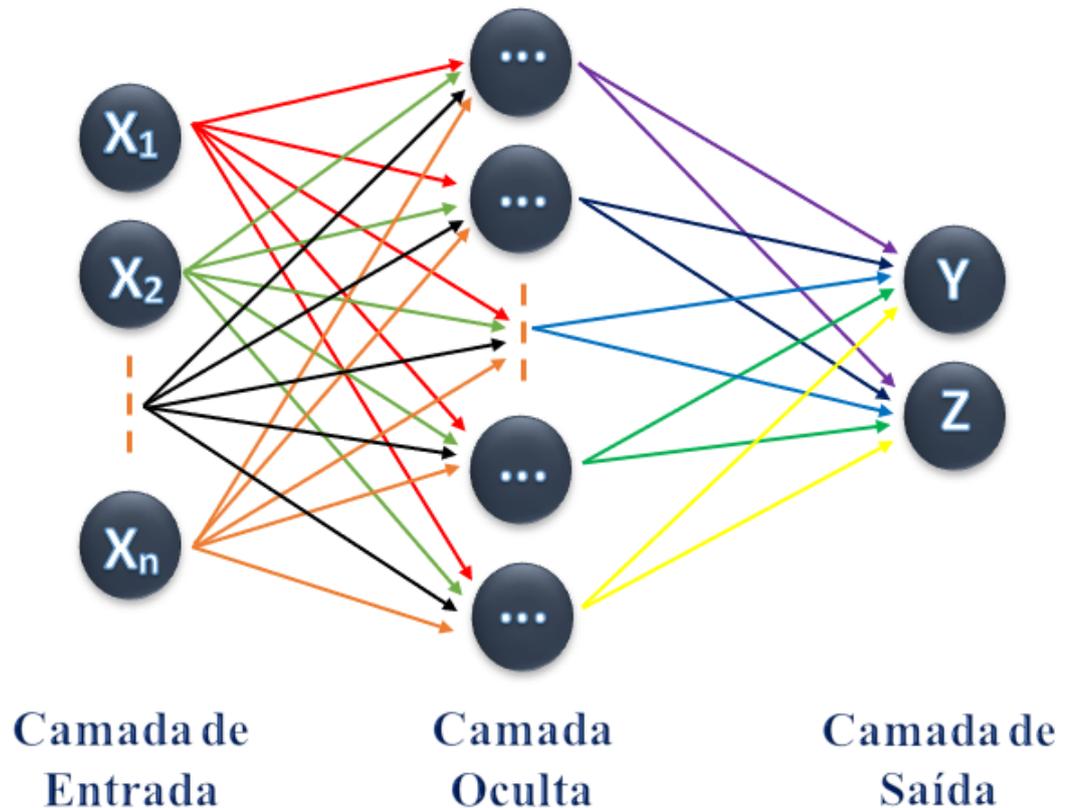
Fonte: Autor (2023).

É claro que, para cada problema, a árvore é construída de maneira diferente e dependendo da complexidade da situação pode conter milhares de nós antes de chegar na decisão final (folha).

3.5 REDES NEURAIIS (NEURAL NETWORKS)

O aprendizado profundo, *deep learning*, é a parte da ciência de dados, subconjunto do campo da inteligência artificial, já brevemente descrito no início do capítulo 4, em que o algoritmo é construído no intuito de imitar os neurônios de um cérebro humano. Existem diversas formas de se construir uma rede neural artificial e muitas das possibilidades são discutidas em (SHARKAWY, 2020). Vamos representar aqui uma das topologias mais simples na Figura 13.

Figura 13 – Exemplo simples de uma rede neural.



Fonte: Autor (2023).

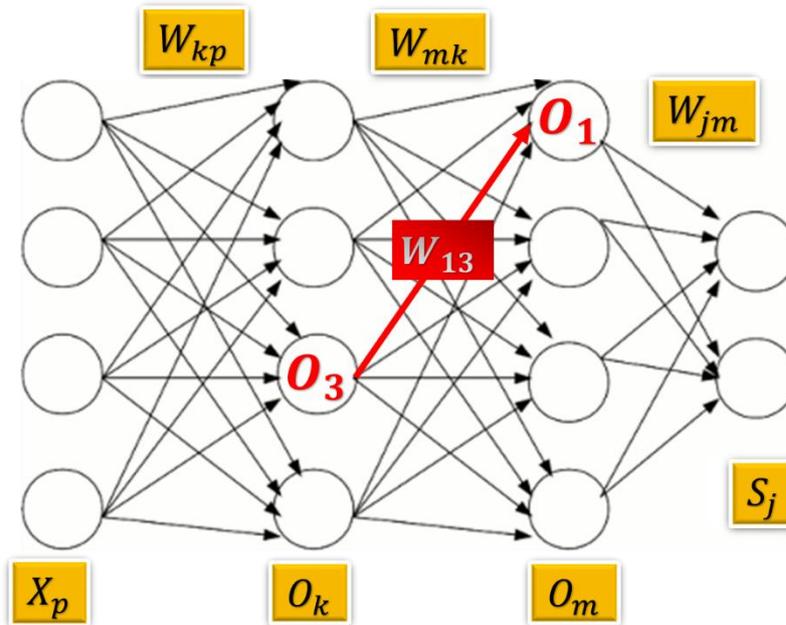
As camadas, da esquerda para direita, recebem os nomes de camada de entrada, oculta e de saída, respectivamente. É possível existir mais de uma camada oculta e nesse caso chama-se “rede multicamadas”. Cada círculo representado é chamado de neurônio e eles são entrelaçados como mostrado na figura, isto é, cada neurônio da camada anterior envia informação para todos os subsequentes.

Na camada de entrada, estarão os nossos *inputs*, ou seja, os dados que temos a fornecer ao algoritmo. Ao entrar com esses dados, eles passarão para a camada oculta, em que cada neurônio de entrada possui um peso de conexão para cada neurônio que o receber. E o valor de cada neurônio da camada oculta será igual a soma das multiplicações de cada dado de entrada vezes seu respectivo peso associado, além do acréscimo de uma constante chamada *bias* – o valor final pode ser ajustado e normalizado. O trabalho da máquina é encontrar o melhor conjunto de valores de pesos e bias para atingir melhores previsões em dados não vistos.

Considere uma rede neural com duas camadas ocultas e duas saídas, exemplificada na Figura 14. Cada neurônio X_p da camada de entrada se conecta

com cada neurônio O_k da primeira camada oculta através dos pesos W_{pk} . A mesma ideia para todas as camadas, de acordo com as setas da figura 14. Por exemplo, o peso da seta que conecta o neurônio 3 da primeira camada oculta com o neurônio 1 da segunda camada oculta é $W_{mk} = W_{13}$.

Figura 14 – Rede neural com duas camadas ocultas.



Fonte: Autor (2023).

O somatório das multiplicações de cada neurônio da camada anterior por seus respectivos pesos, com o acréscimo do *bias* b_j , compõem a camada seguinte. Para exemplificar essa descrição, observe a fórmula para os neurônios da camada de saída.

$$S_j = A \left(\sum_m W_{jm} O_m + b_j \right), \quad (j = 1,2) \quad (13)$$

Aqui j é o número de saídas, que nesse exemplo é 2, e W_{jm} é o peso que conecta o neurônio de saída j com o neurônio m da camada anterior. O parâmetro A é a função de ativação que em geral se usa para realizar ajustes cruciais nos dados de

saída como normalizações, parametrizações, desativação ou ativação de neurônios, etc.

Podemos ainda escrever a equação (13) na forma matricial:

$$\begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = A \left(\begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \end{bmatrix} \cdot \begin{bmatrix} O_1 \\ O_2 \\ O_3 \\ O_4 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \right). \quad (14)$$

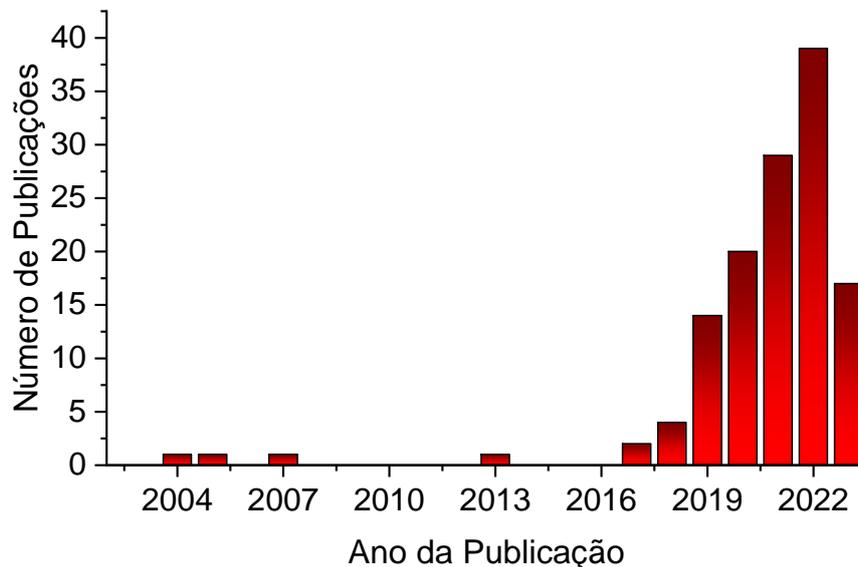
Isso pode ser estendido para qualquer dimensão (quantidade de neurônios nas camadas). Esse é o procedimento padrão para uma rede neural. Os pesos e *bias* iniciam com valores aleatórios (que podem ou não estar normalizados), mas o algoritmo possui etapas de verificação das perdas e custos para ir ajustando esses parâmetros a cada iteração conforme uma métrica de otimização definida no algoritmo. Nessa etapa deve ser estabelecido algum critério de parada pelo usuário, até que os dados estejam se ajustando bem, levando em conta os cuidados com o *overfitting* e *underfitting*. Para mais detalhes, uma referência muito completa sobre ML e *deep learning* é (GÉRON, 2022).

4 APRENDIZADO DE MÁQUINA PARA TERMOMETRIA FOTOLUMINESCENTE

Nos capítulos anteriores foi dissertado acerca das duas vertentes desse trabalho: 1) termometria óptica, 2) aprendizado de máquina. Fizemos uma abordagem histórica de trabalhos pioneiros em ambos os campos, destacando datas importantes, aplicações e diversos trabalhos notáveis.

Nesse capítulo vamos unir esses dois campos e mostrar como podemos implementar algoritmos de aprendizado de máquina dentro da termometria óptica. Vale notar que, embora o interesse seja crescente em utilizar ML em termometria fotoluminescente, ainda há poucos trabalhos publicados na literatura – compare a Figura 15 com a Figura 9 – envolvendo tópicos relacionados a termometria com tópicos relacionados a ML e suas variações.

Figura 15 – Busca na *Web Of Science* para TO com ML com o tópico: ((*nanothermometry OR thermometry OR thermometer OR nanothermometer OR phosphor temperature-measurements OR photoluminescence thermometry*) AND (*machine learning OR deep learning OR artificial intelligence*)).



Fonte: Autor (2023).

Não é algo novo implementar algoritmos de ML em diversos campos da ciência. Como descrevemos no capítulo 4, as aplicações de ML vêm desde os anos de 1950 e têm se espalhado por todas as áreas ao longo do tempo.

4.1 TRABALHOS ANTERIORES

Em 2004, Adamczak e colaboradores desenvolveram um método para realizar previsão das acessibilidades relativas do solvente dos resíduos de aminoácidos nas proteínas usando regressão não-linear baseada em redes neurais (ADAMCZAK; POROLLO; MELLER, 2004). Transformando as entradas em vetores, eles consideraram uma rede neural multicamada com uma codificação de um termômetro na camada de saída.

Em 2013, um grupo norte-americano usou um algoritmo de aprendizado de máquina não-supervisionado chamado *K-means* para separar e classificar picos espectrais de água e gordura (LIN et al., 2013). Nesse caso, o algoritmo se torna um programa de classificação, pois o objetivo é receber o espectro e apontar quais são os picos referentes a água e quais são os picos referentes a gordura.

Em 2014, um grupo belga relatou estudo de redes neurais para termometria de luminescência usando rodamina B (LIU et al., 2014). De acordo com os autores, podem ser considerados como parâmetros de entrada para a rede neural o espectro cru ou combinações de características espectrais.

Em 2018 (KIM et al., 2018) usaram um algoritmo de rede neural que aprende com a evolução do calor para reconstruir imagens térmicas. Esse estudo mostra a viabilidade de um método de monitoramento térmico usando um elemento de ultrassom externo e a reconstrução de imagens via *deep learning*.

Também no mesmo ano de 2018, um grupo alemão publicou pela primeira vez o uso do ML para prever a temperatura de morte de ratos e camundongos doentes, comparando com o monitoramento da temperatura interna e externa do animal (MEI et al., 2018). O grupo fez o monitoramento da temperatura ao longo do tempo e usou os dados dos animais (sobreviventes e mortos) como treino, teste e validação foi possível prever a temperatura e a quantidade de horas que define o limiar da morte deles.

Em 2019 foi utilizado algoritmo de *deep learning* para recuperação das distribuições de temperaturas e concentração em uma combustão de gás (REN et al., 2019). Os autores desenvolveram um algoritmo de rede neural de multicamadas ocultas e o ajuste adequado dos parâmetros permitiu que eles fizessem a previsão e

recuperação das temperaturas e concentrações a partir da emissão espectral infravermelha da combustão.

Ainda no mesmo ano, o aprendizado de máquina foi usado para prever a temperatura de soluções de nanopartículas plasmônicas (MALLAWAARACHCHI et al., 2019). Os autores usaram 3 tipos de algoritmos de árvore de decisão para um treinamento supervisionado. Foi mostrado que embora sejam algoritmos diferentes, ambos possuem uma previsão que concorda com o esperado experimentalmente.

Em 2020 (LEWIS et al., 2020) usaram o tempo de decaimento da luminescência de pontos quânticos como dados de entrada para uma rede neural de 7 camadas. O objetivo é usar essa entrada para obter como resposta a temperatura na camada de saída, constituindo assim um problema de regressão do ML aplicado na termometria óptica possibilitando a realização de um algoritmo de previsão da temperatura.

Em 2020 também outro trabalho foi publicado estudando os efeitos da temperatura em nanopartículas luminescentes contendo európio (ŠEVIĆ et al., 2020). O algoritmo empregado é um dos mais antigos e consolidados, chamado *Principal Component Analysis* que consiste em reduzir a dimensão dos dados para trabalhar com os parâmetros mais importantes. Com isso eles mostraram que a máquina pode ser treinada para diferenciar dados espectrais de diferentes temperaturas.

No ano seguinte, um grupo norte americano desenvolveu um experimento de proposta de termômetro óptico em fibra de cerâmica para analisar altas temperaturas (QIAN et al., 2021). Eles investigaram o regime de altas temperaturas (1000°C-1650°C), o que torna interessante para aplicações industriais. Nesse trabalho, eles colocaram emissões obtidas através do aquecimento como dados de entrada em uma rede neural convolucional, cujo objetivo é fornecer a temperatura na camada de saída. Este trabalho se tornou pioneiro no uso de ML para predição de altas temperaturas.

Mais recentemente, em 2022, um grupo chinês reportou um estudo comparativo da termometria óptica para 3 métodos: Razão de Intensidades, Regressão linear múltipla e Redes Neurais convolucional (CUI et al., 2023). Descrevemos estes métodos nas seções 2.2.2, 2.3 e 4.5, respectivamente, nesta dissertação. No trabalho citado, eles mostraram que usar *deep learning* se tornou uma ferramenta superior aos métodos já outrora amplamente usados (razão de intensidades e

regressão linear múltipla). Com isso, eles calcularam erros relativos e desvio padrão entre a temperatura prevista pelos métodos e aquela mensurada experimentalmente, desse modo mostraram que o *deep learning* errou menos vezes do que os outros métodos.

Reconhecendo a capacidade de análise e predição de dados por meio de algoritmos de aprendizado de máquina e aprendizado profundo, realizamos um estudo usando rede neural multicamada para estimar a temperatura a partir do espectro fotoluminescente de microesferas de Nd:YAG (ver Figura 19). Discutiremos agora as novidades e diferenças do nosso trabalho, bem como sua contribuição para a literatura.

4.1.1 Nosso trabalho (2023)

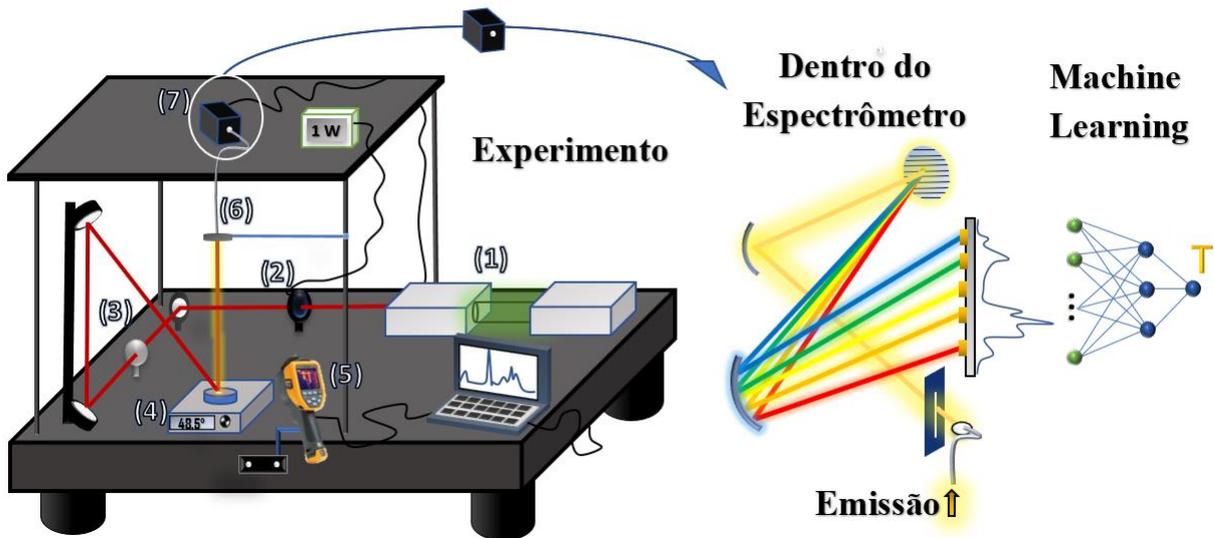
Desenvolvemos neste trabalho um algoritmo de ML, cujo código fonte pode ser acessado no apêndice A, para realizar previsões de temperatura, usando como parâmetros de entrada espectros fotoluminescentes de partículas cristalinas dopadas com Nd³⁺ (seção 5.3). De modo similar ao trabalho de (CUI et al., 2023), comparamos os métodos e mostramos que o uso do ML aumenta a precisão dos resultados. A novidade do nosso trabalho é que, em primeiro lugar, usamos como íon sensibilizador o neodímio, que já é amplamente conhecido por suas excelentes propriedades espectroscópicas; em segundo lugar, no algoritmo desenvolvido, usamos a técnica de Aprendizado de Máquina automatizado (*Automated Machine Learning*), em que essencialmente a máquina analisa os dados e nos fornece o melhor algoritmo para trabalharmos; em terceiro lugar, mostramos que mesmo para um sistema com baixa sensibilidade térmica e com ruídos no sistema de aquisição de dados, o método de aprendizado de máquina é superior aos métodos convencionais para realizar previsões da temperatura, dado um espectro como parâmetro de entrada.

Este trabalho é um caminho para medidas em tempo real; embora a relação sinal-ruído seja muito boa, existem flutuações intrínsecas de uma realização experimental para outra. Além disso, pautamos uma discussão estatística sobre a análise dos resultados experimentais.

4.2 O EXPERIMENTO E A INTERPRETAÇÃO COMPUTACIONAL

Na figura 16, da esquerda para direita, estão representados o arranjo experimental (esquerda), o funcionamento dentro do espectrômetro (centro) e o tratamento das intensidades através do ML no computador (direita).

Figura 16 – Esquemática do experimento, espectrômetro e algoritmo usado.



Fonte: Autor (2023).

- Arranjo Experimental

Foi utilizado um laser de Ti:Safira ($\text{Ti:Al}_2\text{O}_3$) sintonizável (720 nm – 850 nm) em regime contínuo (1) com potências ajustadas de acordo com o experimento feito. Nesse momento, a potência do laser pode ser medida por um fotodetector (2) ligado a um medidor de potência. Retirando o fotodetector, o laser é refletido em um espelho que o desvia para um polarizador e um conjunto de espelhos que elevam a luz na direção vertical (3), deste modo o laser incide na amostra, que está sobre um aquecedor (4), formando um ângulo de 45°. O aquecedor é controlado digitalmente para esquentar a amostra, cuja temperatura é registrada por uma câmera térmica (5) com sistema de detecção no intervalo de -20 °C até 650 °C com precisão de 1 °C. As emissões da amostra são coletadas na direção perpendicular através de uma fibra óptica (6) que encerra no espectrômetro (7). O espectrômetro usado tem um intervalo espectral de 350 nm – 1000 nm com resolução de 1 nm. O espectrômetro

envia as informações para um *software* adequado e assim prosseguimos o tratamento dos dados no computador. Em cada realização experimental foi coletado uma média de 12 mil espectros, simultaneamente com a evolução temporal da temperatura da amostra, para posterior tratamento computacional.

- O espectrômetro (parte central da figura 16)

As emissões que chegam na fibra óptica, são primeiramente direcionadas ao conector de entrada do espectrômetro. Uma vez passado pelo conector de entrada, a luz atravessa uma fenda que age como uma abertura de entrada garantindo um alinhamento da luz. Essa radiação incide num espelho que a reflete para uma grade de difração, que separa os diferentes comprimentos de onda. Os feixes de diferentes comprimentos de onda são enviados a outro espelho que os reflete para a câmera detectora que identifica cada comprimento de onda em seus vários píxeis, definindo assim sua resolução. Para mais detalhes sobre o funcionamento desse espectrômetro, ver (*"How Does a Spectrometer Work?"*, 2017).

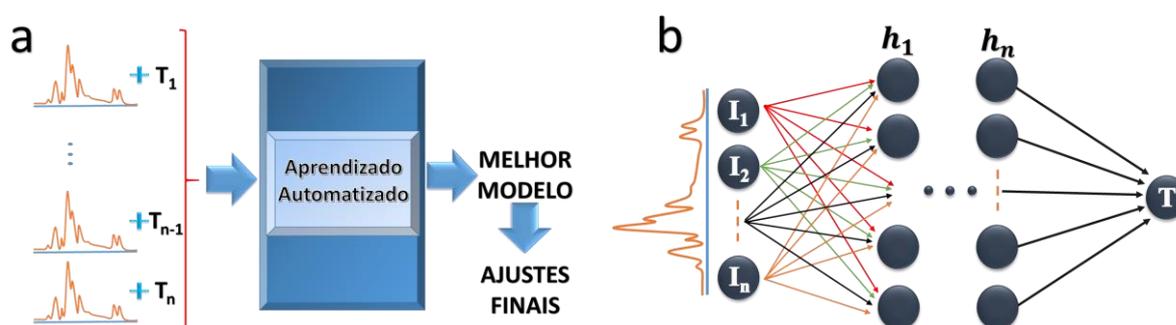
- *Machine Learning* (aprendizado de máquina)

Cada píxel do sensor do espectrômetro faz a leitura de uma faixa de comprimentos de onda de acordo com sua resolução espectral, de modo que todos juntos fornecem o espectro completo. No nosso trabalho, levamos ao ML as intensidades recebidas (espectro) por cada píxel e tratamos isso como nossos dados de entrada (*input*). Desse modo, ao trabalharmos com uma rede neural, por exemplo, cada neurônio de entrada irá receber a intensidade referente a cada píxel do sensor (lado direito da Figura 16 e da Figura 17). Similarmente, no algoritmo de árvores de decisão, cada nó raiz de árvores diferentes serão os píxeis da câmera (Figura 18). O conjunto de pesos e *bias*, para o caso da rede neural, ou o conjunto de decisões, para o caso da árvore de decisão, estão relacionadas as intensidades e o modelo irá buscar o melhor conjunto de parâmetros que se adeque aos dados.

Uma das novidades desse trabalho foi a utilização do aprendizado automatizado (Figura 17a), isto é, fornecemos os dados de entrada para uma ferramenta de otimização e busca do melhor modelo para os dados. O otimizador que usamos se chama *Tree-Based Pipeline Optimization Tool* (TPOT). Embora o TPOT recomende

o melhor algoritmo, podemos ainda aperfeiçoar os parâmetros, que é o processo que chamados de ajustes finais, como refere a figura 17a. Dentre os modelos buscados pelo TPOT, a sugestão para nossos dados foi “*ExtraTreesRegressor*”, que é algoritmo baseado em árvore de decisão, e também o “*MultilayerPerceptron*”, que é essencialmente a construção de rede neural de mais de uma camada oculta.

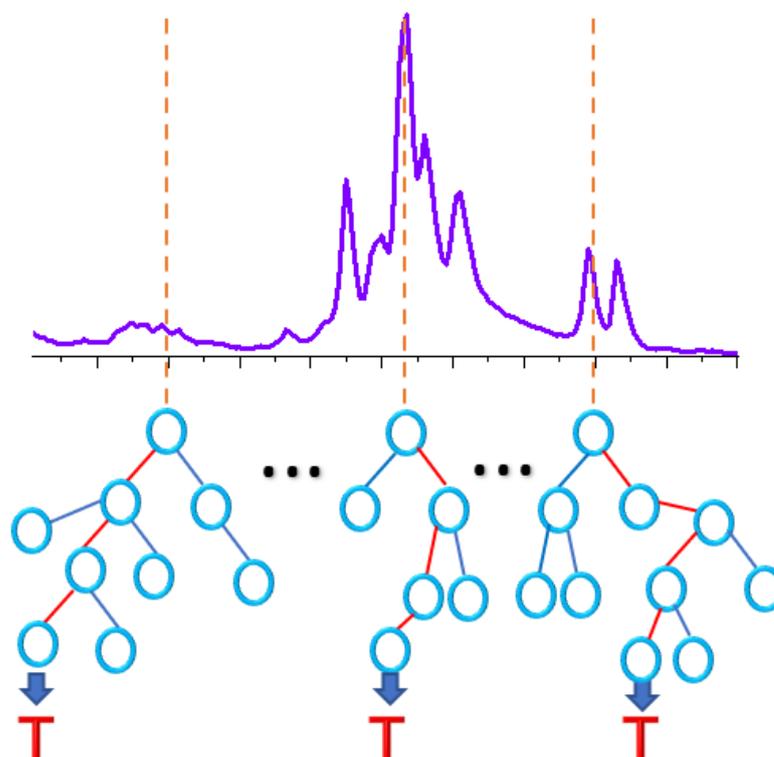
Figura 17 – Aprendizado automatizado e rede neural de múltiplas camadas.



Fonte: Autor (2023).

Para a rede neural multicamadas, podemos ajustar a quantidade de camadas ocultas que desejarmos em busca da melhor configuração. É válido destacar que quanto mais camadas tivermos, mais parâmetros o programa irá otimizar e assim se pode encontrar resultados mais precisos, pelo preço de um maior custo computacional, embora isso não seja uma regra. Neste trabalho o objetivo é usar o ML fornecendo como dados de treino espectros rotulados com suas respectivas temperaturas de partículas cristalinas com Nd^{3+} , tal que a máquina seja capaz de calcular a temperatura a partir de novos espectros fotoluminescentes não rotulados (etapa de teste). Por esta razão, a temperatura é o neurônio de saída na figura acima. Eis que estamos diante de um problema de regressão (e não de classificação) com o objetivo de encontrar um valor específico no final. Aqui estamos tratando do aprendizado supervisionado, como descrito anteriormente, pois os dados de treino que fornecemos a máquina são os espectros (cada píxel referencia um neurônio de entrada) e as temperaturas correspondentes das partículas com Nd^{3+} . Deixamos a máquina treinar com os dados rotulados, para depois estimarmos as previsões com dados futuros.

Figura 18 – Árvore de decisão para termometria óptica.



Fonte: Autor (2023).

No caso da árvore de decisão, pode-se usar o algoritmo chamado *ExtraTreesRegressor* (“*sklearn.ensemble.ExtraTreesRegressor*”, [s.d.]). Como falamos, esse é um problema de regressão, por isso o nome do algoritmo. Além disso, esse algoritmo gera várias árvores de decisão de forma aleatória. Diferente da interpretação que descrevemos para o caso da rede neural, em que os neurônios de entrada caracterizam os píxeis da câmera do espectrômetro, no algoritmo *ExtraTreesRegressor*, cada píxel da câmera é um nó raiz de cada árvore gerada, ou seja, cada intensidade do espectro é responsável por gerar cada árvore de decisão dentro desse algoritmo. De todo modo, as decisões tomadas pela máquina levam a um resultado final: temperatura. Esse é nosso objetivo! Queremos fazer um treinamento na máquina através dos espectros rotulados para que possamos prever a temperatura referente a novos espectros fotoluminescentes.

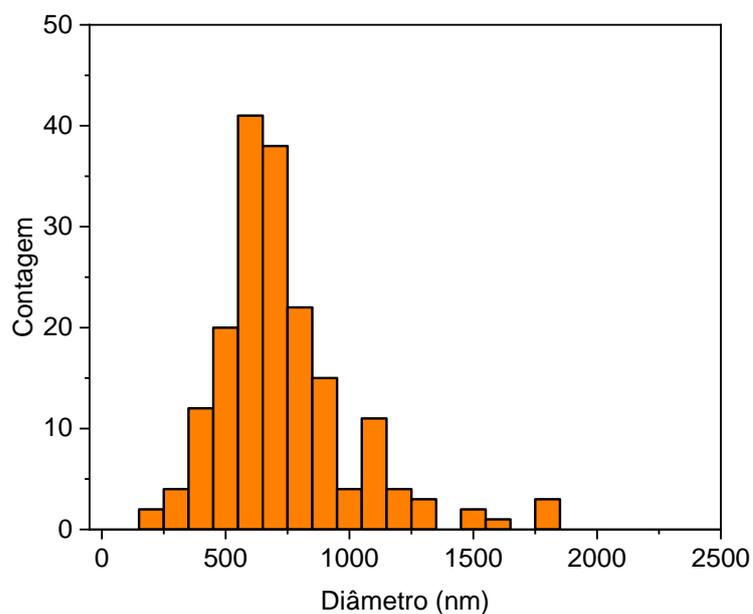
O interessante dos algoritmos de aprendizado de máquina é que a quantidade de parâmetros e os cálculos de ganho de informação empregados nos modelos

acima descritos, geram uma capacidade muito grande do computador fazer boas previsões e mostraremos isso ao comparar nossos resultados com os métodos convencionais.

4.3 A AMOSTRA

A amostra usada consiste em um pó de esferas submicrométricas de Nd³⁺:YAG, cuja síntese e caracterização está disponível em (PUGINA et al., 2021). Em resumo, essas partículas foram preparadas através de pirólise de aerossol em que foi usado suspensão de boemita dopada com 60% de Y³⁺ e co-dopada com 3.5% de Nd³⁺ como precursor. A distribuição de tamanhos das partículas mostra um diâmetro médio de centenas de nanômetros (figura 19).

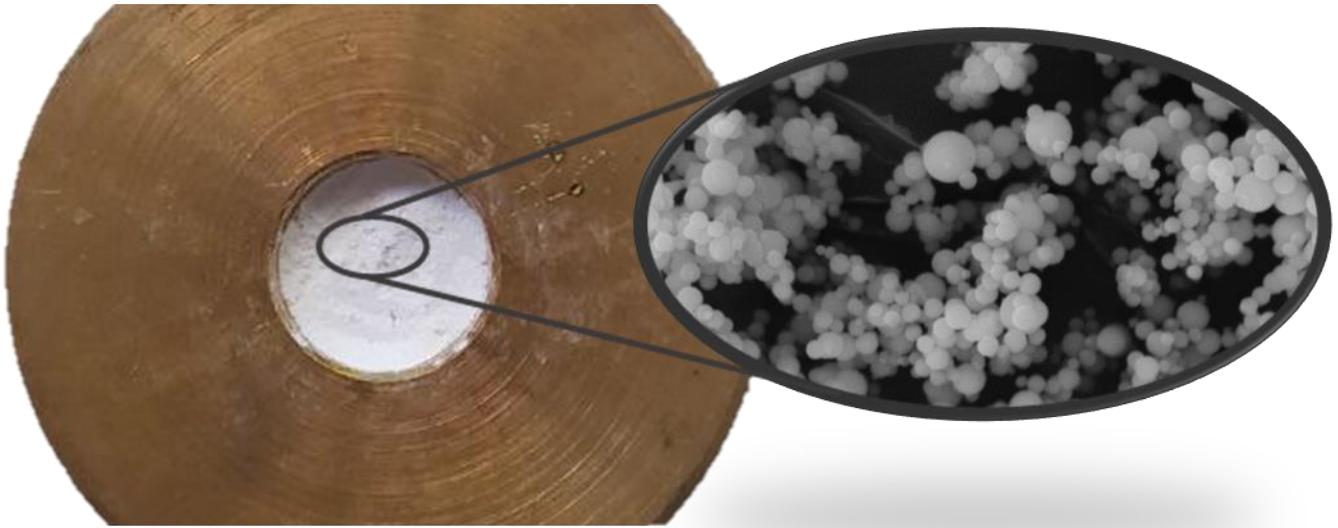
Figura 19 – Distribuição de tamanho das partículas de Nd³⁺:YAG (3.5%).



Fonte: Autor (2023).

A amostra (Figura 20) indicada no arranjo experimental (Figura 16) é colocada num suporte metálico para ser aquecido e excitada simultaneamente.

Figura 20 – Amostra usada: Nd³⁺:YAG (3.5%).

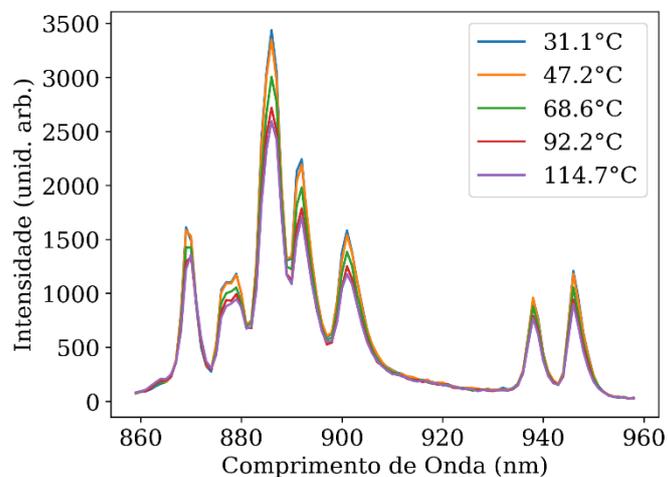


Fonte: Autor (2023). Parte da figura usada de (PUGINA et al., 2021), com permissão.

4.4 RESULTADOS E DISCUSSÕES

Sob excitação de radiação com comprimento de onda de 760 nm, verifica-se diferentes emissões (Figura 21) no regime de conversão descendente de energia, uma vez que as emissões possuem comprimentos de onda maior (portando, energia menor) que o feixe de excitação.

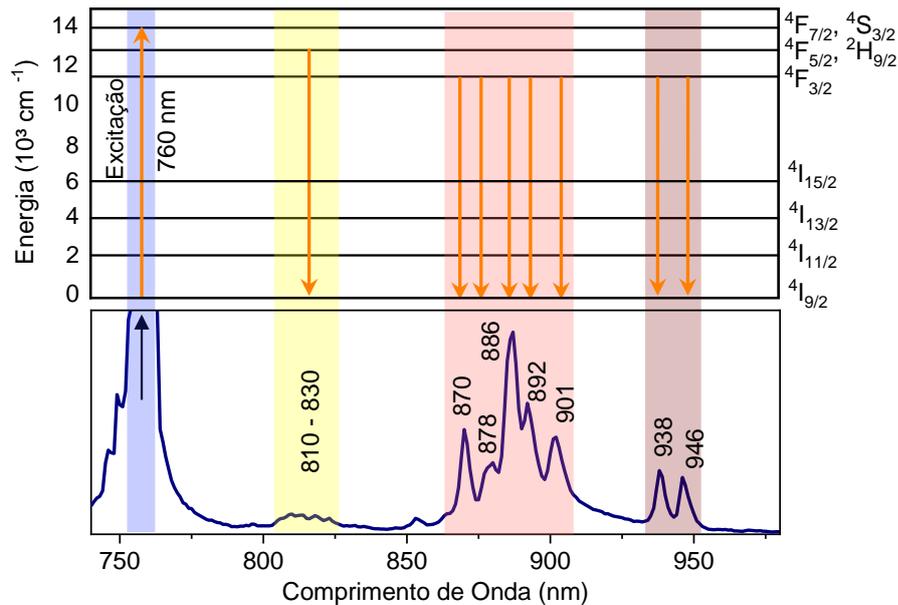
Figura 21 – Espectros de emissão das partículas das partículas de Nd³⁺:YAG (3.5%).



Fonte: Autor (2023).

Como podemos ver na Figura 22, a excitação em 760 nm ($4I_{9/2} \rightarrow 4S_{3/2}, 4F_{7/2}$) proporciona as transições de emissão em torno de 810 nm ($4F_{5/2}, 2H_{9/2} \rightarrow 4I_{9/2}$), 870 nm ($4F_{3/2} \rightarrow 4I_{9/2}$), 878 nm ($4F_{3/2} \rightarrow 4I_{9/2}$), 886 nm ($4F_{3/2} \rightarrow 4I_{9/2}$), 892 nm ($4F_{3/2} \rightarrow 4I_{9/2}$), 901 nm ($4F_{3/2} \rightarrow 4I_{9/2}$), 938 nm ($4F_{5/2}, 2H_{9/2} \rightarrow 4I_{11/2}$) e 946 nm ($4F_{5/2}, 2H_{9/2} \rightarrow 4I_{11/2}$).

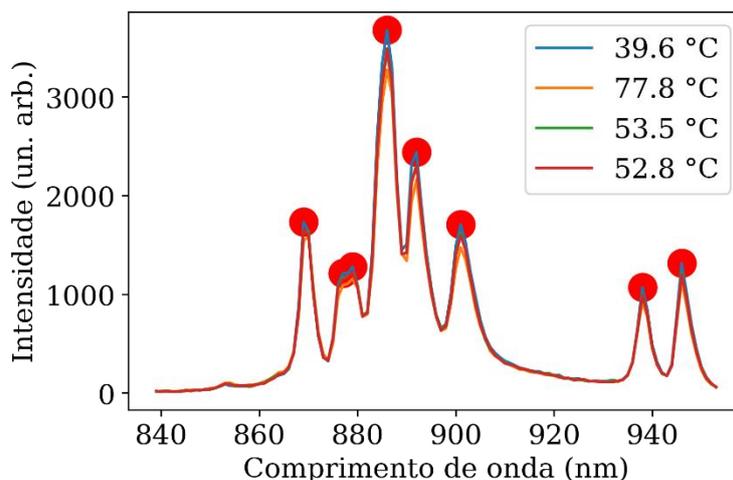
Figura 22 – Níveis de Energia e emissões para a amostra Nd^{3+} : YAG (3.5%).



Fonte: Autor (2023).

O espectro fotoluminescente possui várias emissões, mas destacamos 8 picos (Figura 23), com os quais podemos trabalhar os métodos convencionais: Razão de Intensidades (*luminescence intensity ratio*, LIR) e Regressão linear múltipla (*multiple linear regression*, MLR).

Figura 23 – Picos usados para o LIR e MLR.



Fonte: Autor (2023).

Dada a identificação de 8 picos nesse espectro, fazendo uma combinação de dois a dois, podemos encontrar a quantidade total de possibilidades para trabalhar com a razão de intensidades. A combinação é dada por:

$$C_k^n = \frac{n!}{k!(n-k)!}. \quad (15)$$

Faça $n = 8$ (número de picos) e $k = 2$ (tamanho do grupo, pois as combinações para a LIR são feitas em pares). De (15) temos:

$$C_2^8 = \frac{8!}{2!(6)!} = \frac{8 \times 7}{2} = 28. \quad (16)$$

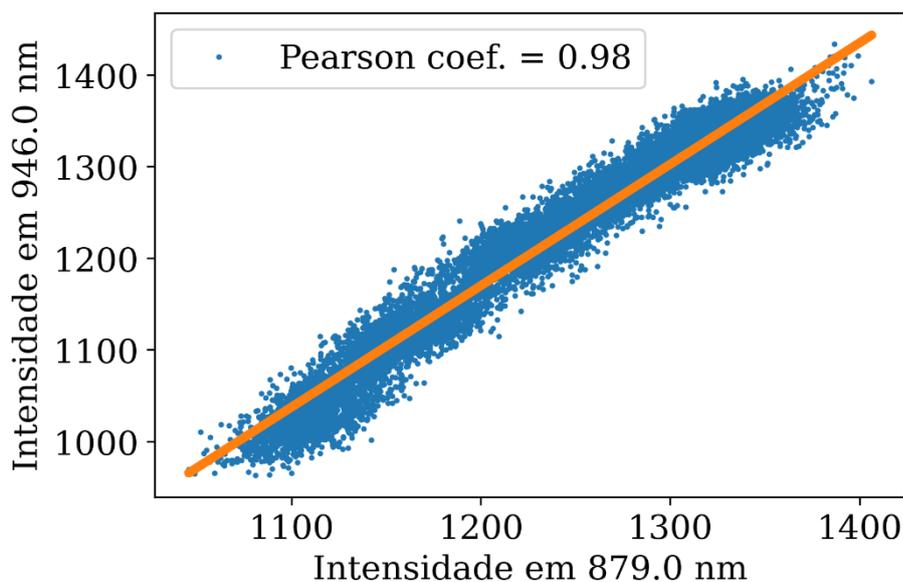
Todas as 28 possibilidades podem ser verificadas no apêndice B.

Em relação ao espectro (figura 23), observa-se que a variação das intensidades não é tão grande no intervalo de temperatura indicado. Essa observação já era esperada, uma vez que já é reportado na literatura que, para o neodímio, a sensibilidade térmica geralmente é baixa (DA SILVA et al., 2020; RAKOV; MACIEL, 2021). É importante destacar que, se desenvolvermos um método que torne precisa a termometria para um sistema de baixa sensibilidade térmica,

espera-se que o método será bem-sucedido para sistemas de alta sensibilidade térmica.

Tomando as intensidades em 946 nm e em 879 nm e representando uma em função da outra (figura 24), nota-se uma flutuação em torno de uma reta (reta laranja ajustada). As flutuações intrínsecas do sistema de detecção de dados acarretam nas flutuações em torno da reta. Essas flutuações ocorrem de um espectro para outro. O coeficiente de Pearson no valor de 0.98 indica um forte comportamento correlacionado entre as emissões, embora ainda ocorram flutuações em torno de uma média. Essa forte correlação, embora com uma dispersão notável entre os pontos, vem do fato que a população de ambos os níveis está em equilíbrio térmico e são diretamente proporcionais, para cada temperatura.

Figura 24 – Relação entre 2 intensidades.

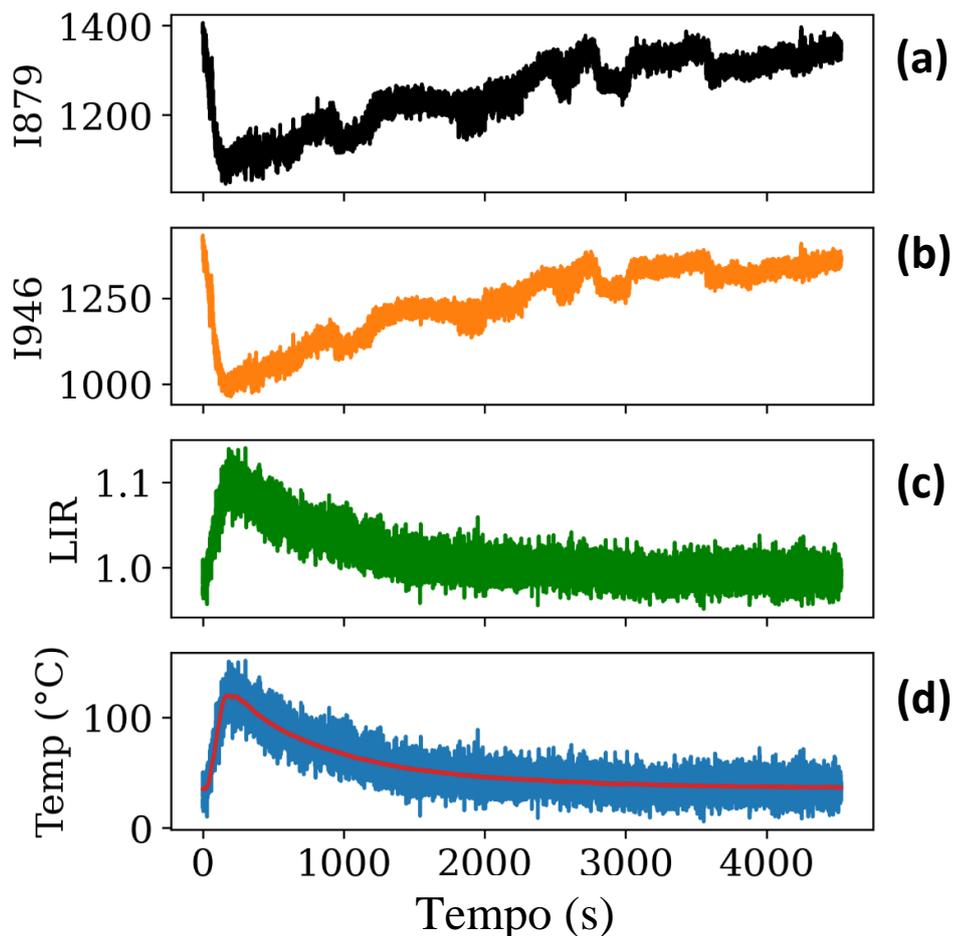


Fonte: Autor (2023).

É importante destacar a boa relação sinal-ruído nos espectros da Figura 23. À primeira vista, o ruído não é um problema. Mas, ao realizar medidas consecutivas observamos flutuações intrínsecas ao sistema de aquisição de dados. Mais especificamente, ao espectrofotômetro. Verificamos esse comportamento em duas marcas distintas: *Stellarnet* e *OceanOptics* (os dados crus podem ser solicitados ao autor). A existência desse ruído pode levar a erros na determinação da temperatura.

Outro problema que pode surgir são variações de intensidade do laser de excitação. Essas, à princípio, poderiam ser eliminadas fazendo a razão de intensidades de emissões de dois níveis de energia excitados por um fóton. Desse modo, obtemos a curva da razão de intensidades (LIR) em função do tempo. De posse da temperatura monitorada experimentalmente pela câmera térmica e com a curva da razão de intensidade (ver apêndice B) em função dessas temperaturas, podemos então propor um termômetro raciométrico para qualquer temperatura a partir da curva ajustada.

Figura 25 – (a) Intensidades das emissões de 879 nm; (b) Intensidades das emissões de 946 nm; (c) LIR; (d) Temperatura determinada pela LIR (azul) e medida pela câmera térmica (vermelha). Todas as curvas em função do tempo.



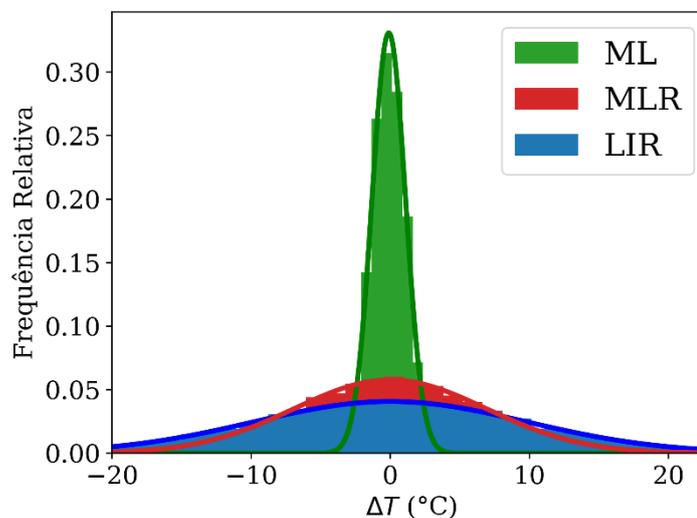
Fonte: Autor (2023).

Ver na Figura 25, intensidades em 879 nm (a), intensidades em 946 nm (b), razão entre a banda integrada referente a essas intensidades (c) e em (d)

temperatura medida pela câmera térmica (vermelho) durante um ciclo experimental (aquecimento e resfriamento das partículas) e previsão feita a partir do ajuste usando o método de LIR (azul).

Observando esses resultados, está claro que a temperatura calculada pela LIR apresenta uma incerteza muito grande (quando comparada a temperatura de referência), o que dificulta a proposta de termômetro óptico para sistemas como esse. Então, como já mencionado anteriormente, podemos partir para análises agora usando o método da regressão linear múltipla (seção 2.3), em que essencialmente considera vários parâmetros além do único LIR para o cálculo da temperatura. No nosso trabalho, combinamos várias razões de intensidades dentre as possíveis que se pode obter usando os 8 picos do espectro (Figura 23). No cálculo da regressão linear múltipla, foram selecionadas todas as combinações de razão de intensidades cujo coeficiente de Pearson deu igual ou maior que 0.85 (foram 6, ver apêndice B). Para o tratamento usando o aprendizado de máquina, usamos todo o espectro como parâmetros de entrada (Figura 17), ao invés de picos/áreas de intensidade como feito nos métodos LIR e MLR. Em todos os casos comparamos a temperatura obtida via um dado método com aquele que foi medida pela câmera térmica (temperatura alvo). Assim, estabelecemos o histograma para cada uma das medidas como função de $\Delta T = T_i - T'_i$, em que T_i é a temperatura experimental medida pela câmera térmica enquanto T'_i é a temperatura prevista pelo método.

Figura 26 – Histograma.



Fonte: Autor (2023).

As medidas de ΔT revelam um comportamento tipo distribuição normal, cuja equação é descrita por:

$$f(\Delta T) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{\Delta T - \mu}{\sigma}\right)^2}, \quad (17)$$

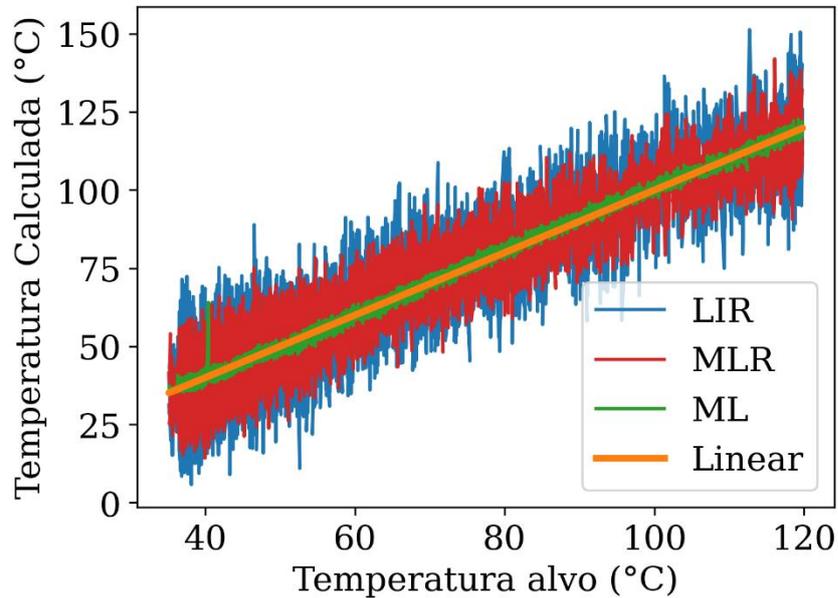
em que σ é o desvio padrão da distribuição e μ localiza o centro da distribuição.

Uma vez que o modelo prevê adequadamente a temperatura, ele tende a estar muito perto da temperatura alvo (figura 27) com um erro dado pelo desvio padrão do histograma. Desse modo, quanto melhor estiver o modelo, ao subtrair a temperatura alvo da temperatura prevista, restará basicamente apenas a incerteza natural do experimento, que são simplesmente flutuações (valores aleatórios) somadas de acordo com a resolução temporal do termômetro usado. A soma desses valores equivale a uma soma de várias incertezas (que são aleatórias). Portanto, para um conjunto muito grande de medidas temos um conjunto grande contendo somas de valores aleatórios, isso leva a uma distribuição gaussiana, de acordo com o teorema central do limite.

O caso ideal seria que ΔT fosse sempre zero, ou seja, que a temperatura prevista (calculada) sempre acertasse a temperatura alvo (experimental). Nesse caso, o histograma seria uma linha centrada em zero. Nesse sentido, quanto mais estreita for a distribuição, melhor é o termômetro óptico. Lançamos então, neste trabalho, uma questão de interesse para trabalhos futuros, que é realizar as medidas da forma da figura 26 para amostras com maiores sensibilidades térmicas e ver o quão estreitas ficariam as distribuições.

Ao colocar a temperatura calculada versus a temperatura alvo, o caso ideal seria uma reta de coeficiente angular 1 e coeficiente linear 0. O termômetro seria o melhor possível com o menor grau de incerteza na temperatura. Vê-se na figura 27 que a abordagem usando o ML é a melhor comparado aos casos dos métodos convencionais.

Figura 27 – Temperatura calculada x temperatura alvo.



Fonte: Autor (2023).

Os ruídos em torno do comportamento linear, atribuímos principalmente ao sistema de detecção experimental além da baixa sensibilidade da amostra.

Para quantificar a qualidade do termômetro e das previsões, calculamos as medidas de performance Erro Médio Absoluto (*Mean Absolute Error*, MAE), a raiz do erro médio quadrático (*Root Mean Square Error*, RMSE), o coeficiente de determinação R^2 e o valor do desvio padrão (sigma) relacionado ao histograma de ΔT . Ver quadro 1.

Quadro 1 – Métricas de performance.

	MÉTODOS		
	LIR	MLR	ML
MAE (°C)	7.803	5.488	0.982
RMSE (°C)	9.799	6.917	1.222
R^2	0.794	0.896	0.998
SIGMA (°C)	9.799	6.917	1.222

Fonte: Autor (2023).

Cada parâmetro tem um significado físico importante para caracterizar e quantificar a adequação do modelo proposto aos dados utilizados.

- Erro Médio Absoluto (*Mean Absolute Error*, MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|. \quad (18)$$

Em que y_i é o valor alvo (experimental) e \hat{y}_i é o valor previsto pelo método. Nota-se que aqui é uma comparação direta e equivalente ao eixo x da figura 26, porém com o somatório em módulo dividido pelo número total de medidas.

- Raiz do Erro Médio Quadrático (*Root Mean Squared Error*, RMSE)

A medição do erro médio quadrático é definida por (“3.3. Metrics and scoring”, [s.d.]):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}. \quad (19)$$

Em que y_i é o valor alvo (experimental) e \hat{y}_i é o valor previsto pelo algoritmo. É notório ainda que o RMSE (para n grande) é exatamente a definição de desvio padrão, que faz coincidir com aquele estimado pela distribuição gaussiana (ver quadro 1).

- Coeficiente de determinação

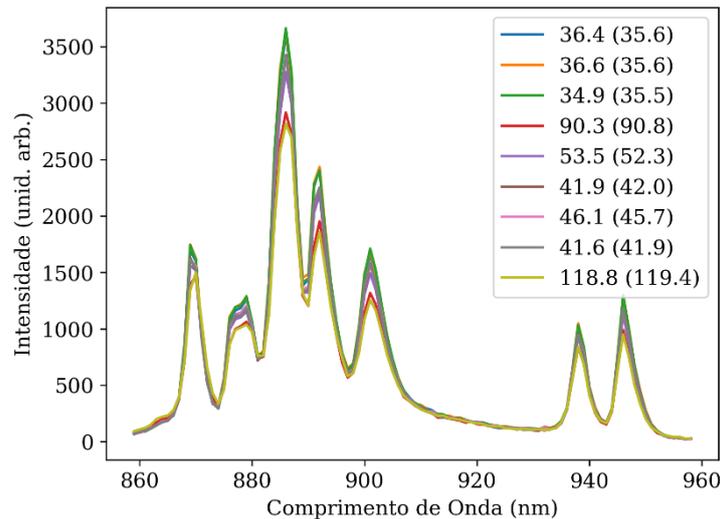
Dentro do código (ver apêndice A), este valor é calculado usando a função “r2_score”, o qual é denotado por R^2 . Este coeficiente mede o quão bem os resultados conseguem prever medidas reais. A definição matemática é (“3.3. Metrics and scoring”, [s.d.]):

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \langle y \rangle)^2} \quad (20)$$

Em que y_i é o valor alvo (experimental), $\langle y \rangle$ é a média dos valores alvo e \hat{y}_i é o valor previsto pelo algoritmo. Notemos que quando o valor previsto sempre acerta o valor alvo, o resultado do coeficiente de determinação será 1, isto é, a previsão é perfeita. No entanto se o valor for 0 corresponde a um modelo constante e, portanto, não há qualquer variabilidade associada ao modelo. Essencialmente quanto mais próximo de 1, melhores as previsões. Nesse contexto, pode-se constatar que sob todas as métricas apresentadas, o ML mostrou melhores resultados que as demais abordagens discutidas.

Na Figura 28 podemos comparar diretamente os valores experimentais (fora dos parênteses) com o que foi previsto pelo *machine learning* (dentro dos parênteses). Nota-se que de fato os valores preditos são muito próximos dos valores experimentais associados.

Figura 28 – Temperatura calculada x temperatura alvo para alguns espectros.

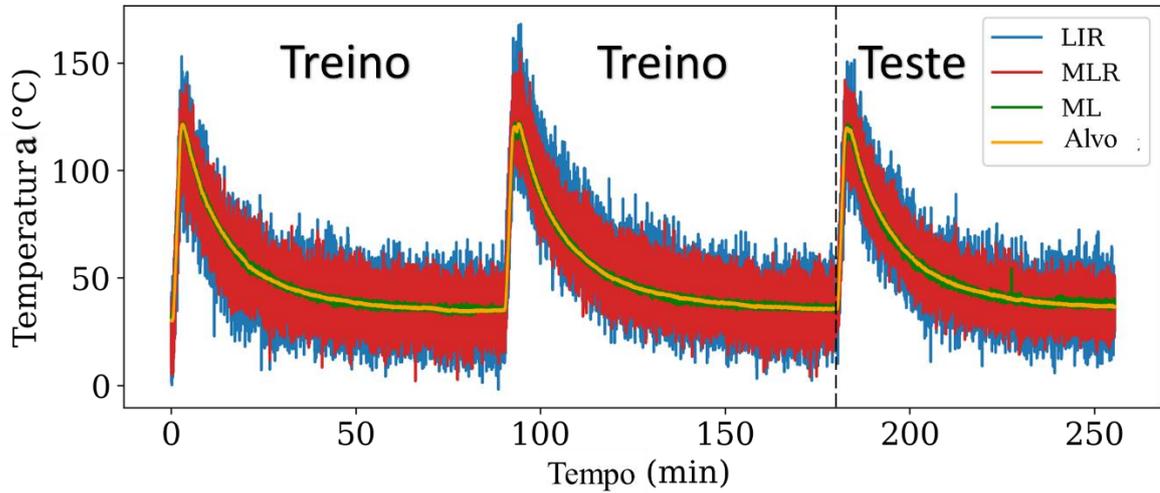


Fonte: Autor (2023).

Para os resultados discutidos até aqui, usamos dois ciclos de aquecimento e resfriamento para treinamento e um ciclo adicional para teste (Figura 29). As temperaturas previstas pelo ML para as figuras anteriores foram através dos dados

de teste, que é o ideal a se fazer. No entanto, por completeza, também realizamos as previsões em cima dos dados de treino, como segue na figura abaixo.

Figura 29 – Treino e Teste entre os ciclos.



Fonte: Autor (2023).

Observa-se que, embora com ruído, a previsão com ML se aproxima bastante dos valores alvo para praticamente todos os instantes de tempo. Assim relatamos que as incertezas associadas a uma previsão de um termômetro está diretamente associada ao método usado para calibrar o equipamento.

5 PERSPECTIVAS

De acordo com nossa discussão do capítulo 5, entendemos que trazer a abordagem do aprendizado de máquina para o campo da termometria óptica é uma exploração emergente, com poucos trabalhos no assunto ainda (figura 15). Após os resultados aqui discutidos, levamos as ideias para novos patamares visando algumas perspectivas, que listadas estão abaixo.

- Considerar outros parâmetros (como, por exemplo, o espectro de excitação) de entrada além dos espectros, para facilitar o treinamento e previsão da máquina.
- Realizar o estudo para diferentes métodos, arquiteturas e variações de redes neurais.
- Estudar a transferência de aprendizado, isto é, treinar com um experimento e testar com outro visando aplicações em sistemas biológicos, mais especificamente, desenvolver um método robusto que seja capaz de lidar com distorções espectrais causadas pela absorção/espalhamento de luz dos tecidos.
- Desenvolver uma plataforma para uso prático em que o usuário apenas fornece o espectro e obtém a temperatura como resposta.
- Estender o estudo para outras amostras e técnicas em conjunto.
- Colocar limitações da física (física para aprendizado de máquina) no processo de treinamento.
- Realizar estudo da resolução temporal e espacial de termômetros ópticos usando diferentes algoritmos de ML.
- Realizar treino por intervalo, ao invés de usar o ciclo completo.
- Adaptar no sistema, um conjunto contendo arduíno e webcam pra monitorar a evolução da temperatura e espectro em tempo real.

A partir deste presente trabalho, deixamos uma contribuição para o futuro da termometria óptica e o uso das ferramentas de aprendizado de máquina.

6 CONSIDERAÇÕES FINAIS

Até onde sabemos, este é o primeiro trabalho que relata a eficiência do aprendizado de máquina para uma amostra de baixa sensibilidade térmica contendo íons de neodímio. Os trabalhos mencionados na seção 5.1 registraram importantes resultados nesse mesmo contexto ou contexto similar, dando suporte teórico para a realização desta dissertação.

Os resultados aqui discutidos se encontram em processo de submissão em forma de artigo em revista internacional. Essencialmente detalhamos a técnica de aprendizado de máquina, com algoritmos não muito sofisticados, para tratamento de dados experimentais de termometria óptica.

Neste trabalho concluímos que o aprendizado de máquina pode lidar muito bem com ruídos experimentais intrínsecos e, mesmo para amostra de baixa sensibilidade térmica, isto é, que o espectro tenha pouca variação com o aumento ou diminuição da temperatura, ainda assim o aprendizado de máquina é capaz de realizar boas previsões da temperatura. Um dos fatores que é responsável pela boa resposta do ML comparado aos outros métodos, é que a forma como o algoritmo é construído (rede neural, por exemplo) leva em conta todos os pontos inicialmente como dados independentes, se livrando da colinearidade existente nos dados. Embora o tema deste trabalho tenha sido “aprendizado de máquina” (*machine learning*), o algoritmo usado também pode ser entendido como “*deep learning*” uma vez que usamos algoritmo de redes neurais multicamadas (ver apêndice B).

Outras observações que valem a pena destacar é, em primeiro, que o ML consegue lidar com as mudanças espectrais devido ao aquecimento e resfriamento, isto é, o ML consegue buscar parâmetros na etapa de treino mesmo que o espectro passe por mudanças como deslocamento ou estreitamento de banda. Em segundo, os resultados obtidos aqui devem ser reprodutíveis graças a não-fotodegradação das partículas, isso deve ser válido para outros experimentos. Por fim, destacamos, nas perspectivas a seguir, algumas propostas que podem ser trabalhadas seguindo o trajeto desse trabalho.

Os códigos estão disponíveis no apêndice A. Se porventura o leitor precise de uma versão futura, pode entrar em contato com o autor da dissertação.

REFERÊNCIAS

- 3.3. Metrics and scoring: quantifying the quality of predictions.** Disponível em: <https://scikit-learn/stable/modules/model_evaluation.html>. Acesso em: 23 mar. 2023.
- ADAMCZAK, R.; POROLLO, A.; MELLER, J. Accurate prediction of solvent accessibility using neural networks–based regression. **Proteins: Structure, Function, and Bioinformatics**, v. 56, n. 4, p. 753–767, 2004.
- ALENCAR, M. A. R. C. et al. Er³⁺-doped BaTiO₃ nanocrystals for thermometry: Influence of nanoenvironment on the sensitivity of a fluorescence based temperature sensor. **Applied Physics Letters**, v. 84, n. 23, p. 4753–4755, 7 jun. 2004.
- ALLISON, S. W.; GILLIES, G. T. Remote thermometry with thermographic phosphors: Instrumentation and applications. **Review of Scientific Instruments**, v. 68, n. 7, p. 2615–2650, 1997.
- ALREBDI, T. A. et al. Comparison of Performance between Single- and Multiparameter Luminescence Thermometry Methods Based on the Mn⁵⁺ Near-Infrared Emission. **Sensors (Basel, Switzerland)**, v. 23, n. 8, p. 3839, 9 abr. 2023.
- ANGRA, S.; AHUJA, S. **Machine learning and its applications: A review**. 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC). **Anais...** Em: 2017 INTERNATIONAL CONFERENCE ON BIG DATA ANALYTICS AND COMPUTATIONAL INTELLIGENCE (ICBDAC). mar. 2017.
- BAKER, S. N.; MCCLESKEY, T. M.; BAKER, G. A. An ionic liquid-based optical thermometer. Em: [s.l.] ACS Publications, 2005.
- BRITES, C. D.; BALABHADRA, S.; CARLOS, L. D. Lanthanide-based thermometers: at the cutting-edge of luminescence thermometry. **Advanced Optical Materials**, v. 7, n. 5, p. 1801239, 2019.
- BRITES, C. D. S.; MILLÁN, A.; CARLOS, L. D. Chapter 281 - Lanthanides in Luminescent Thermometry. Em: JEAN-CLAUDE, B.; VITALIJ K., P. (Eds.). **Handbook on the Physics and Chemistry of Rare Earths**. Including Actinides. [s.l.] Elsevier, 2016. v. 49p. 339–427.
- CUI, J. et al. Convolutional neural networks open up horizons for luminescence thermometry. **Journal of Luminescence**, v. 256, p. 119637, 1 abr. 2023.
- DA SILVA, J. F. et al. Photon-avalanche-like upconversion in NdAl₃(BO₃)₄ nanoparticles excited at 1064 nm. **Applied Physics Letters**, v. 117, n. 15, p. 151102, 12 out. 2020.
- DA SILVA, J. F.; JACINTO, C.; MOURA, A. L. Giant sensitivity of an optical nanothermometer based on parametric and non-parametric emissions from Tm³⁺ doped NaNbO₃ nanocrystals. **Journal of Luminescence**, v. 226, p. 117475, 1 out. 2020.

DEJONG, G. **Generalizations based on explanations**. Proceedings of the 7th international joint conference on Artificial intelligence - Volume 1. **Anais...: IJCAI'81**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 24 ago. 1981. . Acesso em: 30 dez. 2022

DRAMICANIN, M. **Luminescence thermometry: methods, materials, and applications**. [s.l.] Woodhead Publishing, 2018.

GÁLICO, D. A.; MAZALI, I. O.; SIGOLI, F. A. Nanothermometer based on intensity variation and emission lifetime of europium(III) benzoylacetate complex. **Journal of Luminescence**, v. 192, p. 224–230, 1 dez. 2017.

GALINDO, J. A. O. et al. Influence of the surrounding medium on the luminescence-based thermometric properties of single Yb³⁺/Er³⁺ codoped yttria nanocrystals. **Nanoscale Advances**, v. 3, n. 21, p. 6231–6241, 27 out. 2021.

GÉRON, A. **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow**. [s.l.] O'Reilly Media, Inc., 2022.

HANSON, H. G. Fluorescence Intensity Ratio of Sodium Doublet Observed in the Optical Dissociation of Sodium Iodide Vapor. **The Journal of Chemical Physics**, v. 27, n. 2, p. 491–494, ago. 1957.

HERNÁNDEZ-RODRÍGUEZ, M. A. et al. Comparison of the sensitivity as optical temperature sensor of nano-perovskite doped with Nd³⁺ ions in the first and second biological windows. **Sensors and Actuators B: Chemical**, v. 255, p. 970–976, 1 fev. 2018.

How Does a Spectrometer Work? , 24 maio 2017. Disponível em: <<https://www.youtube.com/watch?v=OI3plvLhVcc>>. Acesso em: 6 jan. 2023

JOHRI, P.; VERMA, J. K.; PAUL, S. **Applications of Machine Learning**. [s.l.] Springer Nature, 2020.

KAMEI, Y. et al. Infrared laser-mediated gene induction in targeted single cells in vivo. **Nature Methods**, v. 6, n. 1, p. 79–81, jan. 2009.

KHALID, A. H.; KONTIS, K. 2D surface thermal imaging using rise-time analysis from laser-induced luminescence phosphor thermometry. **Measurement Science and Technology**, v. 20, n. 2, p. 025305, 2009.

KIM, Y. et al. **Low-Cost Ultrasound Thermometry for HIFU Therapy Using CNN**. 2018 IEEE International Ultrasonics Symposium (IUS). **Anais... Em: 2018 IEEE INTERNATIONAL ULTRASONICS SYMPOSIUM (IUS)**. out. 2018.

KINGSFORD, C.; SALZBERG, S. L. What are decision trees? **Nature Biotechnology**, v. 26, n. 9, p. 1011–1013, set. 2008.

KUSAMA, H.; SOVERS, O. J.; YOSHIOKA, T. Line Shift Method for Phosphor Temperature Measurements. **Japanese Journal of Applied Physics**, v. 15, n. 12, p. 2349, 1 dez. 1976.

LAIA, A. S. et al. Temperature sensing with Er³⁺ doped Y₂O₃ nanoparticles operating within the 1st and 2nd biological window: The influence of particle size on the relative sensitivity of thermally decoupled levels. **Journal of Alloys and Compounds**, v. 926, p. 166816, 10 dez. 2022.

LEWIS, C. et al. Use of Machine Learning with Temporal Photoluminescence Signals from CdTe Quantum Dots for Temperature Measurement in Microfluidic Devices. **ACS Applied Nano Materials**, v. 3, n. 5, p. 4045–4053, 22 maio 2020.

LIN, J. S. et al. Multiparametric fat–water separation method for fast chemical-shift imaging guidance of thermal therapies. **Medical Physics**, v. 40, n. 10, p. 103302, 2013.

LIU, L. et al. Fluorescence spectra shape based dynamic thermometry. **Applied Physics Letters**, v. 104, n. 3, p. 031902, 21 jan. 2014.

LOJPUR, V.; NIKOLIĆ, G.; DRAMIĆANIN, M. D. Luminescence thermometry below room temperature via up-conversion emission of Y₂O₃:Yb³⁺,Er³⁺ nanophosphors. **Journal of Applied Physics**, v. 115, n. 20, p. 203106, 28 maio 2014.

MACIA, E. Thermoelectric materials: advances and applications. 2015.

MALLAWAARACHCHI, S. et al. Machine learning based temperature prediction of poly(N-isopropylacrylamide)-capped plasmonic nanoparticle solutions. **Physical Chemistry Chemical Physics**, v. 21, n. 44, p. 24808–24819, 18 nov. 2019.

MATURI, F. E. et al. Going Above and Beyond: A Tenfold Gain in the Performance of Luminescence Thermometers Joining Multiparametric Sensing and Multiple Regression. **Laser & Photonics Reviews**, v. 15, n. 11, p. 2100301, 2021.

MEHTA, P. et al. A high-bias, low-variance introduction to Machine Learning for physicists. **Physics Reports**, A high-bias, low-variance introduction to Machine Learning for physicists. v. 810, p. 1–124, 30 maio 2019.

MEI, J. et al. Body temperature measurement in mice during acute illness: implantable temperature transponder versus surface infrared thermometry. **Scientific Reports**, v. 8, n. 1, p. 3526, 23 fev. 2018.

MÜLLER, I. **A History of Thermodynamics: The Doctrine of Energy and Entropy**. [s.l.] Springer Science & Business Media, 2007.

PRINCIPAIS TERMOMETROS E COMO FUNCIONAM. , 13 nov. 2022. Disponível em: <<https://www.youtube.com/watch?v=2cBdTANj6xo>>. Acesso em: 21 nov. 2022

Professor's perceptron paved the way for AI – 60 years too soon. Disponível em: <<https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon>>. Acesso em: 30 dez. 2022.

PUGINA, R. S. et al. Nd³⁺:YAG microspheres powders prepared by spray pyrolysis: Synthesis, characterization and random laser application. **Materials Chemistry and Physics**, v. 269, p. 124764, 1 set. 2021.

QIAN, J. et al. Machine learning-assited optical thermometer for continuous temperature analysis inside molten metal. **Sensors and Actuators A: Physical**, v. 322, p. 112626, 1 maio 2021.

RAKOV, N.; MACIEL, G. S. Nd³⁺-doped amorphous calcium yttrium silicate ceramic powder for near-infrared thermometry. **Applied Physics A**, v. 127, n. 5, p. 363, 22 abr. 2021.

RASCHKA, S. **What is the difference between a Perceptron, Adaline, and neural network model?** Disponível em: <<https://sebastianraschka.com/faq/docs/diff-perceptron-adaline-neuralnet.html>>. Acesso em: 30 dez. 2022.

REICHL, L. E. **A modern course in statistical physics**. American Association of Physics Teachers, , 1999.

REN, T. et al. Machine learning applied to retrieval of temperature and concentration distributions from infrared emission measurements. **Applied Energy**, v. 252, p. 113448, 15 out. 2019.

SANTOS, E. P. et al. Temperature triggering a photon-avalanche-like mechanism in NdAl₃(BO₃)₄ particles under excitation at 1064 nm. **Journal of Luminescence**, p. 118645, 2021.

ŠEVIĆ, D. et al. Temperature effects on luminescent properties of Sr₂CeO₄:Eu³⁺ nanophosphor: A machine learning approach. **Tehnika**, v. 75, n. 3, p. 279–283, 2020.

SHARKAWY, A.-N. Principle of Neural Network and Its Main Types: Review. **Journal of Advances in Applied & Computational Mathematics**, v. 7, p. 8–19, 20 ago. 2020.

sklearn.ensemble.ExtraTreesRegressor. Disponível em: <<https://scikit-learn/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html>>. Acesso em: 17 jan. 2023.

TURING, A. M. I.—COMPUTING MACHINERY AND INTELLIGENCE. **Mind**, v. LIX, n. 236, p. 433–460, 1 out. 1950.

Types of Thermometer - Uses of Thermometer - Thermometer Types. , 26 jul. 2019. Disponível em: <<https://www.youtube.com/watch?v=SoYOMi8-fTY>>. Acesso em: 21 nov. 2022

Unsupervised Machine learning - Javatpoint. Disponível em: <<https://www.javatpoint.com/unsupervised-machine-learning>>. Acesso em: 4 jan. 2023.

VAN SWIETEN, T. P.; MEIJERINK, A.; RABOUW, F. T. Impact of Noise and Background on Measurement Uncertainties in Luminescence Thermometry. **ACS Photonics**, v. 9, n. 4, p. 1366–1374, 20 abr. 2022.

WANG, C. et al. A review and outlook of ratiometric optical thermometer based on thermally coupled levels and non-thermally coupled levels. **Journal of Alloys and Compounds**, v. 894, p. 162494, 2022.

YAP, S. V. et al. Decay time characteristics of $\text{La}_2\text{O}_2\text{S}:\text{Eu}$ and $\text{La}_2\text{O}_2\text{S}:\text{Tb}$ for use within an optical sensor for human skin temperature measurement. **Applied Optics**, v. 47, n. 27, p. 4895–4899, 20 set. 2008.

ZHAO, D. et al. 808 nm-Light-Excited Near-Infrared Luminescent Lanthanide Metal–Organic Frameworks for Highly Sensitive Physiological Temperature Sensing. **Chemistry – A European Journal**, v. 26, n. 14, p. 3145–3151, 2020.

APÊNDICE A – CÓDIGOS

Detalhes ou dúvidas sobre o código podem ser encaminhadas para o e-mail do autor (emanuel.pinheiro@ufpe.br).

Os códigos aqui disponibilizados têm autoria dos professores André Moura, Askery Canabarro e o autor da dissertação.

Código 1 – Neste código pode se realizar a análise dos dados pelo algoritmo de machine learning e gerar uma planilha de resultados previstos pela máquina. Desse modo comparamos com os resultados alvo, cujos resultados foram colocados nesta dissertação. O código foi construído em linguagem Python.

```
#-----
#PARTE 1: Procurando modelo pelo auto Machine Learning, usando o TPOT
#-----

!pip install tpot

from tpot import TPOTRegressor
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

from sklearn.model_selection import train_test_split

file = pd.read_csv('data.dat', delimiter = '\t', header = None)

# encontra posições para cortar os dados
lamb_inicial = 860; lamb_final = 960 #nm
lamb_inicial = np.absolute(file[0][1:] - lamb_inicial); lamb_inicial =
lamb_inicial.argmin()
lamb_final = np.absolute(file[0][1:] - lamb_final); lamb_final = lamb_final.argmin()

#corta os dados em lamb_inicial e lamb_final
file.drop(range(lamb_final,len(file)),inplace = True);
file.drop(range(0,lamb_inicial),inplace = True)

file.isnull().values.any()
df = file
temperatura.dropna(inplace=True)
X = df.iloc[:,1 :df.shape[1]].values
X = np.transpose(X)
```

```

y = temperatura.values.reshape((-1,1))

X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    train_size=0.8, random_state=2)

# TPOT cfg
tpot = TPOTRegressor(verbosity=2,
                    scoring="neg_mean_squared_error",
                    random_state=7,
                    generations=15,
                    population_size=15,
                    early_stop=5)

%%time
# fit
tpot.fit(X_train, y_train)

#-----

#PARTE 2: Usando o modelo encontrado nos dados experimentais e guardando as
previsões realizadas pelo machine learning
#-----

from sklearn.ensemble import ExtraTreesRegressor
from sklearn.linear_model import RidgeCV
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline, make_union
from sklearn.preprocessing import RobustScaler
from sklearn.svm import LinearSVR
from tpot.builtins import StackingEstimator
from tpot.export_utils import set_param_recursive
from sklearn.neural_network import MLPRegressor
from sklearn.ensemble import RandomForestRegressor

# Average CV score on the training set was: -3.163778017859067
exported_pipeline = make_pipeline(
    StackingEstimator(estimator=LinearSVR(C=0.0001, dual=False, epsilon=0.0001,
    loss="squared_epsilon_insensitive", tol=1e-05)),
    StackingEstimator(estimator=RidgeCV()),
    RobustScaler(),
    #ExtraTreesRegressor(bootstrap=False, max_features=0.8, min_samples_leaf=9,
    min_samples_split=12, n_estimators=100)
    MLPRegressor()
    #RandomForestRegressor()
)
# Fix random state for all the steps in exported pipeline
set_param_recursive(exported_pipeline.steps, 'random_state', 7)

exported_pipeline.fit(X_train, y_train)

```

```

#results = exported_pipeline.predict(testing_features)
# all points
results = exported_pipeline.predict(X)

# TPOT Predict
y_pred = exported_pipeline.predict(X_test)
#
n = 9
s = int(len(X_test)/n)
p = s - 1
print(s)
for i in range (n):
    #print(round(y_test[p*(i+1)][0],1))
    LABEL = str(round(y_test[p*(i+1)][0],1)) + ' (' + str(round(y_pred[p*(i+1)],1))+ ')'
    plt.plot(file[0], X_test[p*(i+1)], label = LABEL)
    #print(round(y_test[p*(i+1)][0],1))
plt.legend(loc = 'upper right')
plt.xlabel('Wavelength (nm)', size=12); plt.ylabel('Intensity (arb. units)', size=12)
plt.savefig('ML2 espect.png', dpi = 300, bbox_inches='tight')
plt.show()

import matplotlib
matplotlib.rcParams.update({'font.size': 15, 'font.family': 'serif'})

i = list(range(len(y)))
ind = np.argsort(y.ravel())
a = y[ind]
b = results[ind]
plt.plot(b,a)
plt.xlabel('Target temperature (°C)')
plt.ylabel('ML calc. temperature (°C)')
m, n = np.polyfit(b, a, 1)
plt.plot(b, m*b+n, c = 'orange')
plt.savefig('yab10ML.png')
title = 'r2: '+str(round(R2,4))
i = list(range(len(y)))
ind = np.argsort(y.ravel())
a = y[ind]
b = results[ind]
plt.plot(i,b, label = 'preds')
plt.plot(i,a, label = 'target')
title = 'r2: '+str(round(R2,4))
plt.title(title)
plt.legend(loc = 'best');

tempo_s = []

for i in range(len(temperatura)):
    tempo_s.append(i*0.3)

```

```

results = exported_pipeline.predict(X)
plt.plot(tempo_s,results, label = 'Calculated'); plt.plot(tempo_s,temperatura, label =
'Target')

plt.xlabel('Time (s)'); plt.ylabel('Temperature (°C)'); plt.legend(loc = 'best')
plt.legend(loc = 'upper right')
plt.savefig('Temp espect.png', dpi = 300, bbox_inches='tight')
print(max(tempo_s))

from scipy.stats import norm

a = np.reshape(a,len(a))
bins = 80
mu, sigma = norm.fit(b-a)
plt.hist(b-a, bins, density=True, facecolor='b', alpha=1)

plt.xlabel(r'\Delta T$ (°C)')
plt.ylabel('Relative frequency')
xmin, xmax = plt.xlim(); x = np.linspace(xmin, xmax, 100); p = norm.pdf(x, mu,
sigma); plt.plot(x, p, 'r', linewidth=2)
plt.xlim(-37.17,43.24)
plt.title('sigma = {:.2}'.format(sigma))

plt.show()
print(sigma)
print(a)

mu, sigma = norm.fit(results-temperatura)
plt.hist(results-temperatura,bins, density = True)
xmin, xmax = plt.xlim(); x = np.linspace(xmin, xmax, 100); p = norm.pdf(x, mu,
sigma); plt.plot(x, p, 'r', linewidth=2)
#plt.xlim(-37.17,43.24)
plt.title('mu = {:.2}, sigma = {:.2}'.format(mu, sigma))

#testar terceiro ciclo
file = pd.read_csv('NOVO CICLO 3.dat',delimiter = '\t', header = None)

temperatura = file.iloc[0,:]
temperatura.dropna(inplace=True)
lamb_inicial = lamb_inicial_2; lamb_final = lamb_final_2
lamb_inicial = np.absolute(file[0][1:] - lamb_inicial); lamb_inicial =
lamb_inicial.argmin()
lamb_final = np.absolute(file[0][1:] - lamb_final); lamb_final = lamb_final.argmin()

#corta os dados em lamb_inicial e lamb_final
file.drop(range(lamb_final,len(file)),inplace = True);
file.drop(range(0,lamb_inicial),inplace = True)
df = file
X = df.iloc[:,1 :df.shape[1]].values
X = np.transpose(X)

```

```

y = temperatura.values.reshape((-1,1))

results = exported_pipeline.predict(X)

T_calc = results.reshape(len(results))
mu, sigma = norm.fit(T_calc-temperatura)
plt.hist(T_calc-temperatura,bins, density = True)
xmin, xmax = plt.xlim(); x = np.linspace(xmin, xmax, 100); p = norm.pdf(x, mu,
sigma); plt.plot(x, p, 'r', linewidth=2)

plt.title('sigma = {:.2f}'.format(sigma))
plt.title('mu = {:.2}, sigma = {:.2}'.format(mu, sigma))
df = pd.DataFrame(np.concatenate((np.reshape(np.array(temperatura),(1,-
1)),np.reshape(np.array(T_calc),(1,-1))), axis = 0))
df = df.transpose()
df.to_csv('ML.csv', index=False)

#-----
#PARTE 3: Comparar resultados do ML com LIR e MLR
#-----

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import scipy
import matplotlib
from scipy.signal import find_peaks
from itertools import combinations
from scipy.stats import norm
import matplotlib.mlab as mlab
import matplotlib as mpl

matplotlib.rcParams.update({'font.size': 15, 'font.family': 'serif'})

file = pd.read_csv('NOVO CICLO 3.dat',delimiter = '\t', header = None)

temperatura = file.iloc[0,:]

#histograma da temperatura
plt.hist(temperatura[1:], bins = 50); plt.xlabel('Temperatura °C'); plt.ylabel('Counts')
plt.show()

#espectro completo
plt.plot(file[0][1:], file[1][1:]); plt.title('completo'); plt.xlabel('Wavelength (nm)');
plt.ylabel('Intensity (arb. units)')
plt.show()

tempo_s = []

```

```

for i in range(len(temperatura)-1):
    tempo_s.append(i*0.3)

plt.plot(tempo_s,temperatura[1:]); plt.xlabel('Time (s)'); plt.ylabel('Target Temp. (°C)')
# encontra posições para cortar os dados
lamb_inicial = 840; lamb_final = 955 #nm
lamb_inicial = np.absolute(file[0][1:] - lamb_inicial); lamb_inicial =
lamb_inicial.argmin()
lamb_final = np.absolute(file[0][1:] - lamb_final); lamb_final = lamb_final.argmin()

#corta os dados em lamb_inicial e lamb_final
file.drop(range(lamb_final,len(file)),inplace = True);
file.drop(range(0,lamb_inicial),inplace = True)
file.head()

peaks, _ = find_peaks(file[10031], height=900)
lambd = file.iloc[:,0]

lamb1 = 2 #1
lamb2 = -1 #-2
x = file.iloc[peaks[lamb1],1:]; y = file.iloc[peaks[lamb2],1:]
max_peak = file.iloc[peaks[lamb1],0]; sec_peak = file.iloc[peaks[lamb2],0]
#max_peak deveria ser chamada de first_peak

razao = x / y
ajuste2 = scipy.stats.linregress(temperatura[1:],razao); razao_calc =
ajuste2[1]+ajuste2[0]*temperatura[1:]

fig, axs = plt.subplots(3, sharex = True); plt.xlabel('Temperature (°C)')

axs[0].set_ylabel('I{:.0f}/I{:.0f}'.format(max_peak, sec_peak));
axs[1].set_ylabel('I'+str(max_peak)); axs[2].set_ylabel('I'+str(sec_peak))
axs[0].plot(temperatura[1:], razao, 'tab:red'); axs[1].plot(temperatura[1:],x,
'tab:purple'); axs[2].plot(temperatura[1:],y)
axs[0].plot(temperatura[1:],razao_calc)
axs[0].set_title('Max $S_r$ = {:.2f} %'.format(100*max(ajuste2[0]/(razao_calc))))
plt.show()

tempo_s = []

for i in range(len(temperatura)-1):
    tempo_s.append(i*0.3)

fig, axs = plt.subplots(4, sharex=True, figsize=(5, 6))
plt.xlabel('Time (s)'); axs[3].set_ylabel('Temp (°C)')
axs[2].set_ylabel('R') #axs[2].set_ylabel('I{:.0f}/I{:.0f}'.format(max_peak, sec_peak))
axs[2].plot(tempo_s, razao,'g'); axs[0].plot(tempo_s,x, 'k'); axs[1].plot(tempo_s,y,
'tab:orange')

```

```

axs[0].set_ylabel('I{:.0f}'.format(max_peak));
axs[1].set_ylabel('I{:.0f}'.format(sec_peak))

temp_calc = (razao - ajuste2[1])/ajuste2[0]
axs[3].plot(tempo_s,temp_calc)

tempo_s = []

for i in range(len(temperatura)):
    tempo_s.append(i*0.3)

axs[3].plot(tempo_s,temperatura, 'tab:red')
plt.savefig('3.png', dpi = 300, bbox_inches='tight')
#plt.savefig('3.svg', bbox_inches='tight')
plt.show()

ajusteR = scipy.stats.linregress(temperatura[1:],razao)

plt.plot(temperatura[1:],          razao);          plt.plot(temperatura[1:],
ajusteR[0]*temperatura[1:]+ajusteR[1], linewidth = 4.0); plt.title('Sr =
{:.2f}'.format(100*max(ajusteR[0]/(ajusteR[0]*temperatura[1:]+ajusteR[1])))
plt.xlabel('Targed temperature (°C)'); plt.ylabel('I{/I}'.format(max_peak, sec_peak));
plt.show()
plt.plot(temperatura[1:],          100*ajusteR[0]/(ajusteR[0]*temperatura[1:]+ajusteR[1]),
linewidth = 4.0); plt.show()

#fig, axs = plt.subplots(21, sharex = False, sharey = False)
pearson_values = []
comb = combinations(peaks,2)
for i, j in enumerate(comb):
    lamb1 = np.where(peaks == j[0])[0][0]; lamb2 = np.where(peaks == j[1])[0][0]
    x = file.iloc[peaks[lamb1],1:]; y = file.iloc[peaks[lamb2],1:]
    max_peak = file.iloc[peaks[lamb1],0]; sec_peak = file.iloc[peaks[lamb2],0]
#max_peak deveria ser chamada de first_peak

    razao = x / y
    pearson = x.corr(y); ajuste = scipy.stats.linregress(x,y)
    pearson_values.append(pearson)

    plt.plot(x, y, 'o', label= f'Pearson = {np.round(pearson,2)}');
plt.plot(x,ajuste[0]*x+ajuste[1], linewidth = 4.0)
plt.xlabel('I_'+str(max_peak)+'(arb. units)'); plt.ylabel('I_'+str(sec_peak)+'(arb.
units)')
plt.legend()
print(i)
plt.show()

funcoes = pd.DataFrame()

```

```

razoes = pd.DataFrame()
funcoes[0] = 0; funcoes[1] = 0
comb = combinations(peaks,2)
Temperatura_final = pd.DataFrame()
razoes_calcDF = pd.DataFrame()

coeficiente = [] #usar para calculo do dT no MLR
desviosP = [] #usar para calculo do dT no MLR

for i, j in enumerate(comb):

    lamb1 = np.where(peaks == j[0])[0][0]; lamb2 = np.where(peaks == j[1])[0][0]
    x = file.iloc[peaks[lamb1],1:]; y = file.iloc[peaks[lamb2],1:]
    max_peak = file.iloc[peaks[lamb1],0]; sec_peak = file.iloc[peaks[lamb2],0]
    #max_peak deveria ser chamada de first_peak

    razao = x / y
    ajusteTESTE = scipy.stats.linregress(temperatura[1:],razao)

    #if i == 10 or i==12 or i==15 or i==17 or i==25 or i==27: #usar para calculo do dT
no MLR
    if ajusteTESTE[0]<0: #esses aqui troca o sinal do coeficiente angular
        razao = y/x
        razoes[i] = razao
        ajuste2 = scipy.stats.linregress(temperatura[1:],razao); razao_calc =
ajuste2[1]+ajuste2[0]*temperatura[1:]
        #coeficiente.append(1/ajuste2[0]) #usar para calculo do dT no MLR
        #razoes_calcDF[i] = razao_calc #usar para calculo do dT no MLR

        #slope, intercept, r_value, p_value, std_err =
scipy.stats.linregress(temperatura[1:],razao) #usar para calculo do dT no MLR
        #std_residuals = np.sqrt((1-r_value**2)*np.var(razao)) #usar para calculo do dT no
MLR
        #desviosP.append(std_residuals) #usar para calculo do dT no MLR

        plt.xlabel('Temperature (°C)')
        plt.ylabel('I'+str(max_peak)+'/I'+str(sec_peak))
        #plt.title('# {} slope{} intercept {} Sr={}'.format(i,ajuste2[0],
ajuste2[1],100*max(ajuste2[0]/(razao_calc))))

        pearson = temperatura[1:].corr(razao)

        plt.plot(temperatura[1:], razao, label= f'Pearson = {np.round(pearson,2)}')
        plt.legend()
        plt.plot(temperatura[1:],razao_calc)
        funcoes.loc[len(funcoes.index)] = [ajuste2[1], ajuste2[0]]
        Temperatura_final[i] = (razao - ajuste2[1])/ajuste2[0]

        ajusteMLR = scipy.stats.linregress(razao, temperatura[1:])
        print(f'{lamb1}/{lamb2}', ajusteMLR[0]*razao_calc[1])

```

```

plt.show()
comb = combinations(peaks,2)

#parece está trocado R com temperatura calculada
pearson_temperaturas = []
for i, j in enumerate(comb):
    pearson_temperaturas.append(temperatura[1:].corr(Temperatura_final.iloc[:,i]))
    diferenca = (Temperatura_final.iloc[:,i]-temperatura)**2
    sigma_T = (sum(diferenca[1:])/len(temp_calc))**0.5
    plt.plot(temperatura[1:], Temperatura_final.iloc[:,i]); plt.title('Pair #{} pearson {:.2f}
sigma {:.1f}'.format(i,pearson_temperaturas[i],sigma_T))
    plt.plot(temperatura[1:],temperatura[1:], linewidth = 4.0)
    plt.xlabel('Target temperature (°C)'); plt.ylabel('Calculated temperature (°C)')
    ajusteN = scipy.stats.linregress(temperatura[1:],Temperatura_final.iloc[:,i])
    #print('Sr = ', max(100*ajusteN[0]/(ajusteN[0]*temperatura[1:]+ajusteN[1])))
plt.show()

pearson_temperaturas_threshold = 0.85
deletar = []
Ndelet = []
for i, j in enumerate(pearson_temperaturas):
    print(j)
    if j < pearson_temperaturas_threshold:
        deletar.append(i)
    else:
        Ndelet.append(i)

# abrir dados ML
ML = pd.read_csv('ML.csv')
#histogramas
bins = 40

#ML
plt.hist(ML.iloc[:,0] - ML.iloc[:,1], bins, density=True, facecolor='tab:green', label = 'M.
Learn.')

#MLR
plt.hist(temperatura[1:]-Temp_media, bins, density=True, facecolor='r', label =
'MLR')

#single fit
plt.hist(temperatura[1:]-temp_comp, bins, density=True, facecolor='tab:blue', label =
'LIR')

xmin, xmax = plt.xlim(); x = np.linspace(xmin, xmax, 1000);

#ML

```

```
mu, sigma = norm.fit(ML.iloc[:,0] - ML.iloc[:,1]); print(mu, sigma); p = norm.pdf(x, mu, sigma); plt.plot(x, p, 'g', linewidth=3)
```

```
#single fitM
```

```
mu, sigma = norm.fit(temperatura[1:]-temp_comp); print(mu,sigma); p = norm.pdf(x, mu, sigma); plt.plot(x, p, 'b', linewidth=3)
```

```
#MLR
```

```
mu, sigma = norm.fit(temperatura[1:]-Temp_media); print(mu,sigma); p = norm.pdf(x, mu, sigma); plt.plot(x, p, 'tab:red', linewidth=3)
```

```
plt.xlabel(r'$\Delta T$ (°C)'); plt.ylabel('Relative frequency'); plt.legend(loc = 'best'); plt.xlim(-20,23)
```

```
plt.savefig('4.png', dpi = 300, bbox_inches='tight')
```

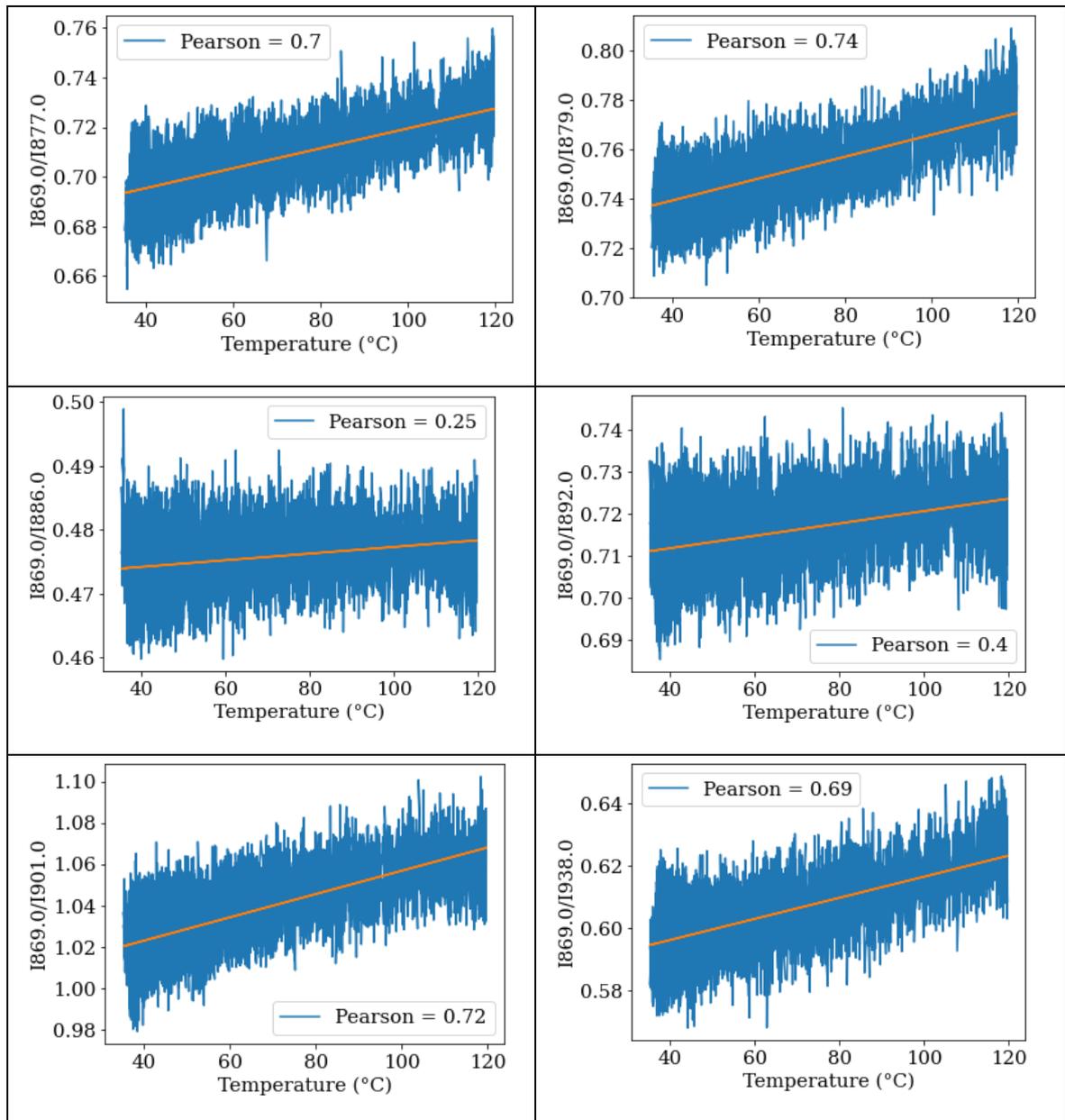
```
#plt.savefig('4.pdf', bbox_inches='tight')
```

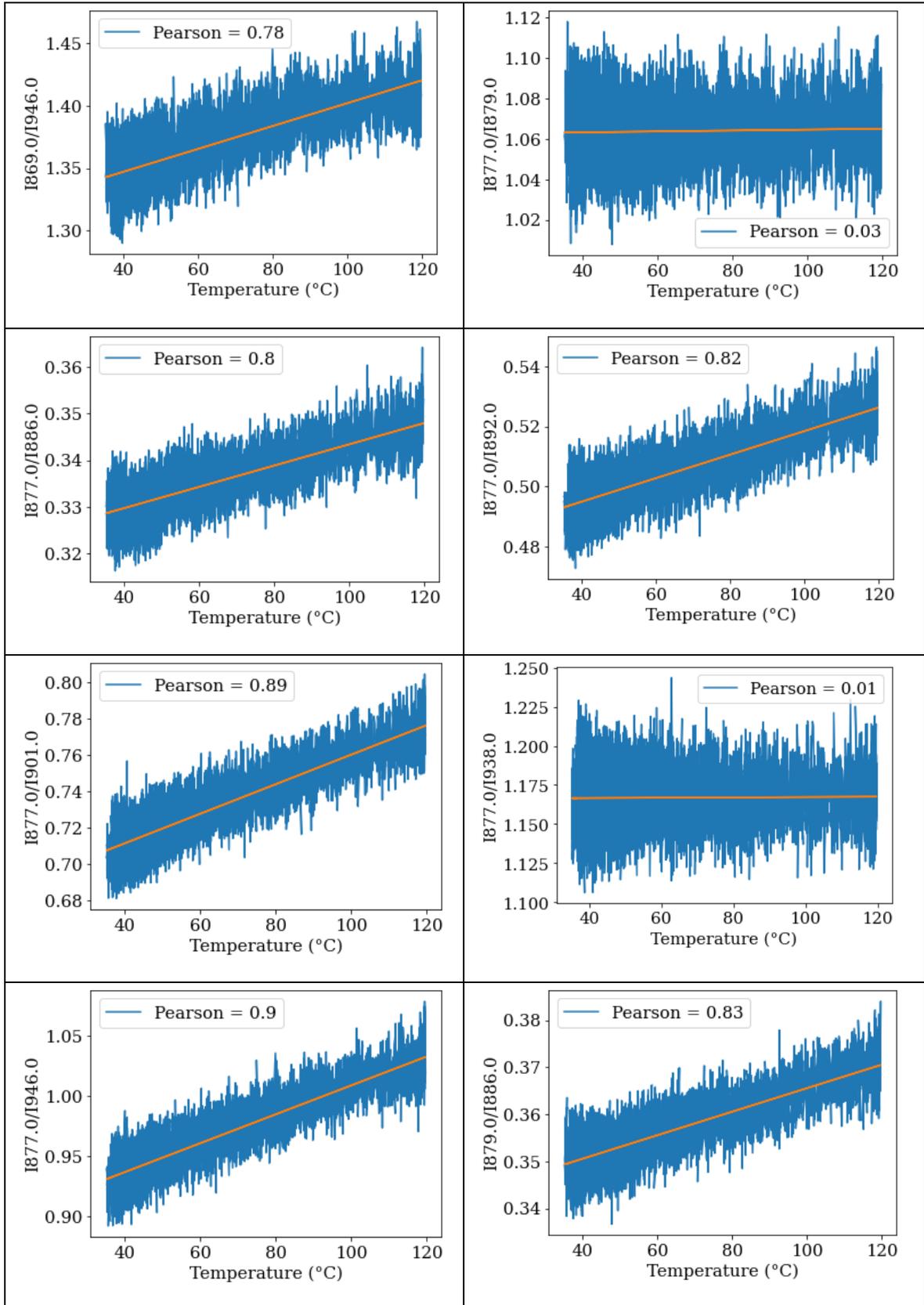
```
plt.show()
```

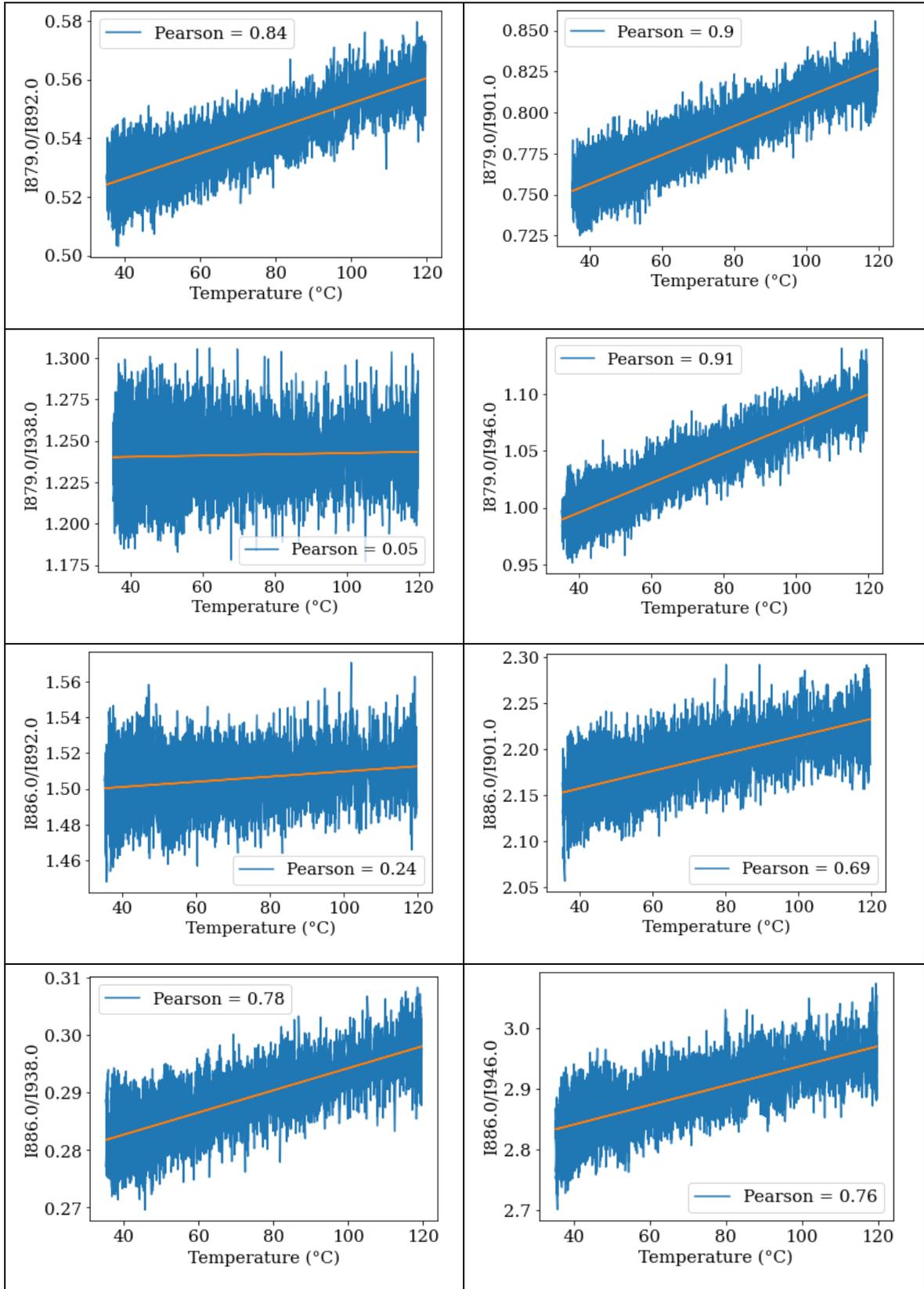
APÊNDICE B – MATERIAL SUPLEMENTAR

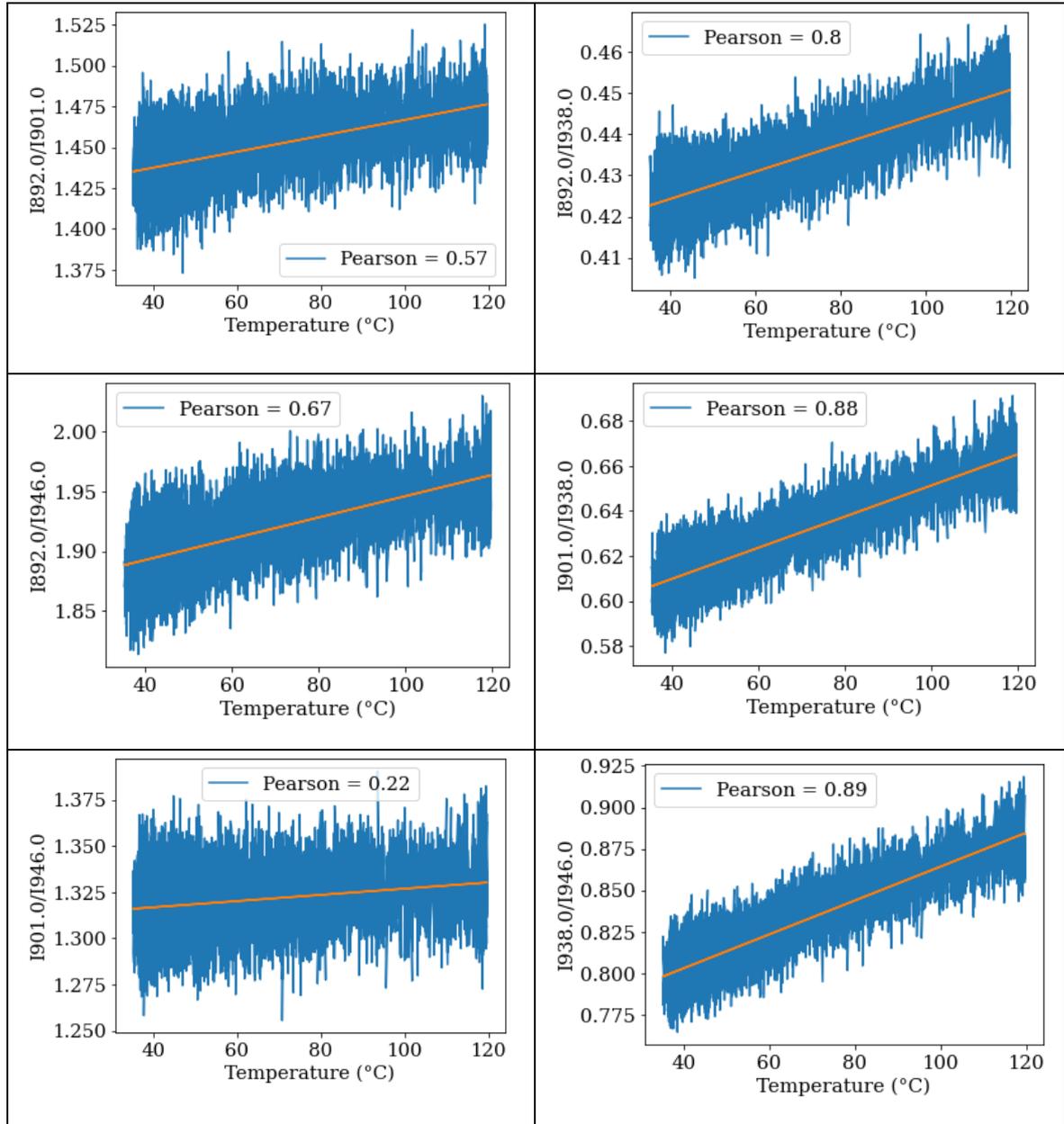
Para o método da regressão linear múltipla podemos escolher muitos parâmetros que variam linearmente com a temperatura (seção 2.3). Para este trabalho, notamos que a razão de intensidade entre dois picos, dentre aqueles que observamos na emissão, soa bem ruidoso, mas tendem a um comportamento linear. Vejamos todas as possibilidades abaixo.

Figura 30 – Razões de intensidades para todas as possíveis combinações.





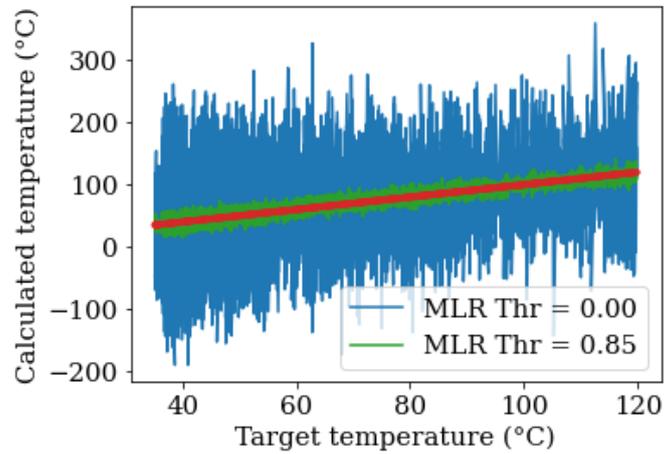




Fonte: Autor (2023).

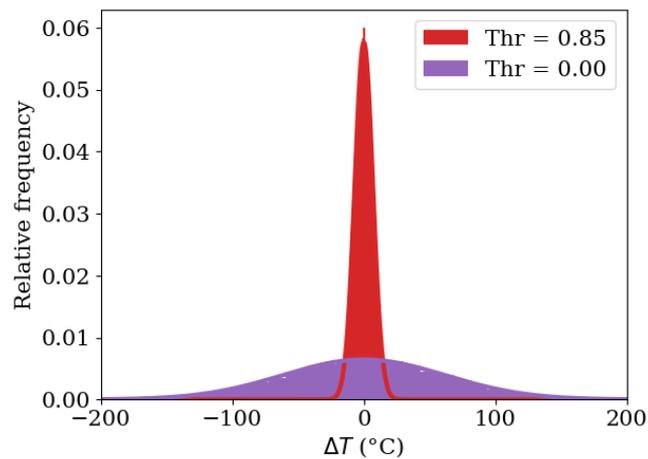
Como mencionamos no capítulo 5, o ruído faz parte da discussão e do mérito da pesquisa. Dentre todas essas possibilidades, para o método da regressão linear múltipla usamos apenas aquelas em que o coeficiente de Pearson foi igual ou maior que 0.85 (MLR Thr = 0.85). Veja (abaixo) que se usar todas as possibilidades (MLR Thr = 0.00) os ruídos associados são notavelmente maiores.

Figura 31 – Comparação da previsão para diferentes quantidades de combinações no MLR.



Fonte: Autor (2023).

Figura 32 – Histograma para Thr = 0 e Thr = 0.85.



Fonte: Autor (2023).

A rede neural utilizada foi uma multicamada com 3 camadas ocultas de 50 neurônios cada, dado que essa foi a melhor arquitetura dentre as buscadas.

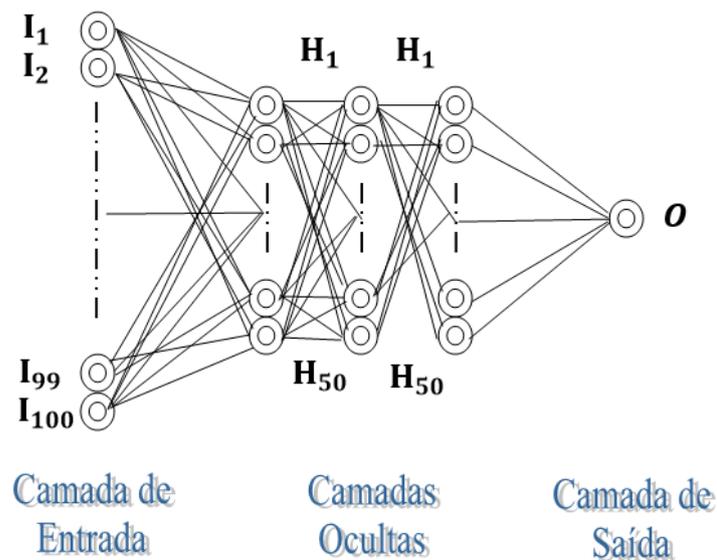
Figura 33 – Treino com diferentes arquiteturas da MLP.

ARCH	MAE	MSE	R2	Training Time	
1	(50,)	0.744240	0.932825	0.998223	00:00:39
2	(50, 50)	0.713342	0.899703	0.998286	00:01:03
3	(50, 50, 50)	0.683707	0.828663	0.998421	00:01:03
4	(100,)	0.733719	0.959845	0.998172	00:00:47
5	(100, 100)	0.742668	0.952351	0.998186	00:01:16
6	(100, 100, 100)	0.725123	0.903731	0.998278	00:01:27
7	(200,)	0.759993	1.008453	0.998079	00:01:12
8	(200, 200)	0.967153	1.571306	0.997007	00:02:07
9	(200, 200, 200)	0.768751	1.017807	0.998061	00:02:48
10	(500,)	0.726495	0.911832	0.998263	00:02:33
11	(500, 500)	0.794332	1.088428	0.997927	00:07:11
12	(500, 500, 500)	0.837641	1.204358	0.997706	00:12:09
13	(1000,)	0.799885	1.170822	0.997770	00:04:43
14	(1000, 1000)	0.950092	1.486164	0.997169	00:04:13
15	(1000, 1000, 1000)	0.763001	0.966058	0.998160	00:20:38

Fonte: Autor (2023).

Podemos representar a multicamada da seguinte forma:

Figura 34 – Esboço da MLP usada.

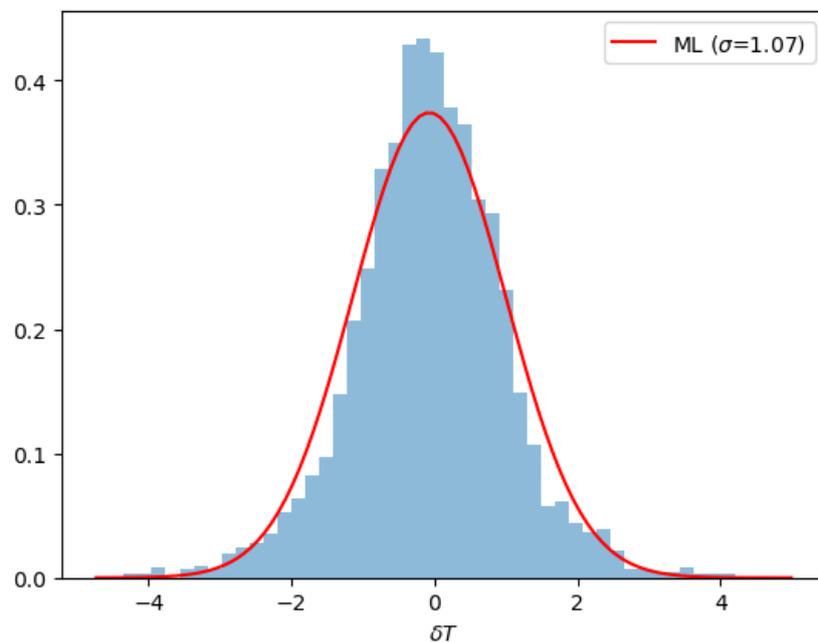


Fonte: Autor (2023).

Que para fins didáticos representam a rede neural desse problema de regressão.

Se ao invés de uma rede neural pura, usarmos um modelo de convolução (convolutional neural network), dependendo da quantidade de dados usados e da etapa usada, é possível melhorar os resultados. Nesse caso o desvio padrão foi 1.07 enquanto que os resultados do quadro 1 mostra 1.22. Mas apresentamos resultados em cima do teste de validação ao invés do conjunto de teste. O estudo usando CNN está previsto para ser concluído nos próximos meses.

Figura 35 – Resultado usando a CNN.



Fonte: Autor (2023).