



UNIVERSIDADE ESTADUAL PAULISTA

“JÚLIO DE MESQUITA FILHO”

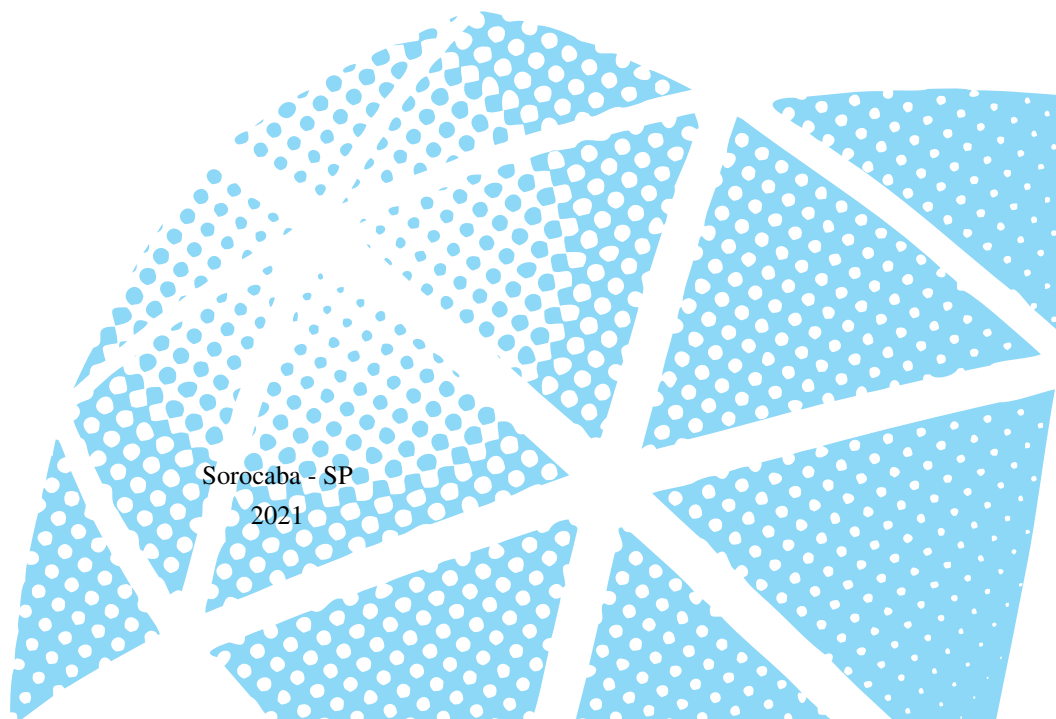
Instituto de Ciência e Tecnologia de Sorocaba

ALEXANDRE FELIPE DE OLIVEIRA

**LOCALIZAÇÃO 3D EM AMBIENTES INTERNOS COM REDES BLUETOOTH LOW
ENERGY UTILIZANDO TÉCNICAS DE APRENDIZADO DE MÁQUINA**

Sorocaba - SP

2021



ALEXANDRE FELIPE DE OLIVEIRA

**LOCALIZAÇÃO 3D EM AMBIENTES INTERNOS COM REDES BLUETOOTH LOW
ENERGY UTILIZANDO TÉCNICAS DE APRENDIZADO DE MÁQUINA**

Texto apresentado ao Programa de Pós-Graduação em Engenharia Elétrica (PGEE) do Instituto de Ciência e Tecnologia de Sorocaba como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Sorocaba - SP
2021



O481 Oliveira, Alexandre Felipe de.
Localização 3D em ambientes internos com redes Bluetooth Low
Energy utilizando técnicas de Aprendizado de Máquina / Alexandre Fe-
lipe de Oliveira. -- Sorocaba, 2021
86 p.: il., tabs., fotos

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp),
Instituto de Ciência e Tecnologia, Sorocaba
Orientador: Alexandre da Silva Simões

1.Localização. 2.Internet das coisas. 3.Inteligência artificial.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de
Ciência e Tecnologia, Sorocaba. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.



UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Sorocaba

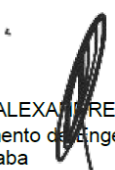
CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: LOCALIZAÇÃO EM AMBIENTES INDOOR COM REDES DE SENSORES SEM FIO

AUTOR: ALEXANDRE FELIPE DE OLIVEIRA

ORIENTADOR: ALEXANDRE DA SILVA SIMÕES

Aprovado como parte das exigências para obtenção do Título de Mestre em ENGENHARIA ELÉTRICA, área: Automação pela Comissão Examinadora:


Prof. Dr. ALEXANDRE DA SILVA SIMÕES (Participação Virtual)
Departamento de Engenharia de Controle e Automação / Instituto de Ciência e Tecnologia - UNESP - Câmpus de Sorocaba


Prof. Dr. FLÁVIO TONIDANDEL (Participação Virtual)
Ciência da Computação / Centro Universitário FEI


Prof. Dr. EDUARDO PACIÊNCIA GODOY (Participação Virtual)
Departamento de Engenharia de Controle e Automação / Instituto de Ciência e Tecnologia - UNESP - Câmpus de Sorocaba

Sorocaba, 17 de maio de 2021

Agradecimentos

Meus agradecimentos a todos os familiares, amigos, professores e funcionários do Instituto de Ciência e Tecnologia de Sorocaba, que direta ou indiretamente contribuíram para a realização deste trabalho. Em especial, dedico meus agradecimentos:

- À minha noiva Gabriela de Goes Mattozo, por todo o amor, apoio, paciência, inspiração e motivação durante todos os momentos da minha jornada no mestrado;
- Aos meus pais, Idineide e Alex, pelos ensinamentos, amor, apoio e incentivo, bases essenciais para minha evolução como pessoa desde sempre. À minha família por sempre me motivarem e apoiarem em todos os momentos;
- Ao Prof. Alexandre da Silva Simões pela sua orientação, apoio, confiança, amizade e ensinamentos durante todo o período do mestrado. As conversas e debates de ideias foram essenciais para a condução deste trabalho e para minha evolução como pesquisador;
- À Prof^a Esther Luna Colombini, por todos os ensinamentos e colaboração para a realização deste trabalho;
- Aos membros da banca examinadora, Prof. Flavio Tonidandel e Prof. Eduardo Paciência Godoy, que gentilmente aceitaram participar e colaboraram muito com esta dissertação com seu conhecimento;
- À Lilian Helena Mathilde, toda a Seção de Pós-graduação, e à Célia Maria Damasio Leles, da Secretaria da Direção do Campus. O apoio de vocês durante todo esse período foi essencial, muito obrigado por toda atenção e paciência;
- Aos amigos do GASI (Grupo de Automação e Sistemas Integrados) Murillo Ferreira, Adson Alves e Maria Tejada Begazo pela parceria, colaboração e amizade em aulas, trabalhos e pesquisa. Aos amigos Vinícius Nascimento e Felipe Crispim pelas conversas, ideias, e amizade de sempre.

Resumo

Recentemente, aplicações de localização em ambientes internos e fechados têm utilizado uma infraestrutura de redes de sensores sem fio, com tecnologias de comunicação como Wi-Fi, *Bluetooth*, *Zigbee*, UWB (*Ultra Wideband*), entre outras. Neste tipo de implementação tipicamente estão presentes um dispositivo móvel – cuja posição será estimada – e dispositivos fixos – denominados âncoras – que servem como pontos de referência. O processo de estimativa de localização é feito a partir da comunicação entre estes dois elementos. Contudo, esse sinal, em ambientes internos, está sujeito a interferências e depreciações em sua qualidade. Algoritmos de Aprendizado de Máquina têm sido aplicados nesse cenário com o intuito de estimar a posição de dispositivos a partir dos sinais das entradas das âncoras. O treinamento desses algoritmos requer uma base de dados que mapeie os sinais de entrada das múltiplas âncoras disponíveis no ambiente – tal como recebidos – para a posição do dispositivo que as recebeu. Essa abordagem é denominada de análise de cena, ou simplesmente *fingerprint*. Nesse cenário, utilizando como parâmetro de entrada os níveis de potência de sinal (RSSI) recebidos das âncoras no ambiente, é razoável assumir como hipótese que aquelas com sinais de maior potência sejam mais relevantes para os algoritmos de localização. Utilizando esta estratégia, o presente trabalho propõe a adoção de um *filtro de âncoras* como metodologia para reduzir o número de sinais a serem observados pelos algoritmos de localização. Utilizando a tecnologia sem fio *Bluetooth Low Energy* e como métrica o indicador RSSI, uma base de dados de localização 3D com diversas âncoras foi experimentalmente levantada em um ambiente interno. Com a abordagem de *filtro de âncoras* foram implementadas e testadas 3 técnicas de Aprendizado de Máquina para estimação de localização: K-vizinhos mais próximos (KNN), árvore de regressão e redes neurais artificiais. Os testes realizados demonstraram que, no contexto do ambiente e dos algoritmos escolhidos, utilizando o sinal de a partir de 3 âncoras do ambiente já é possível obter um nível satisfatório de qualidade na localização.

Palavras-chave: Localização. Internet das coisas. Inteligência artificial.

Abstract

Recently, indoor localization applications have been using wireless sensor network infrastructure, with communication technologies such as Wi-Fi, *Bluetooth*, *Zigbee*, UWB (*Ultra Wideband*) , among others. In this type of implementation there are typically a mobile device – whose position will be estimated – and fixed devices – called anchors – that serve as reference points. The location estimation process is made from the communication between these two elements. However, this signal, in indoor environments is subject to quality interference and deterioration. Machine Learning Algorithms have been applied in this scenario in order to estimate the position of devices from the anchor input signals. Training these algorithms requires a database that maps the input signals from the multiple anchors available in the environment – as received – to the position of the receiving device. This approach is called scene analysis, or simply *fingerprint*. In this scenario, using as input parameter the received signal strength indication (RSSI) detected from the anchors in the environment, it is reasonable to assume that those with higher power signals are more relevant for the localization algorithms. Using this strategy, the present work proposes the adoption of an *anchor filter* as a methodology to reduce the number of signals to be observed by the localization algorithms. Using the *Bluetooth Low Energy* wireless technology and the RSSI indicator as a metric, a 3D localization database with several anchors was experimentally built indoors. Using the *anchor filter* approach, 3 Machine Learning techniques were implemented and tested to the localization estimation task: K-Nearest Neighbors (KNN), regression tree and artificial neural networks. The tests performed showed that, in the context of the environment and the chosen algorithms, using the 3 anchors signal from the environment it is already possible to obtain a satisfactory level of quality in the localization estimation.

Keywords: Location. Internet of things. Artificial intelligence.

Lista de Figuras

Figura 1 – Arquitetura básica de uma rede de sensores sem fio.	18
Figura 2 – Espectro de frequência de algumas tecnologias <i>wireless</i>	23
Figura 3 – Comparação de complexidade de hardware x acuracidade de localização.	25
Figura 4 – Modelo de perda de sinal por distância (path loss) para um dispositivo BLE.	28
Figura 5 – Esquema da técnica de Time of Flight com Two Way Range.	29
Figura 6 – Sinal de entrada atingindo um conjunto de antenas e medição de ângulo de chegada.	30
Figura 7 – Conceito fundamental da técnica de trilateração 3D.	32
Figura 8 – Conceito fundamental da técnica de triangulação.	33
Figura 9 – Mapa e operação básica de algoritmo baseado em fingerprint.	34
Figura 10 – Etapas do algoritmo de árvore de regressão.	40
Figura 11 – Modelo matemático simples de um neurônio.	41
Figura 12 – Modelagem para funções linearmente e não linearmente separáveis.	41
Figura 13 – Esquema de uma rede neural perceptron multicamadas.	42
Figura 14 – Desenho do setup experimental para localização 3D.	46
Figura 15 – Planta baixa do ambiente de testes, com células e posição das âncoras.	48
Figura 16 – Condições de detecção e não detecção de âncoras.	51
Figura 17 – Abordagem proposta para o filtro de âncoras.	52
Figura 18 – Fluxo de trabalho com os algoritmos de Aprendizado de Máquina.	53
Figura 19 – Posição dos pontos de medição para <i>fingerprint</i> e das âncoras no laboratório do GASI.	56
Figura 20 – Suporte para medição.	57
Figura 21 – Setup experimental para coleta e criação da base de dados.	58
Figura 22 – Fluxograma do procedimento de coleta e criação da base de dados.	59
Figura 23 – Fluxo macro para o processo da criação da base de dados de fingerprint.	59
Figura 24 – Servidor web implementado na plataforma Node-RED.	61
Figura 25 – Histogramas das medidas de RSSI dos dispositivos âncoras com distância de 3 metros.	62
Figura 26 – Gráfico de boxplot das medidas de RSSI dos dispositivos âncoras com distância de 3 metros.	63
Figura 27 – Plataforma de desenvolvimento - DOIT ESP32 DEVKIT V1.	66
Figura 28 – Função de distribuição acumulada (CDF) de erro para 2 sistemas de localização.	68
Figura 29 – Arquitetura da rede neural perceptron multicamadas utilizada.	70
Figura 30 – Raiz do Erro Quadrático Médio (RMSE) x número k de vizinhos (KNN) - Filtro de âncoras.	71
Figura 31 – Raiz do Erro Quadrático Médio (RMSE) x número k de vizinhos (KNN) - Abordagem clássica.	71
Figura 32 – Raiz do Erro Quadrático Médio (RMSE) x profundidade da árvore de regressão - Filtro de âncoras.	72
Figura 33 – Raiz do Erro Quadrático Médio (RMSE) x profundidade da árvore de regressão - Abordagem clássica.	73
Figura 34 – Raiz do Erro Quadrático Médio (RMSE) x neurônios na camada oculta da rede MLP - Filtro de âncoras.	73
Figura 35 – Raiz do Erro Quadrático Médio (RMSE) x neurônios na camada oculta da rede MLP - Abordagem clássica.	74
Figura 36 – Gráficos da função de distribuição acumulada (CDF) do erro de localização - parte 1.	76
Figura 37 – Gráficos da função de distribuição acumulada (CDF) do erro de localização - parte 2.	77

Lista de Tabelas

Tabela 1 – Comparação de tecnologias de localização em ambientes internos com redes de sensores sem fio. .	24
Tabela 2 – Trabalhos na área de localização em ambientes internos com redes de sensores sem fio.	45
Tabela 3 – Posição das âncoras.	56
Tabela 4 – Estrutura da base de dados de localização 3D.	62
Tabela 5 – Resumo dos erros de localização encontrados nos cenário propostos.	74

Lista de Abreviações e Siglas

ANN	<i>Artificial Neural Networks</i>
AOA	<i>Angle of Arrival</i>
BLE	<i>Bluetooth Low Energy</i>
CDF	<i>Cumulative Distribution Function</i>
CIR	<i>Channel Impulse Response</i>
GASI	Grupo de Automação e Sistemas Integrados
GPS	<i>Global Positioning System</i>
HTTP	<i>Hyper Text Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IMU	<i>Inertial measurement unit</i>
IOT	<i>Internet of Things</i>
ISM	<i>Industrial, Scientific and Medical</i>
JSON	<i>JavaScript Object Notation</i>
KNN	<i>K-Nearest Neighbors</i>
LOS	<i>Line of Sight</i>
LR-WPAN	<i>Low Rate Wireless Personal Area Network</i>
MAC	<i>Media Access Control</i>
MLP	<i>Multilayer Perceptron</i>
MSE	<i>Mean Square Error</i>
NLOS	<i>Non Line of Sight</i>
RFID	<i>Radio Frequency Identification</i>
RMSE	<i>Root Mean Square Error</i>
RNA	Rede Neural Artificial
RSSI	<i>Received Signal Strength Indicator</i>
SIG	<i>Special Interest Group</i>
SOC	<i>System on Chip</i>
SSID	<i>Service Set Identifier</i>
TDOA	<i>Time Difference of Arrival</i>
TOA	<i>Time of Arrival</i>
TOF	<i>Time of Flight</i>

TWR	<i>Two-Way Ranging</i>
UWB	<i>Ultra Wideband</i>
WLAN	<i>Wireless Local Area Network</i>
WSN	<i>Wireless Sensor Networks</i>

Sumário

1	INTRODUÇÃO	17
1.1	Contextualização	17
1.2	Motivação	18
1.3	Objetivos	19
1.4	Organização do Trabalho	19
2	REDES DE SENSORES SEM FIO PARA LOCALIZAÇÃO EM AMBIENTES INTERNOS .	21
2.1	Bluetooth	21
2.1.1	Bluetooth Low Energy	22
2.2	Zigbee	22
2.3	WiFi (IEEE 802.11b)	23
2.4	UWB	23
2.5	Comparação de performance de localização entre redes de sensores sem fio	24
3	TÉCNICAS DE MEDIÇÃO DE DISTÂNCIA BASEADAS EM REDES DE SENSORES SEM FIO	27
3.1	Indicador de potência do sinal recebido - RSSI	27
3.2	Técnicas de medição de distância baseadas em tempo	28
3.3	Técnicas de medição de distância baseadas em recursos específicos de hardware	29
4	TÉCNICAS DE LOCALIZAÇÃO PARA AMBIENTES INTERNOS	31
4.1	Proximidade	31
4.2	Trilateração	31
4.3	Triangulação	32
4.4	Fingerprint	32
4.4.1	Treinamento	33
4.4.2	Teste	34
5	ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA LOCALIZAÇÃO INTERNA .	37
5.1	K-vizinhos mais próximos - (KNN)	37
5.2	Naive Bayes	38
5.3	Árvore de regressão	39
5.4	Redes neurais artificiais	39
5.4.1	Perceptron multicamadas (MLP)	41
5.5	Algoritmos de agrupamento de dados	42
6	TRABALHOS CORRELATOS	45
7	A ABORDAGEM DO FILTRO DE ÂNCORAS	51
7.1	Algoritmo para adequação da base de dados para a proposta (filtro de âncoras)	52
7.2	Algoritmos de Aprendizado de Máquina para localização	52
8	CRIAÇÃO DA BASE DE DADOS PARA FINGERPRINT	55

8.1	Pesquisa de base de dados para localização em ambientes internos utilizando fingerprint . . .	55
8.2	Escolha do ambiente	55
8.3	Definição do posicionamento das âncoras e pontos para medição	56
8.4	Suporte para medição	57
8.5	Procedimento da coleta	58
8.6	Algoritmos para coleta de dados	59
8.6.1	Algoritmo para envio de notificações pelas âncoras	60
8.6.2	Algoritmo para varredura do sinal das âncoras e envio dos dados de leitura	60
8.6.3	Servidor web para envio dos dados coletados pelo dispositivo móvel	60
8.6.4	Algoritmo para organização e criação da base de dados	61
8.7	Testes preliminares à criação da base de dados de fingerprint	62
9	MATERIAIS E MÉTODOS	65
9.1	Materiais	65
9.1.1	Dispositivo IoT	65
9.1.2	Hardware para organização dos dados e implementação de algoritmos	65
9.1.3	Ferramentas e linguagens de programação	66
9.2	Métodos	67
9.2.1	Métricas para avaliação	67
9.2.1.1	<i>Raiz quadrada do erro médio - RMSE</i>	67
9.2.1.2	<i>Função de distribuição acumulada - CDF</i>	67
9.2.2	Procedimento Experimental	68
9.2.2.1	<i>EXP-01: K-vizinhos mais próximos (KNN)</i>	68
9.2.2.2	<i>EXP-02: Árvore de regressão</i>	69
9.2.2.3	<i>EXP-03: Rede Neural Artificial (RNA)</i>	69
9.3	Comparação das abordagens	70
10	RESULTADOS E DISCUSSÃO	71
10.1	EXP-01: K-vizinhos próximos (KNN)	71
10.2	EXP-02: Árvore de regressão	72
10.3	EXP-03: Rede Neural Artificial (RNA)	72
10.4	Influência da altura	74
10.5	Comparativo das abordagens	75
11	CONCLUSÕES	79
11.1	Contribuições	80
11.2	Trabalhos futuros	80
	REFERÊNCIAS	83

1 INTRODUÇÃO

"The desert mocked the map-makers."

(Dean F. Wilson)

Hopebreaker

1.1 Contextualização

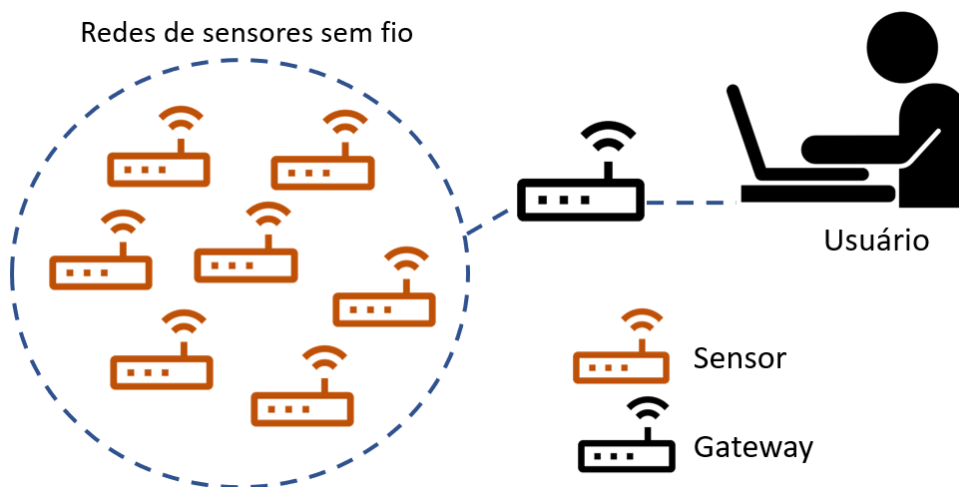
Um dos tópicos mais importantes na indústria de tecnologia nos últimos anos, a internet das coisas (do inglês *Internet of Things* - IoT) está incorporada em um amplo espectro de produtos, sensores, sistemas, com base na evolução constante de poder computacional, aliado a miniaturização de eletrônicos, oferecendo novas possibilidades de solução para demandas em diversos segmentos, desde a indústria, agricultura e governo (ROSE; ELDRIDGE; CHAPIN, 2015). O termo "*Internet of Things*" foi utilizado pela primeira vez em 1999 pelo britânico Kevin Ashton, pesquisador do Massachusetts Institute of Technology (MIT), para descrever uma aplicação para rastreabilidade de produtos na cadeia logística através do uso de etiquetas de identificação por radiofrequência (do inglês *Radio Frequency Identification* - RFID), e assim como nessa implementação, conceitualmente a expressão *Internet of Things* descreve cenários onde se reúne conectividade com a internet a dispositivos, sensores e itens do dia a dia para realização de tarefas de monitoramento e controle (ROSE; ELDRIDGE; CHAPIN, 2015).

Nesse sentido, dispositivos conectados são encontrados cada vez mais facilmente em aplicações do cotidiano, como termômetros, sensores industriais, relógios que medem batimentos cardíacos, entre outros, e a quantidade dos mesmos cresce exponencialmente, sendo que estudos recentes estimam que o número de dispositivos IoT pode chegar a 125 bilhões em 2030 (MARKIT, 2017). Essa popularização torna estes dispositivos cada vez mais baratos, acessíveis, e com consumos cada vez menores, permitindo anos de vida útil de bateria (LU et al., 2017).

Aplicações compostas por um grande número de dispositivos IoT são as denominadas redes de sensores sem fio (do inglês *Wireless Sensor Networks* - WSN), onde sensores com baixo processamento e capacidade de energia e armazenamento limitados, alimentados a bateria, se comunicam em redes de curto alcance entre si ou mesmo com uma estação de maior capacidade computacional, os gateways (SHAIKH; ZEADALLY, 2016). Os dispositivos sensores tem funções específicas de aquisição de dados como temperatura, umidade, condição de equipamentos, etc. Enquanto isso, com maior capacidade de processamento, os gateways também tem a tarefa de coletar informações dos sensores em campo e realizar o envio dos dados para algum local onde eles serão consumidos, por um usuário ou sistema de automação, o quem também exige que estes dispositivos possam transmitir pacotes utilizando mais de um protocolo de comunicação, um de baixo consumo e curto alcance para varredura dos sensores, e outro para direcionar a internet (SHAIKH; ZEADALLY, 2016). A arquitetura básica de uma rede de sensores sem fio pode ser visualizada na figura 1.

Frente a uma demanda crescente de consumo dos setores da indústria, governo, serviços, e mesmo dos consumidores finais, se investe cada vez mais no desenvolvimento, produção e comercialização de dispositivos IoT, o que causa uma presença massiva de dispositivos como smartphones, sensores inteligentes e redes Wi-Fi em praticamente qualquer tipo de ambiente fechado, o que permite uma ampla gama de aplicações, entre elas a localização em ambientes internos (ZAFARI; GKELIAS; LEUNG, 2019). Em locais externos, máquinas, carros e pessoas podem ter sua localização estimada utilizando o sistema de posicionamento global (do inglês *Global Positioning System* - GPS), entretanto, para ambientes internos e fechados, esse tipo de recurso para posicionamento é ineficaz, uma vez que a propagação do sinal de satélite pode ser facilmente interferida por coberturas, podendo não atingir o interior destes ambientes (DAI et al., 2019).

Figura 1 – Arquitetura básica de uma rede de sensores sem fio.



Fonte: Autor.

Nesse contexto, redes de sensores sem fio podem oferecer a infraestrutura e equipamentos para cumprir a tarefa de localização em ambientes internos, tema que tem sido amplamente investigada nas últimas décadas, para aplicações de localização diversas (ZAFARI; GKELIAS; LEUNG, 2019). Em ambientes hospitalares, aplicações de localização podem atender demandas como a localização de pacientes e das equipes médicas, auxiliar no rastreamento de equipamentos assistenciais, visibilidade e monitoramento do fluxo de pessoas nas dependências hospitalares, bem como a segurança de recém-nascidos (CALDERONI et al., 2015). Na indústria, a gestão e rastreabilidade de ativos sempre foi uma questão importante, devido a quantidade de equipamentos e materiais utilizados nos processos produtivos, muitas vezes de alto valor, que dependem frequentemente de registros e controles manuais, que podem trazer prejuízos financeiros, aumento no tempo de produção, e impacto nas metas da instituição (LEE et al., 2019).

As técnicas de localização baseadas em redes de sensores sem fio utilizam, a partir da comunicação entre esses dispositivos, métodos de medição de posição, distância e orientação, para dessa forma calcular uma localização estimada de um determinado objeto. Entretanto, esse tipo de implementação deve considerar diferentes equipamentos, tecnologias e algoritmos que irão impactar no custo de desenvolvimento, complexidade e acurácia (YASSIN et al., 2016).

Nesse sentido, este trabalho apresenta o detalhamento para a criação de uma base de dados para localização 3D em ambientes fechados, e implementação da proposta de **filtro de âncoras**, abordagem que seleciona dentre os pontos de referência detectados, constituídos por dispositivos sem fio fixos no ambiente (também chamados de âncoras), os que tem maior nível de sinal percebido. A implementação dessa abordagem tem o intuito de testar a hipótese de que, em um conjunto de âncoras detectadas, as que apresentam menor nível de sinal percebido tem menos influência como atributos de entrada de algoritmos de localização.

1.2 Motivação

Aplicações para localização em ambientes internos têm sido demandadas por diversos segmentos da sociedade, por questões de segurança, produtividade, entretenimento, entre outras (YASSIN et al., 2016). A execução desse tipo de empreitada exige abordagens e tecnologias diferentes com relação a disponibilidade de recursos e infraestrutura para que sejam viabilizadas. Certas tecnologias sem fio, como o *Bluetooth* estão presentes na maioria dos dispositivos

eletrônicos pessoais, aparelhos utilizados ambientes fechados, locais públicos ou privados, permitindo explorar o potencial destes equipamentos para aplicações de localização (GUPTA, 2016). Sendo assim, optou-se para a proposta deste trabalho o uso da tecnologia sem fio *Bluetooth Low Energy* (BLE) para localização em ambientes internos.

No contexto de Aprendizado de Máquina, bases de dados para treinamentos de modelos e algoritmos são essenciais, e a disponibilização destas bases permitem o estudo e evolução de toda uma comunidade de desenvolvedores. Repositórios como o da Universidade da Califórnia em Irvine disponibiliza e mantém centenas de bases de dados como um serviço para a comunidade de aprendizado de máquina, enquanto sites como o Kaggle proporcionam não somente bases de dados, mas um ambiente de máquina virtual para desenvolvimento de implementação de forma gratuita (KAGGLE, 2021; UCI, 2021). Bases de dados para aplicações de localização indoor também podem ser encontradas, em repositórios públicos ou próprios dos autores de trabalhos científicos, no entanto os disponibilizados são para um contexto 2D, não sendo encontradas bases para implementações com 3 dimensões, reduzindo a possibilidade de comparação de resultados entre diferentes pesquisadores. Nesse sentido, uma das partes deste trabalho será a criação e disponibilização de uma base de dados para localização 3D em um ambiente fechado.

Em ambientes internos, sistemas de localização com redes de sensores sem fio encontram diversos obstáculos para propagação dos sinais, como móveis e equipamentos, que podem provocar eventos de reflexão, refração, difração, espalhamento e multi percurso durante a transmissão entre os dispositivos (LU et al., 2010). Mesmo sem a presença destes objetos, espaços internos costumam ter menores dimensões, dificultando a comunicação entre dispositivos e âncoras, e dessa forma degradando a acurácia dos algoritmos de localização. Dessa forma, uma abordagem para tratamento dos atributos de entrada é necessária para obtenção de melhores resultados dos sistemas de localização em ambientes internos. Dessa maneira, este trabalho propõe uma abordagem para redução e seleção dos melhores parâmetros de entrada de algoritmos de localização em ambientes internos.

1.3 Objetivos

Considerando o contexto de IoT (*Internet of Things*), este trabalho tem os seguintes objetivos:

- Criação de uma base de dados de localização 3D em ambientes internos utilizando a tecnologia de *Bluetooth Low Energy* para utilização nesta dissertação e disponibilização para a comunidade científica;
- No contexto de algoritmos para localização em ambientes internos, a implementação de uma abordagem para seleção dos melhores e/ou mais representativos atributos de entrada, denominada **filtro de âncoras**, detalhamento dos testes para validação da hipótese levantada e discussão de resultados obtidos.

1.4 Organização do Trabalho

Esta dissertação está organizada da seguinte forma:

- O capítulo 1 traz uma breve contextualização do trabalho, salientando a motivação, objetivos do seu desenvolvimento, além das contribuições esperadas.
- No capítulo 2 as principais tecnologias de redes de sensores sem fio empregadas para sistemas de localização em ambientes internos serão apresentadas, sendo discutidos os benefícios e restrições no uso de cada uma, e finalmente uma comparação entre desempenho e recursos e infraestrutura necessárias.
- O capítulo 3 apresenta as principais técnicas para medição de distância baseadas em redes de sensores sem fio, com um detalhamento das equações envolvidas em cada um dos métodos.

- No capítulo 4 são descritos os principais métodos para localização em ambientes internos.
- O capítulo 5 é estruturado para apresentar os principais algoritmos de Aprendizado de Máquina empregados para aplicações de localização em ambientes internos.
- O capítulo 6 traz um levantamento de trabalhos correlatos ao tema de localização em ambientes internos com uso de redes de sensores sem fio.
- O capítulo 7 apresenta a proposta deste trabalho, com a discussão das principais justificativas para adoção da abordagem e algoritmo para a sua implementação.
- O capítulo 8 descreve o procedimento adotado para a criação da base de dados para implementação do trabalho e aplicação dos algoritmos de Aprendizado de Máquina. Os algoritmos e plataformas utilizadas são expostos detalhadamente.
- O capítulo 9 estrutura a metodologia, materiais e ferramentas utilizadas para esse trabalho, como os ambientes para testes, linguagens de programação e plataformas e experimentos propostos. Além disso as métricas para avaliação dos resultados também são apresentadas.
- No capítulo 10 é realizada a análise dos resultados comparativos, para validação da proposta deste trabalho, utilizando as métricas apresentadas anteriormente.
- O capítulo 11 traz as discussões finais e conclusões desta dissertação, além de propostas de trabalhos futuros.

2 Redes de sensores sem fio para localização em ambientes internos

É possível estimar a localização de um objeto em um ambiente interno através de implementações utilizando a comunicação entre dispositivos sem fio, no qual alguns equipamentos irão operar como pontos de referência com posição fixa, também chamados de âncoras, e um outro dispositivo terá sua localização estimada através de métricas baseadas na comunicação entre estes elementos (LAZIK et al., 2015). Em ambientes internos da indústria, varejo, saúde, alguns tipos de rede de comunicação sem fio como *Bluetooth* e Wi-Fi já estão amplamente presentes (KHALID; AKAN, 2016) e têm sido implantados para fins diversos como a localização em tempo real e inventário de itens (MOTRONI et al., 2018). Nas próximas seções serão abordadas as seguintes tecnologias: *Bluetooth* (e sua versão *Bluetooth Low Energy*), tecnologia sem fio escolhida para implementação deste trabalho, além de outras possíveis para aplicação de localização em ambientes internos, como *Zigbee*, Wi-Fi (padrão IEEE 802.11b), UWB (*Ultra Wideband*). Por fim uma breve comparação entre as tecnologias será apresentada com relação a acurácia e complexidade com base na literatura consultada.

2.1 Bluetooth

A tecnologia de comunicação sem fio *Bluetooth* tem como base o padrão IEEE 802.15.1 (normalizado pela organização *Institute of Electrical and Electronics Engineers*), sendo um protocolo de comunicação sem fio para curto alcance, utilizado para dispositivos de baixo consumo energético (TJENSVOLD, 2007). Esse padrão opera na banda de frequência não licenciada de 2.4 GHz, banda de rádio dedicada ao desenvolvimento de aplicações industriais, científicas e médicas (do inglês *Industrial, Scientific and Medical* - ISM), permitindo o suporte a aplicações e produtos como equipamentos médicos, fones de ouvido, sensores industriais, e outras aplicações diversas (SIG, 2021). Devido ao fato de outros protocolos de comunicação sem fio, como o Wi-Fi (IEEE 802.11b) e *Zigbee* (IEEE 802.15.4), também operarem na banda de 2.4 GHz, cada tecnologia utiliza um método de sinalização diferente de forma a evitar interferências (TJENSVOLD, 2007).

Na década de 90, a empresa Ericsson começou a investir no desenvolvimento de uma tecnologia de rádio de curto alcance que apoiaria a comunicação entre aparelhos celulares e fones de ouvido sem fio (SIG, 2021). Em parceria com IBM, Intel, Nokia e Toshiba foi formalizado então em 1998 um grupo para desenvolvimento, padronização e interoperabilidade da tecnologia, o *Bluetooth SIG*, do inglês *Special Interest Group* (SIG, 2021). A inspiração ao nome dado a essa tecnologia de comunicação tem origem em Harald Blatand, rei da Dinamarca entre os anos 940 a 985 d.C., considerado um dos principais responsáveis pela união entre povos nórdicos, em referência a união das multinacionais da região da Escandinávia envolvidas no projeto (MILLER; BISDIKIAN, 2001). *Bluetooth* é a tradução do nome do rei Blatand para o inglês, com explicações diversas e incertas sobre sua origem, como a que o apelido seria devido a sua pele morena e cabelos escuros, ou outras mais folclóricas, de que o rei era conhecido por seu gosto por mirtilos, que deixavam seus dentes com uma tonalidade azulada (MILLER; BISDIKIAN, 2001). O logotipo símbolo da tecnologia, inclusive, é inspirado nas iniciais de Harald Blatand, *ᚼ, em caracteres rúnicos (MILLER, 2001).

Para aplicações como transmissão de áudio de alta qualidade entre um smartphone e fones de ouvido, transferência de dados entre um tablet e dispositivo médico, ou telemetria em prédios inteligentes, o padrão *Bluetooth* disponibiliza atualmente dois principais tipos de rádio, sendo eles o *Bluetooth Low Energy* (BLE) e *Bluetooth Classic*, projetados

para atender às necessidades exclusivas de acordo com a demanda com relação a consumo de dados e energia (SIG, 2021). O padrão BLE, tecnologia escolhida para a proposta deste trabalho será descrita na próxima seção.

2.1.1 Bluetooth Low Energy

Um dos aprimoramentos da tecnologia *Bluetooth*, o padrão BLE (*Bluetooth Low Energy*) foi adicionado como parte da especificação do *Bluetooth* 4.0, destinado principalmente a dispositivos de baixíssimo consumo de energia, uma vez que dispositivos compatíveis com esse padrão podem operar por meses, ou até anos com baterias do tamanho de moedas ou menores, sem a necessidade de recarga ou troca (GUPTA, 2016). Também conhecido como *Bluetooth Smart* o padrão BLE é dedicado especificamente para aplicações que necessitem de baixo consumo, que exijam apenas transmissões esporádicas e com baixa taxa de transferência de dados (GUPTA, 2016). Exemplos de aplicações com BLE podem ser citados: detecção de proximidade e presença de dispositivos como chaves e celulares, comunicação com equipamentos médicos como termômetros e medidores de glicose, dispositivos vestíveis, sensores para agricultura, e outros mais (GUPTA, 2016). Devido à esta ampla presença em diversas aplicações do cotidiano, o *Bluetooth Low Energy* foi a tecnologia sem fio escolhida para implementação da proposta deste trabalho.

O rádio BLE pode suportar taxas de dados de 125 Kb/s a 2 Mb/s, embora por via de regra as taxas mais baixas sejam mais utilizadas, uma vez que o BLE não se aplica a aplicações como streamings de áudio, por exemplo (SIG, 2021). A faixa de potência para o padrão BLE, pode variar de 1mW a 100 mW, com consumo energético médio de menos de 5% em relação ao *Bluetooth Classic* (SIG, 2021)

Implementações de localização em ambientes internos com BLE tem atraído grande atenção devido não somente ao baixo consumo de energia dos rádios (SHAO; NIRJON; FRAHM, 2016) mas também a disponibilidade desse padrão em praticamente qualquer aparelho celular (XIAO et al., 2017). Aplicações com dispositivos utilizando BLE são empregadas na localização de pessoas dentro de supermercados, shoppings e hospitais (CHOOSAKSAKUNWIBOON et al., 2018) ou para o rastreamento de ativos dentro de armazéns utilizando dispositivos alimentados por bateria, dedicados a emissão de sinais BLE, os chamados *beacons* (LEE et al., 2019).

Uma característica específica do BLE que torna mais adequada para aplicações de localização é opção de topologia de rede de broadcast, arquitetura que permite que em ambientes internos, *beacons* sejam distribuídos de maneira fixa como âncoras, e a leitura do sinal proveniente de cada *beacon* possibilite o processamento de algoritmos de localização (VAŠČÁK; SAVKO, 2018).

2.2 Zigbee

A tecnologia sem fio *ZigBee* foi implementada com referência no IEEE 802.15.4, padrão proposto para abordar as necessidades de redes de área pessoal sem fio de baixa taxa de transmissão de dados (do inglês *Low Rate Wireless Personal Area Network* - LR-WPAN) operando na banda de frequência ISM de 2.4 GHz, com foco na habilitação de redes de sensores sem fio simples, implementações usualmente com restrições de baixo custo e baixo consumo de energia (RAMYA; SHANMUGARAJ; PRABAKARAN, 2011). Desenvolvido pela aliança *ZigBee*, que tem como membros centenas de empresas fabricantes de semicondutores e desenvolvedores de software, as razões sugeridas para uso dessa tecnologia são: confiabilidade, fácil implantação, baixo custo e consumo energético (permitindo equipamentos alimentados a bateria), suporte a grande números de nós, licenças de uso aberta ou de preços insignificantes (RAMYA; SHANMUGARAJ; PRABAKARAN, 2011).

As características anteriormente citadas tornam a tecnologia *ZigBee* adequada para diversos tipos de aplicação em *smart cities*, indústria, agricultura, e outras (OLIVEIRA et al., 2017). Explorando principalmente a vantagem de baixo consumo e longo alcance, que permite dispositivos móveis alimentados a bateria em grandes ambientes internos e

fechados, aplicações de localização tem utilizado a tecnologia *ZigBee* em sua arquitetura ([SADOWSKI; SPACHOS, 2018](#)).

2.3 WiFi (IEEE 802.11b)

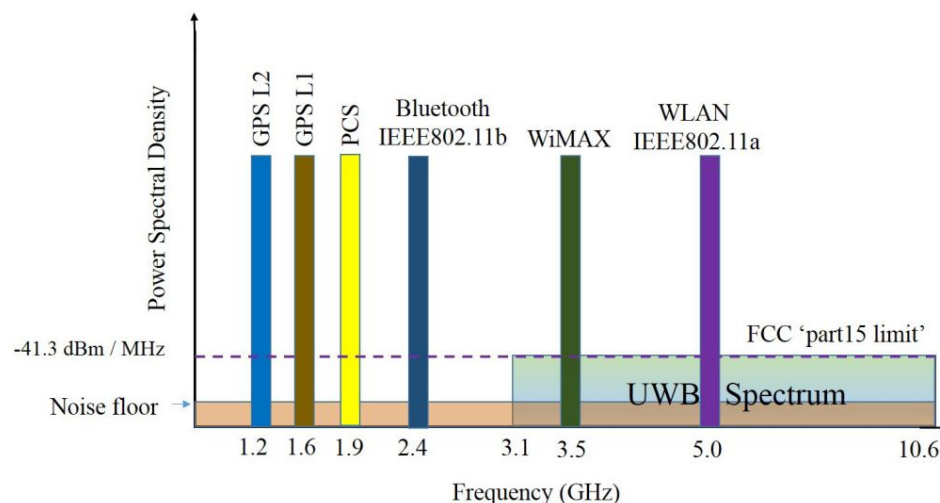
Desde 1990, com a introdução do padrão IEEE 802.11 de redes de área local sem fio (do inglês *Wireless Local Area Network* - WLAN) verificou-se um crescimento exponencial do mercado de dispositivos para computação móvel e acesso à Internet, impactando em áreas públicas, escritórios e residências ([HIERTZ et al., 2010](#)). O padrão de rede Wi-Fi, que opera na banda de 2.4 GHz, está amplamente presente em aplicações na indústria, *hotspots* em locais públicos, corporativos e residenciais, devido a sua adoção em equipamentos como impressoras, *smartphones*, computadores pessoais, telefones VoIP, televisores, e muitos outros dispositivos que incorporam a tecnologia de rede de área local sem fio. ([LIU et al., 2007](#); [SALMAN et al., 2018](#)).

Para o cenário de localização em ambientes internos e IoT no entanto, dispositivos Wi-Fi tem desvantagem com relação a outras tecnologias em relação a potência necessária, consumindo para tarefas de transmissão e recepção cerca de 7 vezes mais potência em relação ao *ZigBee*, e um tendo um consumo 10 vezes mais potência em relação ao *Bluetooth Low Energy* ([KAZEEM; AKINTADE; KEHINDE, 2017](#)). Em função da menor eficiência energética, e a acurácia de localização semelhante, o padrão Wi-Fi tem se tornado menos interessante do que outras tecnologias *wireless* para aplicações de localização em ambientes internos ([ZAFARI; GKELIAS; LEUNG, 2019](#)).

2.4 UWB

Enquanto tecnologias como *Bluetooth*, *ZigBee*, Wi-Fi e outras utilizam porções de bandas de frequência ISM, faixas de trabalho disputadas, o amplo espectro utilizado pelo UWB (*Ultra Wideband*), de 3.1 à 10.6 GHz (vide figura 2), permite uma melhor performance de comunicação imune à colisões e interferências ([ABINADER et al., 2018](#); [ZAFARI; GKELIAS; LEUNG, 2019](#)). A título de exemplo, um produto desenvolvido com essa tecnologia para aplicações de localização interna, o Dart UWB da fabricante Zebra, utiliza um intervalo de frequência de 6.35 GHz a 6.75 GHz ([ZEBRA, 2021](#)).

Figura 2 – Espectro de frequência de algumas tecnologias *wireless*



Fonte: ([DODDIPALLI; KOTHARI, 2018](#))

Em aplicações com UWB, pulsos ultracurtos com duração de menos de 1 nanossegundo são transmitidos operando em uma grande largura de banda, maiores do que 500 MHz, dentro do espectro de 3.1 to 10.6 GHz, utilizando técnicas de medição de distância baseadas em tempo, conforme detalhado na seção 3.2, menos afetado por interferências e condições de falta de linha de visada direta (do inglês *Non Line of Sight* - NLOS), por ser capaz de penetrar em certos tipos de materiais, além de utilizar um ciclo de trabalho muito baixo que resulta em consumo de energia reduzido (ZAFARI; GKELIAS; LEUNG, 2019).

Pelos motivos citados anteriormente, essa tecnologia tem sido particularmente atraente para localização em ambientes internos, com este modelo podendo chegar a um acurácia de posicionamento de 30cm (RIDOLFI et al., 2016), entretanto o uso do UWB exige uma infraestrutura específica e rede proprietária, e a aplicação de métodos de cálculo de distância baseados em tempo dependem de sincronização estrita entre transmissores e receptores (ZAFARI; GKELIAS; LEUNG, 2019).

2.5 Comparação de performance de localização entre redes de sensores sem fio

A comparação de redes de sensores sem fio aplicadas a localização em ambientes internos pode ter diferentes inferências e conclusões de acordo com a ótica de implementação, algoritmos utilizados, hardware disponível, etc. Entretanto, é possível citar algumas características básicas com relação a viabilidade e benefícios da aplicação de cada uma destas tecnologias, entre elas podem ser citadas: acurácia (que pode variar para cada tecnologia de acordo com a bibliografia), consumo de energia, vantagens e alguns inconvenientes de cada uma das tecnologias. Com base nos trabalhos de (LASHKARI et al., 2018; ZHOU et al., 2018) foi possível coletar algumas dessas informações e organizar a tabela 1, e o gráfico na figura 3.

Das quatro tecnologias de sensores sem fio apresentadas a UWB se mostra como de maior complexidade, isso devido a necessidade de uma infraestrutura proprietária, por não ser uma rede disponível de forma ampla em outros tipos de aplicações cotidianas, e como consequência tem uma baixa variedade de produtos de prateleira, ocasionando um alto custo (ZAFARI; GKELIAS; LEUNG, 2019). No entanto aplicações com UWB encontram resultados de acurácia submétricos, devido a sua imunidade a interferência e larga banda de frequência para atuação (LASHKARI et al., 2018). A tecnologia *Zigbee*, embora também dependente de uma infraestrutura e hardware específico ainda pouco difundida em ambientes internos, tem sido aplicado na área de sensores para a agricultura, com uma arquitetura de redes em malha sem fio, proporcionando dispositivos de baixo consumo e mais acessíveis (OLIVEIRA et al., 2017).

Tabela 1 – Comparação de tecnologias de localização em ambientes internos com redes de sensores sem fio.

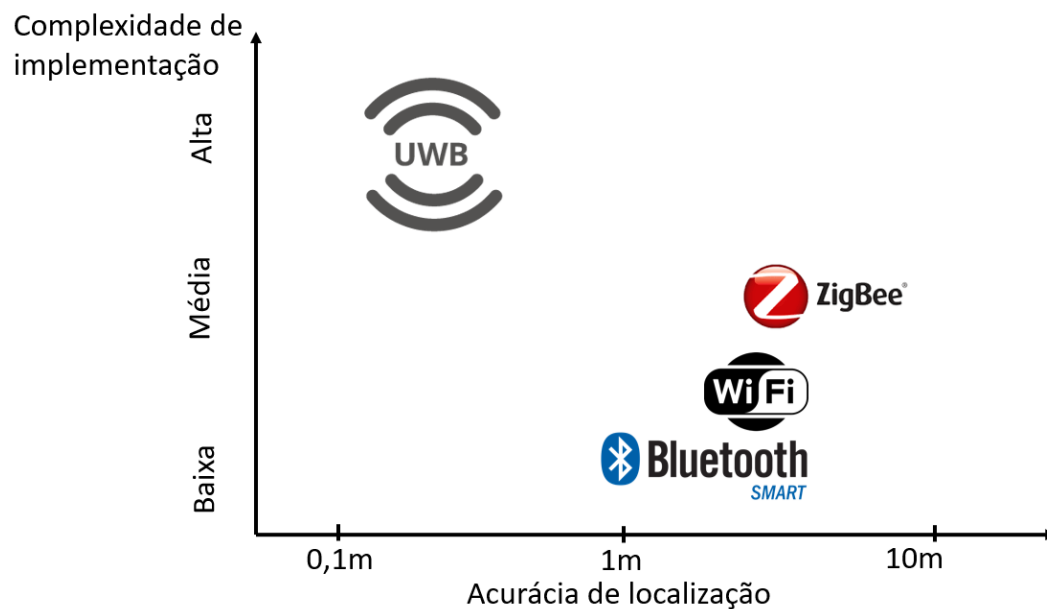
Tecnologia	UWB	Zigbee	Wi-Fi	Bluetooth Low Energy
Consumo de energia	Moderado	Baixo	Moderado	Muito Baixo
Acurácia	30cm-1m	1-10m	1-7m	1-5m
Vantagens	Imune a interferências	Baixo consumo	Ampla disponibilidade	Baixo consumo, Ampla disponibilidade
Desvantagens	Alto custo, Necessário infraestrutura proprietária	Necessário infraestrutura proprietária	Propenso a interferência e ruído	Propenso a interferência e ruído

Fonte: Autor.

Em contrapartida, a ampla presença de infraestrutura Wi-Fi em ambientes fechados públicos, residenciais e corporativos faz com este padrão seja provavelmente o mais acessível para desenvolvimento de sistemas de localização, entretanto com resultados de acurácia próximos ou inferiores às outras tecnologias que compartilham as faixas de frequência ISM (*Industrial, Scientific and Medical*) (LASHKARI et al., 2018). Assim como o Wi-Fi, implementações de localização em ambientes internos com *Bluetooth Low Energy* podem ser consideradas de baixa complexidade devido a disponibilidade desse tipo de comunicação em dispositivos pessoais como tablets, celular e computadores pessoais, com uma variedade de softwares e aplicativos já desenvolvidos que podem servir como base para novas aplicações (GUPTA, 2016).

Por fim, a tecnologia UWB para aplicações de localização, única a alcançar resultados sub-métricos de acurácia, ainda encontra obstáculos na adoção da solução para uma variedade de ambientes devido ao custo e complexidade de implantação de tecnologia e pouca difusão em dispositivos portáteis (LASHKARI et al., 2018).

Figura 3 – Comparação de complexidade de hardware x acuracidade de localização.



Fonte: Autoria própria com base em (LASHKARI et al., 2018; ZHOU et al., 2018)

3 Técnicas de medição de distância baseadas em redes de sensores sem fio

Aplicações de localização dependem de métodos para estimar a distância entre os dispositivos fixos com posições previamente conhecidas (âncoras) e os dispositivos que se pretende localizar (YASSIN et al., 2016). A utilização de redes de sensores sem fio para esse tipo de aplicação permite utilizar parâmetros da comunicação entre os dispositivos, os quais são disponibilizados por fabricantes de chips para outras necessidades, como verificação de potência ou qualidade de sinal, latência e outros. Neste capítulo serão apresentadas algumas das técnicas para medição de distância baseadas em redes de sensores sem fio empregadas para aplicações de localização, mais especificamente as possíveis para redes de curto alcance, utilizadas em ambientes internos.

3.1 Indicador de potência do sinal recebido - RSSI

O conceito do índice de potência de sinal recebido (do inglês *Received Signal Strength Indicator* - RSSI) descreve a relação entre sinais de dispositivos sem fio, referente a potência transmitida por determinado dispositivo e o nível de potência detectada pelo receptor. O RSSI geralmente é medido em decibéis-miliWatts (dBm) ou miliWatts (mW), e fabricantes de semicondutores com suporte a conectividade disponibilizam comandos para obtenção de dados de RSSI associado ao pacotes recebidos de outros dispositivos (GUPTA, 2016). Para aplicações de localização, o RSSI pode ser usado para estimar a distância entre um dispositivo transmissor (Tx) e um receptor (Rx), utilizando o modelo o seu modelo de perda de sinal, que estima a degradação na intensidade de um sinal emitido a partir de uma antena e propagado através de um espaço até o alvo receptor, apresentado na equação 1:

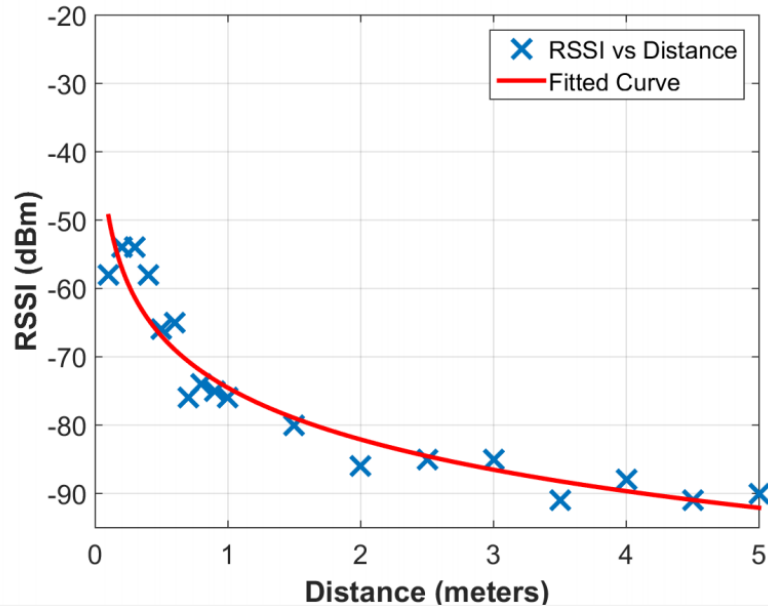
$$RSSI = -10n\log_{10}(d) + C \quad (1)$$

No modelo de perda de sinal, n é o parâmetro de perda de intensidade de sinal na trajetória relacionado ao ambiente específico de transmissão sem fio, d é a distância entre os dispositivos de transmissão e recepção, e C é uma constante fixa de perdas do sistema (SADOWSKI; SPACHOS, 2018). Dessa forma, quanto maior o valor de RSSI, menor é a distância entre Tx e Rx, conforme pode ser observado na figura 4, referente ao modelo de perda de sinal de um dispositivo *Bluetooth Low Energy*, um *beacon* Gimbal Series 10, utilizado em um ambiente interno no trabalho de (SADOWSKI; SPACHOS, 2018), no gráfico as marcações de "X" em azul representam medições em laboratório do nível de RSSI medido em relação a distância e em vermelho se observa a curva ajustada.

O indicador RSSI é uma medida relativa de potência que depende de características específicas de cada fornecedor de chip, como potência de saída da antena, sensibilidade de recepção, entre outras especificações. O rádio *Bluetooth Low Energy* do chipset ESP32-WROOM-32 da fabricante Espressif, por exemplo, tem um intervalo de medição de potência de sinal que pode ir de 0 até -97 dBm, enquanto no rádio *Zigbee* CC2420 da Texas Instruments a variação de detecção de RSSI está entre 10 e -95 dBm (TI, 2021; ESPRESSIF, 2021). Dessa forma a aplicação de técnicas de medição de distância baseadas em RSSI depende não somente do respectivo modelo de perda de sinal, mas que também sejam observadas as especificações do dispositivo utilizado com relação ao intervalo de detecção de potência recebida.

O comportamento logarítmico decrescente do modelos de RSSI x distância física pode influenciar na degradação da acurácia conforme o espaço entre transmissor e receptor aumenta, essa relação pode afetar a performance de algoritmos de localização (XUE et al., 2018), o que exige abordagens que considerem esse comportamento, deterioração devido

Figura 4 – Modelo de perda de sinal por distância (path loss) para um dispositivo BLE.



Fonte: (SADOWSKI; SPACHOS, 2018)

a obstáculos, e interferência e flutuação no sinal (XIAO et al., 2017). Devido a simplicidade de implementação de algoritmos utilizando RSSI, e presença em praticamente todos tipos dispositivos IoT, essa medidas têm sido utilizada em diversas aplicações localização e proximidade (BIANCHI; CIAMPOLINI; MUNARI, 2018). Este trabalho irá utilizar o RSSI como técnica de medição de distância, baseado na tecnologia de comunicação sem fio BLE (*Bluetooth Low Energy*).

3.2 Técnicas de medição de distância baseadas em tempo

Técnicas de medição de distância baseadas em tempo medem o intervalo da transmissão até a recepção de um sinal para estimar espaço entre dois dispositivos de comunicação sem fio (BABA, 2011). Diferentes técnicas podem ser desenvolvidas utilizando essa métrica, uma delas é o de tempo de voo (do inglês *Time of Flight* - TOF), também denominada de *Two-Way Ranging* TOF, nesse método um dispositivo A envia, através de comunicação sem fio, um pacote de dados (com sua identificação ou pseudo-ruído qualquer) para um segundo dispositivo B, que ao receber, responde de volta com outro pacote depois de um intervalo de tempo t_{reply} (referente ao suposto período processamento desse dispositivo), o dispositivo A ao detectar a resposta, calcula o período desde o início do processo, o chamado intervalo de *round-trip time* t_{rtt} , conforme observado na figura 5 (MALETIC et al., 2019).

Subtraindo o tempo de resposta t_{reply} do tempo total t_{rtt} , resta apenas o tempo de envio e de retorno do sinal, que aqui se assume que esses dois intervalos são iguais, logo se divide essa diferença por 2, e assim se obtém o tempo de voo (TOF) em segundos, vide equação 2. A distância é então calculada multiplicando c (velocidade da luz $\simeq 3 \times 10^8$ m/s) pelo TOF obtido, chegando na distância estimada em metros em os dois dispositivos (BABA, 2011).

$$TOF = \frac{t_{rtt} - t_{reply}}{2}, \quad d = c \cdot TOF \quad (2)$$

A técnica de tempo de chegada (do inglês *Time of Arrival* - TOA), mais simples do que a *Two-Way Ranging* TOF, depende da identificação do intervalo em segundos entre o momento de transmissão de um sinal t_1 e o instante de

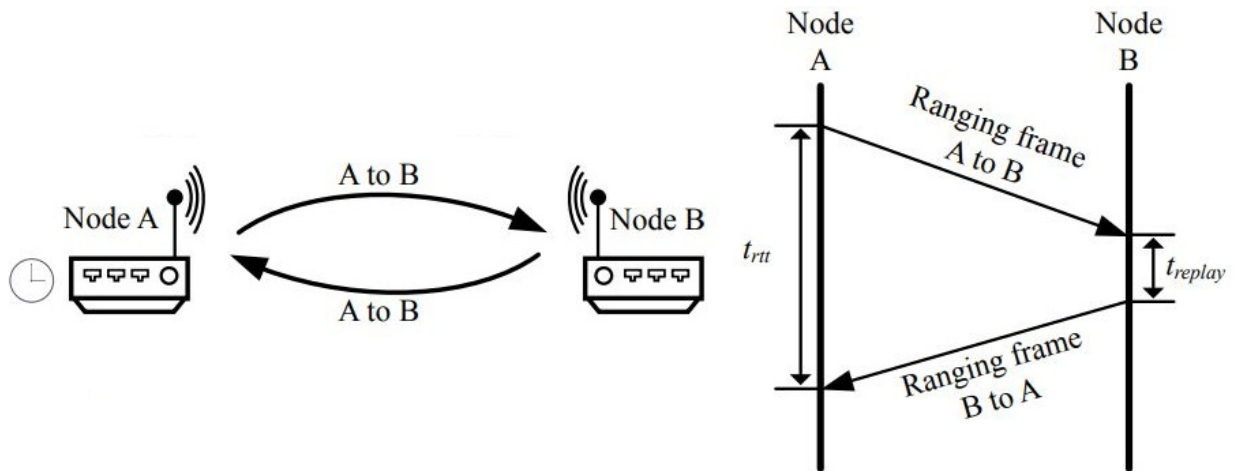
chegada t_2 do mesmo, permitindo estimar a distância em metros entre os dois dispositivos (vide equação 3), em que c é a velocidade da luz (BENSKY, 2019). Essa técnica exige, no entanto, uma ótima sincronização de tempo entre os nós, dados que os instantes de envio e de chegada serão detectados por dispositivos diferentes.

$$d = (t_2 - t_1) \cdot c \quad (3)$$

Outras técnicas podem explorar diferentes processos para estimar a distância, o método de diferença de tempo de chegada (do inglês *Time Difference of Arrival* - TDOA), explora a diferença nos tempos de propagação dos sinais de diferentes transmissores, ou seja, depende do cálculo do tempo de percurso a partir de um dispositivo e sua aquisição por mais de um receptor, diferentemente da técnica TOF, por exemplo, onde é necessário apenas o tempo de propagação do sinal absoluto entre 2 dispositivos.

A utilização de técnicas para medição de distância baseadas em tempo depende criticamente da precisão dos dispositivos em detectar os instantes de envio e recepção dos pacotes, considerando que, se um sinal sem fio propaga-se na velocidade da luz, uma imprecisão de microssegundos de detecção podem causar erros de medição de centenas de metros (BABA, 2011).

Figura 5 – Esquema da técnica de Time of Flight com Two Way Range.



Fonte: (MALETIC et al., 2019)

3.3 Técnicas de medição de distância baseadas em recursos específicos de hardware

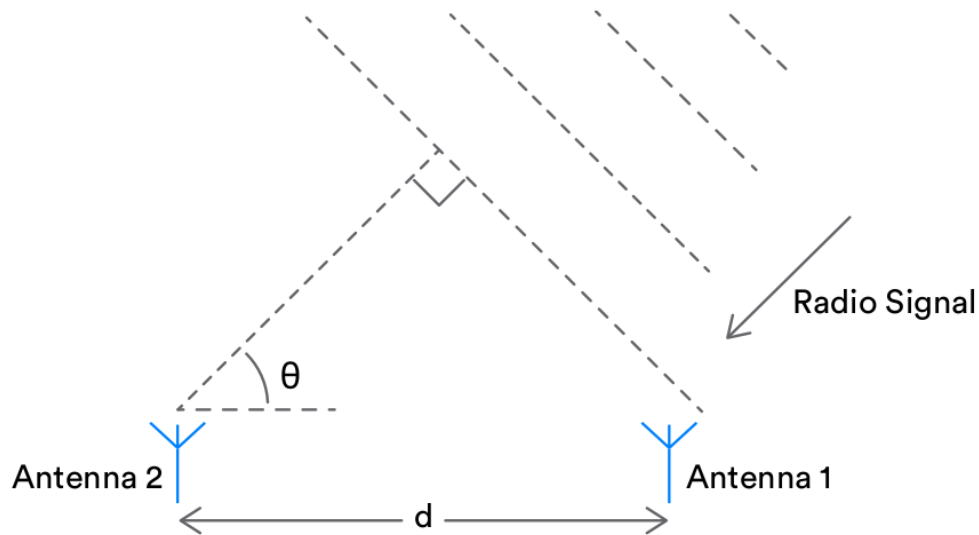
Algumas técnicas para medição de distância com redes de sensores sem fio dependem de recursos específicos de hardware, como arranjos de antenas para recepção de sinal. Uma destas técnicas é a de ângulo de chegada (do inglês *Angle of Arrival* - AOA), utilizada para medição de posicionamento a partir do cálculo da direção de um sinal recebido, através da estimativa do seu ângulo em que o sinal atingiu o arranjo de antenas. Para essa abordagem é necessário um arranjo de antenas direcionais, possibilitando calcular o ângulo em que o sinal transmitido colide com o receptor (SCHÜSSEL, 2016). Basicamente, é possível calcular o ângulo de chegada de um sinal, se tivermos as seguintes informações: distância entre as antenas no arranjo, diferença de fase entre o mesmo sinal chegando nas duas antenas, e o comprimento de onda do sinal (WOOLLEY, 2020).

De posse destas informações é possível calcular o ângulo do sinal utilizando a equação 4:

$$\theta = \arcsen\left(\frac{\psi \cdot \gamma}{-2\pi \cdot d}\right) \quad (4)$$

onde θ representa o ângulo de chegada do sinal, ψ é a diferença de fase do sinal recebido entre as antenas 1 e 2, γ é o comprimento de onda, e d a distância entre as duas antenas. A figura 6 demonstra a ideia de como um sinal de entrada pode ter seu ângulo de chegada calculado a partir de duas antenas. A técnica de ângulo de chegada permite aplicações para detectar a direção de objetos, através da utilização de dispositivos IoT, e com a aplicações de localização, implementando algoritmos como o de triangulação, conforme será melhor detalhado na seção 4.3.

Figura 6 – Sinal de entrada atingindo um conjunto de antenas e medição de ângulo de chegada.



Fonte: (WOOLLEY, 2020).

4 Técnicas de localização para ambientes internos

A partir dos tipos de redes de sensores sem fio apresentados, métodos para estimar distância e ângulo entre esses dispositivos *wireless*, algumas técnicas e algoritmos podem ser utilizados para inferir a localização de um objeto. Nesse sentido, alguns dos métodos para localização serão descritos nessa seção, entre elas: proximidade, trilateração, triangulação e *fingerprint*, sendo esta última a técnica estado da arte para implementações de localização em ambientes internos com dispositivos IoT, e a abordagem utilizada na proposta deste trabalho.

4.1 Proximidade

Um dos métodos mais básicos para localização em ambientes fechados é a técnica da proximidade, que se baseia no seguinte conceito: se o dispositivo que se pretende localizar está dentro do alcance de comunicação, ou seja, que possa enviar ou receber dados de uma determinada estação ou âncora, de localização previamente conhecida, logo o dispositivo pode ter seu posicionamento aproximado ou associado ao local onde este ponto de acesso (a âncora) que o identificou foi fixado (LIU, 2014). Este tipo de técnica não exige cálculos específicos para estimar a localização devido a sua simplicidade, necessitando apenas vincular o dispositivo a ser localizado com a posição da âncora conhecida.

4.2 Trilateração

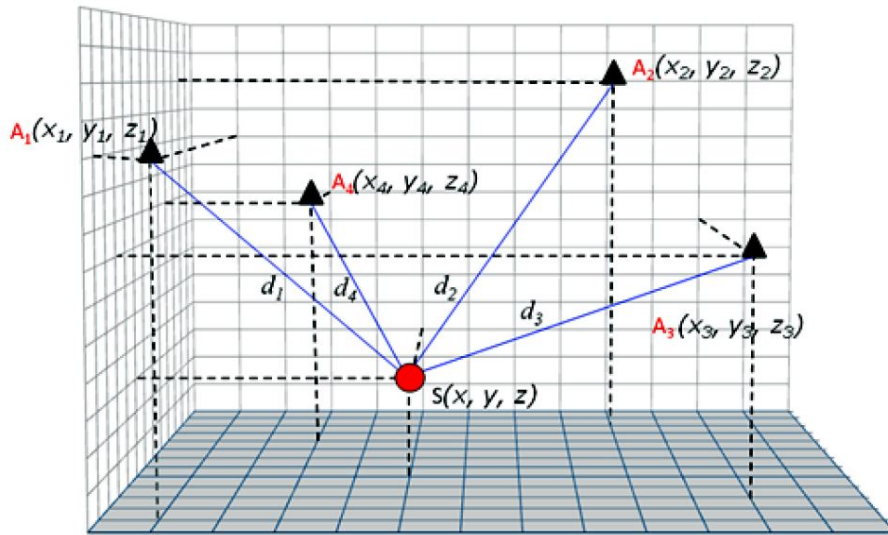
A trilateração é uma técnica utilizada para estimar as coordenadas de um determinado objeto, a partir de dados de elementos emissores fixos com posição já previamente conhecida (BLAS; IPIÑA, 2017). O processo de trilateração pode ser descrito basicamente como o problema de encontrar a intersecção de três esferas, no centro de cada uma das esferas há uma estação ou elemento emissor A_i , de posição $pi = (x_i, y_i, z_i)$, e di é a medição de distância entre cada estação i e a intersecção (THOMAS; ROS, 2005). De posse das informações de pi , e di de cada uma das estações em relação a intersecção das esferas, a localização desconhecida é encontrada realizando para o sistema de equações quadráticas descrito em (5).

$$\begin{aligned} (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 &= d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 &= d_2^2 \\ (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 &= d_3^2 \\ (x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2 &= d_4^2 \end{aligned} \tag{5}$$

Na figura 7, se observa um objeto de localização incógnita S , os segmentos em azul correspondem às medidas de distância estimadas por algum método de medição de distância conforme citado no capítulo 3 entre o objeto e as âncoras A_i . Para localizar um objeto em um espaço bidimensional, são necessárias três estações fixas (ou âncoras), assim como para um espaço 3D são necessárias no mínimo 4 âncoras (SHIH; MARRÓN, 2010).

O exemplo mais famoso e comum hoje em dia de aplicação com trilateração é o do GPS (*Global Positioning System*), que estima uma localização desconhecida a partir do cálculo do intervalo que um sinal leva ao ser enviado de um satélite até um receptor em terra, utilizando métodos de alcance baseados em tempo como a TOA (*Time of Arrival*) (DJUKNIC; RICHTON, 2001). Em teoria o sistema GPS necessita que este dispositivo receba sinais de pelo menos 3 satélites, mas para melhor acurácia e dependendo da localidade, é possível receber o sinal de dezenas deles (DJUKNIC; RICHTON, 2001). Para ambientes internos a trilateração pode utilizar outros métodos de alcance, como por exemplo o modelo de perda de sinal, que estima como o sinal de um emissor irá se degradar em função da distância.

Figura 7 – Conceito fundamental da técnica de trilatação 3D.



Fonte: (SHIH; MARRÓN, 2010).

No entanto o método de trilatação tem um melhor desempenho em condições de linha de visada direta (do inglês *Line of Sight* - LOS) pois nestas circunstâncias as métricas de medição de distância tendem a ter melhor acurácia, enquanto na presença de obstáculos essa técnica de localização pode ter desempenhos inferiores e consequentemente maiores erros.

4.3 Triangulação

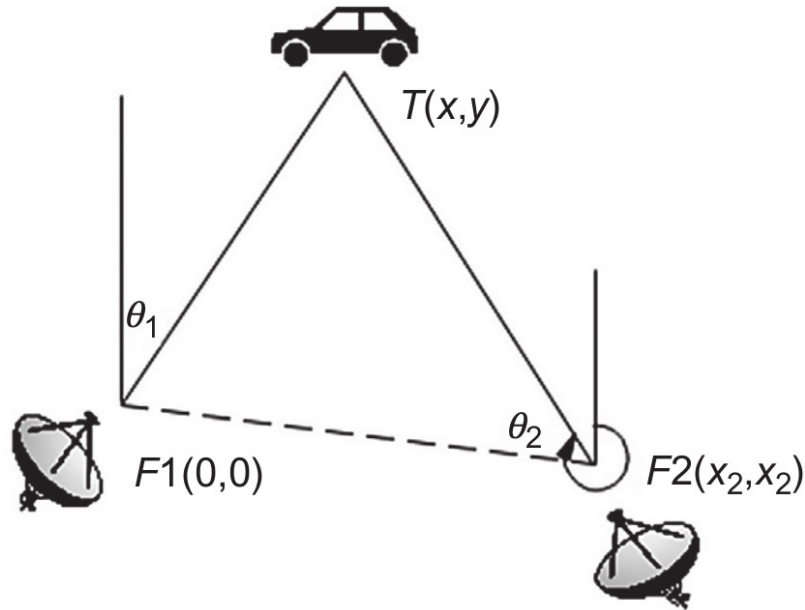
Outra técnica de localização é a triangulação, que utiliza as propriedades geométricas de triângulos para estimar o posicionamento de um objeto (LEE et al., 2014). Este princípio consiste em no mínimo em três pontos, sendo dois pontos de referência ($F1$ e $F2$), com localização previamente conhecida, e um ponto-alvo que se deseja estimar sua posição (T), como é ilustrado na figura 8. A técnica de triangulação depende de métodos como o de medição de ângulo de chegada de um sinal (AOA, conforme citado na seção 3.3), de modo que, possuindo a informação desse ângulo em relação aos pontos de referência a localização do ponto alvo pode ser estimada através da equação 6 (BENSKY, 2019).

$$y = \frac{y_2 \cdot \tan(\theta_2) - x_2}{\tan(\theta_2) - \tan(\theta_1)}, \quad x = y \cdot \tan(\theta_1) \quad (6)$$

4.4 Fingerprint

A técnica de *fingerprint* utiliza uma infraestrutura de redes sem fio em áreas internas ou outros locais onde o GPS não funciona corretamente, podendo inclusive oferecer serviços de valor agregado para redes *wireless* já existentes (KHALEL, 2010). Essa técnica de localização também é conhecida como análise de cena, dado que ela requer uma pesquisa/*site survey* do ambiente onde o sistema de posicionamento será processado (ZAFARI; GKELIAS; LEUNG, 2019). O *site survey* consiste na criação prévia de uma base de dados fundamentada em alguma métrica para cálculo de distância ou ângulo (apresentadas no capítulo 3) no ambiente interno onde se deseja que o processo de localização seja feito.

Figura 8 – Conceito fundamental da técnica de triangulação.



Fonte: (BENSKY, 2019).

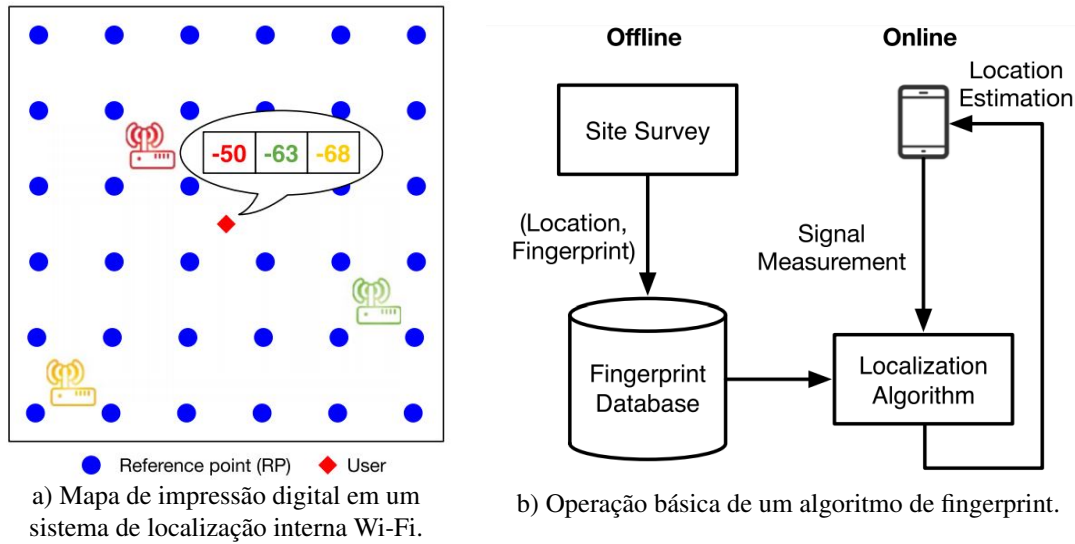
No contexto de Aprendizado de Máquina supervisionado, uma base de dados tem atributos, que servem como entrada para os algoritmos, e rótulos, que são dados que classificam cada conjunto de entradas. Para uma base de dados de *fingerprint* os atributos de entradas são dados detectados, como o nível de potência sinal recebido das âncoras (RSSI) ou o tempo de chegada do sinal (TOA) por exemplo, e os rótulos são as localizações referentes a cada ponto de coleta (JIANG et al., 2016). Neste sentido, é possível dividir o processo de análise de cena em 2 fases: Fase de treinamento, também chamada de *offline*, que se inicia com a criação da base de dados e posteriormente o treinamento de algoritmos de Aprendizado de Máquina que serão utilizados, e a fase de testes (*online*), onde o modelo é colocado em prática para estimativa de localização. As duas fases serão detalhados com mais profundidade a seguir.

4.4.1 Treinamento

Na primeira parte do processo, a fase *offline*, o ambiente de interesse (onde deve ser estimada a localização de algum elemento) é dividida com a ajuda de uma planta baixa ou mapa do local em partes ou células menores, que serão utilizados como pontos de referência (JIANG et al., 2016). O número total de células pode ser representado por N , e cada ponto n tem uma posição conhecida p_n , representada em um cenário 3D como $p_n = [x_n, y_n, z_n]$. O número de células escolhido influencia na granularidade das informações de localização colhidas, no tamanho da base de dados e no modelo de algoritmo de Aprendizado de Máquina que será treinado. A divisão de células no ambiente é feita, quando possível, no formato de um *grid* ou rede, de modo a obter uma distribuição regular dos dados.

A partir da tecnologia e métrica escolhidas é realizado o processo de coleta dos dados de entrada, detectados por um elemento receptor (intensidade do sinal recebido, por exemplo), referente a um número de âncoras M , em que, para cada posição p_n é coletado um vetor f_n com dados de medição r para cada âncora m detectada, formando para cada ponto o registro $f_n = [r_{1n}, r_{2n}, \dots, r_{mn}]$ (XIAO et al., 2017). Dessa forma, a base de dados *fingerprint* F pode ser expressada como um conjunto de tuplas f_n e p_n para as N células escolhidas:

Figura 9 – Mapa e operação básica de algoritmo baseado em fingerprint.



Fonte: (HE; CHAN, 2015)

$$F = \{(f_1, p_1), (f_2, p_2), \dots, (f_n, p_n)\}$$

Conforme explicado anteriormente, a base de dados servirá para treinar algum tipo de algoritmo de Aprendizado de Máquina supervisionado, ela é composta basicamente por dados de entrada e rótulos. Uma representação visual dessa metodologia pode ser observada na figura 9a) em que os pontos azuis representam cada ponto de referência onde os dados foram coletados durante a criação da base, referente às 3 âncoras fixadas no ambiente.

A partir da base de dados de *fingerprint*, métodos estatísticos e de Aprendizado de Máquina supervisionado serão então utilizados para criação de modelos que, a partir dos dados de entrada, seja possível extrapolar e estimar a localização do elemento móvel. Alguns modelos usados são: Naive Bayes, K-vizinhos mais próximos (do inglês *K-Nearest Neighbors* - KNN), redes neurais, entre outros, os quais serão melhor abordados no capítulo 5.

A fase de treinamento procura adquirir uma quantidade satisfatória de medições para que se tenha uma massa de dados com informações suficientes que caracterizem como a métrica escolhida (RSSI, TOA, etc.) se comporta no ambiente (PECORARO et al., 2018). Para explorar resultados diferentes, a etapa de treinamento do *fingerprint* pode também optar por, quando possível, coletar mais de uma métrica de medição referente às âncoras, ou mesmo dados de mais sensores como informações de entrada dos algoritmos de localização, principalmente em cenários onde o elemento receptor que será localizado pode se movimentar durante a etapa *online* (BELMONTE-HERNÁNDEZ et al., 2019).

4.4.2 Teste

Na segunda fase, considerada *online*, com a base de dados F construída e o modelo de aprendizado treinado, o elemento receptor (que se deseja localizar) irá realizar novas leituras baseados na métrica escolhida em uma localização desconhecida p_i , formando o vetor f_i com dados detectados das M âncoras, $f_i = [r_{1i}, r_{2i}, \dots, r_{mi}]$. Coletado o vetor f_i será então utilizado o algoritmo treinado para inferir sua localização desconhecida $p_i = [x_i, y_i, z_i]$. É possível observar na figura 9a) um exemplo de localização durante a fase de teste, onde, em um ponto de localização desconhecida em

vermelho, é realizada uma leitura referente ao nível de sinal detectado das âncoras, que será utilizada para inferir a localização incógnita. Na figura 9b) o fluxo básico de *fingerprint* é exibido, detalhando as fases *offline* e *online*.

5 Algoritmos de aprendizado de máquina para localização interna

Entre as abordagens para sistemas de localização em ambientes internos, as baseadas em algoritmos de Aprendizado de Máquina têm tido sucesso em sua implementação devido a sua capacidade de estimar o posicionamento através do uso de dispositivos sem fio mesmo em condições de falta de linha de visada direta (NLOS), multipercursos (múltiplas ondas de rádio oriundas do mesmo transmissor chegando em momentos diferentes) e flutuação de sinal, sem afetar drasticamente a acurácia, o que acontece com algoritmos de localização puramente baseadas nas técnicas de medição, como trileração e triangulação (ROY; CHOWDHURY, 2021).

Algoritmos de Aprendizado de Máquina dependem inicialmente de uma base de dados com informações suficientes para que um modelo seja treinado, e possa proporcionar resultados de acurácia aceitáveis. Nesse sentido, a aplicação destes algoritmos depende de um estágio de coleta de dados, no caso de aplicação de localização, das métricas utilizadas para medição, explicadas no capítulo 3, associadas a posição da coleta, primeira etapa para o processo de *fingerprint*, conforme descrito na seção 4.4.1. Este capítulo irá abordar as principais técnicas de Aprendizado de Máquina utilizadas para aplicações de localização com base de dados de *fingerprint*, em especial para ambientes internos. Dentre as técnicas de Aprendizado de Máquina apresentadas aqui 3 serão utilizadas para as implementações deste trabalho: K-vizinhos mais próximos (KNN), árvore de regressão e redes neurais artificiais.

5.1 K-vizinhos mais próximos - (KNN)

O algoritmo de K-vizinhos mais próximos (do inglês *K-Nearest Neighbors* - KNN) é um tipo de aprendizado baseado na comparação de instâncias para resolver problemas de classificação e regressão. No KNN, uma amostra é classificada pela sua similaridade com seus k vizinhos mais próximos, sendo atribuída a ela a mesma classe dessas instâncias vizinhas, ou no caso de um problema de regressão, a média dos valores dos k vizinhos mais próximos (ROY; CHOWDHURY, 2021). Para aplicações de localização, é feita uma comparação entre dados coletados da instância, referente ao objeto com posição desconhecida, com informações existentes na base de dados, por exemplo, os valores RSSI (nível de potência de sinal recebida) comparados aos valores da base de dados utilizando como cálculo a distância euclidiana (SADOWSKI; SPACHOS; PLATANIOTIS, 2020).

Conforme se observa na equação 7, para uma situação hipotética de atributos de entrada baseados no nível de potência detectado (RSSI), a primeira etapa consiste em calcular a distância euclidiana entre o registro em uma localização desconhecida j e uma posição conhecida i , presente na base de dados. A distância é calculada considerando as medições referentes a uma âncora n , de um total de âncoras M . D_{ij} é calculada entre os RSSI coletados na posição desconhecida ($RSSI_{jn}$), e o registro com posição conhecida ($RSSI_{in}$), referente a instância da base de dados i de um total de instâncias M . Após realizado o cálculo da distância entre a coleta atual e todas as instâncias da base de dados, as k correspondências mais próximas são selecionadas e as posições (x_i, y_i, z_i) desses pontos são utilizadas para estimar a localização do receptor (equação 8). No caso mais simples ($k = 1$), o algoritmo encontra a única correspondência mais próxima e usa a localização dessa instância como estimativa (CHUENURAJIT; PHIMMASEAN; CHERNTANOMWONG, 2013).

$$D_{ij} = \sqrt{\sum_{n=1}^M (RSSI_{in} - RSSI_{jn})^2}, \quad i = 1, 2, 3, \dots, M \quad (7)$$

$$(x_j, y_j, z_j) = \frac{1}{k} \sum_{i=1}^k (x_i, y_i, z_i) \quad (8)$$

5.2 Naive Bayes

Abordagens bayesianas são algoritmos de regressão probabilísticos, baseado no teorema de bayes, que assume a independência entre os atributos de entrada, desconsiderando a correlação entre as variáveis (ROY; CHOWDHURY, 2021). Em uma aplicação de localização, utilizando como exemplo a métrica de RSSI, a utilização de um filtro bayesiano em uma base de dados de *fingerprint* compara vetores de RSSI de âncoras fixas coletados a partir de um local desconhecido com valores RSSI armazenados em uma base de dados construída durante o treinamento. A posição do receptor é então estimada calculando e comparando a probabilidade de se receber o vetor de RSSI em um local específico (KLEPAL; PESCH et al., 2007). A partir do teorema de bayes, a localização da proposta em questão é estimada através da equação 9.

$$P(L_i|x) = \frac{P(x|L_i) \cdot P(L_i)}{P(x)} \quad (9)$$

Na equação 9, L_i representa o vetor de RSSI's recebido das âncoras em uma determinada posição i , armazenada na base de dados, x é o vetor de RSSI's coletados a partir de uma posição desconhecida, e $P(L_i|x)$ a probabilidade do vetor RSSI x estar na posição i , dado o vetor RSSI armazenado L_i (SADOWSKI; SPACHOS; PLATANIOTIS, 2020). Para calcular $P(x)$, conforme é visto na equação 10, a soma das probabilidades de um vetor de valores RSSI x observado em uma localização desconhecida, iteradas em relação ao vetor de medições na localização i e multiplicada pela probabilidade do vetor RSSI ocorrer na base de dados, sendo n o número de registros de locais armazenados (SADOWSKI; SPACHOS; PLATANIOTIS, 2020). $P(x|L_i)$ é calculado considerando todos os valores RSSI possíveis com distribuição gaussiana baseados nas medidas que foram coletadas durante a fase de treinamento.

$$P(x) = \sum_{i=1}^n (P(x|L_i) \cdot P(L_i)) \quad (10)$$

Nesse sentido, para determinar a localização que maximiza a função $P(L_i|x)$, considerando todos os registros de posição da base de dados, pode ser realizada uma média de probabilidades. O cálculo para essa finalidade pode ser observado na equação 11, sendo n o número de registros de vetores de RSSI da base de dados, e j referindo a uma instância específica de localização conhecida (SADOWSKI; SPACHOS; PLATANIOTIS, 2020).

$$P(L_i|x) = \frac{1}{n} \sum_{j=1}^n P(L_{ij}|x_j) \quad (11)$$

Outras abordagens baseadas na inferência bayesiana também são utilizadas em uma variedade de aplicações de localização em ambientes internos, assim como rastreamento e navegação. Filtros bayesianos podem ser empregados em aplicações de localização para redução de ruídos das leituras de sensores, de forma a reduzir o erro quadrático médio das leituras dos sensores, e no caso de sistemas com redes de sensores sem fio, minimizar o erro das estimativas de localização de outros algoritmos (ROY; CHOWDHURY, 2021). Aplicações em ambientes internos têm sido desenvolvidas utilizando filtros como o filtro de partículas e filtro de kalman, para melhora da acurácia de localização, entretanto, sendo aplicados em cenários mais especificamente de navegação, minimizando o erro de posição durante trajetórias (AYKAÇ; ERÇELEBI; ALDIN, 2017).

5.3 Árvore de regressão

O método de indução a partir de árvores de decisão pode ser considerada como uma das formas mais simples e ainda assim bem-sucedidas de Aprendizado de Máquina, técnica que utiliza uma estrutura em formato de árvore para construir os modelos de classificação ou regressão (RUSSELL; NORVIG, 2010). O ponto de partida da árvore é denominado nó raiz, de onde são geradas as ramificações, que representam as hipóteses ou testes, também chamadas de nós folhas, e dessa forma, uma árvore chega a sua decisão (discreta ou contínua), através da execução de uma sequência de testes (RUSSELL; NORVIG, 2010).

A árvore de regressão é um algoritmo capaz de criar modelos referentes a funções complexas e não lineares, particionando regiões da função através das suas ramificações. Considerando um problema de regressão com resposta contínua Y e entradas X_1 e X_2 , é possível modelar uma função não linear a partir desse princípio de particionamento. A figura 10a) mostra o plano com os atributos X_1 e X_2 , e onde são agrupadas partições, em que cada uma é possível modelar a função Y (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Para simplificar a modelagem, o espaço dos atributos de entrada é dividido nas regiões R_n , onde a função se comporta de maneira diferenciada em relação a outras regiões. As regiões são delimitadas pelas faixas t_n , que permitem, através da estrutura de uma árvore com ramificações binárias, simplificar e classificar o problema da modelagem para partições menores da função (HASTIE; TIBSHIRANI; FRIEDMAN, 2009), podendo ser modeladas como uma constante, conforme visto na figura 10b). No topo da árvore está o conjunto de dados referente a função a ser modelada, descendo o nível das ramificações os testes condicionais direcionam para um ramo ou outro da árvore, e nos nós folhas terminais são classificadas nas respectivas regiões (HASTIE; TIBSHIRANI; FRIEDMAN, 2009). Na figura 10c) é demonstrado um gráfico tridimensional da superfície do modelo a partir do algoritmo de árvore de regressão.

De forma semelhante, para funções não lineares com um número maior de entradas e múltiplas saídas, caso de problemas de localização, o algoritmo de árvore de regressão permite utilizar sua estrutura para treinar modelos de regressão através de uma constante reorganização das suas ramificações. No entanto, caso o tamanho da base de dados de treinamento for extensa, uma única árvore pode se tornar exageradamente complexa, tendendo a uma sobreajuste (do inglês *overfitting*), quando o modelo perde a capacidade de generalizar, com um ótimo desempenho para o conjunto de dados de treinamento, mas ineficaz para predição de novos resultados (ROY; CHOWDHURY, 2021).

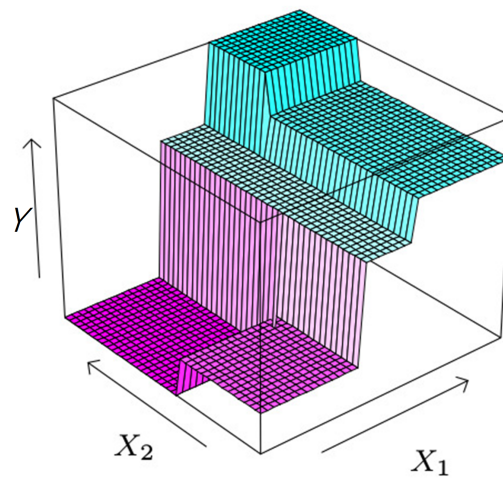
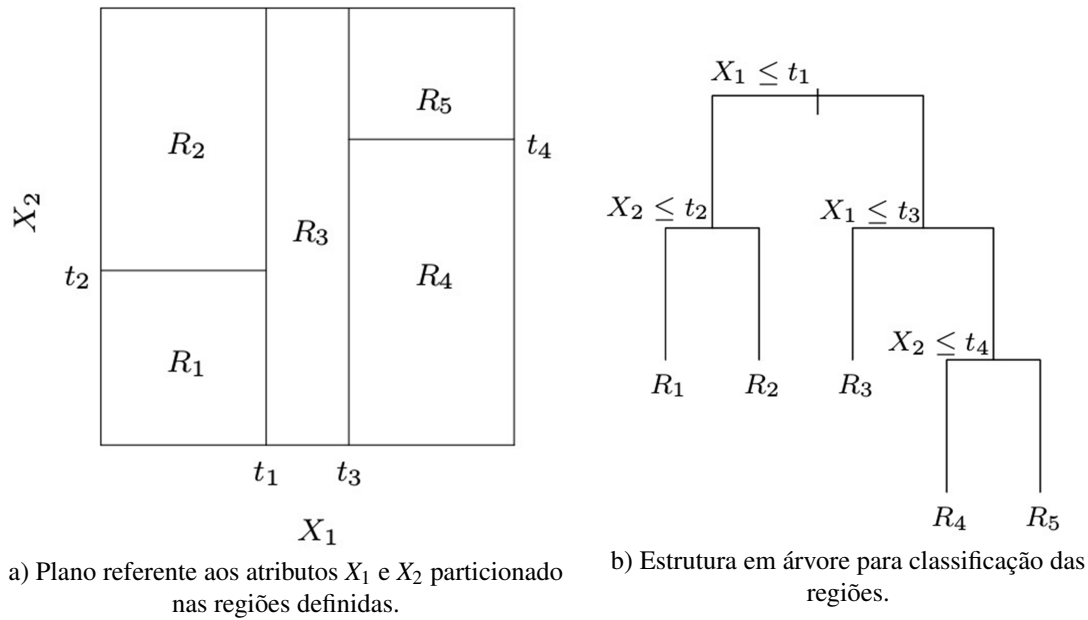
5.4 Redes neurais artificiais

Redes neurais artificiais (do inglês *Artificial Neural Networks* - ANN, também conhecidas pela sigla RNA) são estruturas inspiradas em funções de um cérebro biológico, onde são criados modelos computacionais que podem ser treinados, aprendendo a estimar e generalizar a partir da observação da relação entre dados de entrada e saída, possibilitando implementações para reconhecimento de padrões, tarefas de classificação, modelos preditivos de regressão linear e não linear, entre outras (RUSSELL; NORVIG, 2010).

No contexto de inteligência artificial, um neurônio, também chamado de nó ou perceptron, é uma unidade computacional que é composto por uma ou mais ligações de entrada, uma função de ativação, e uma conexão para a saída. É possível observar na figura 11 um modelo simplificado de um neurônio, onde o processo se inicia na inserção do vetor de entradas a_i , com cada nó de entrada ponderado pelos pesos $w_{i,j}$, a função de entrada $\sum^{in_j}$ realiza soma ponderada das entradas. Em seguida, a função de ativação g é aplicada nesta soma ponderada para gerar a saída do neurônio a_j (vide equação 12), que representa a predição do modelo (RUSSELL; NORVIG, 2010).

$$a_j = g(in_j) = g\left(\sum_{i=0}^n w_{i,j} a_i\right) \quad (12)$$

Figura 10 – Etapas do algoritmo de árvore de regressão.

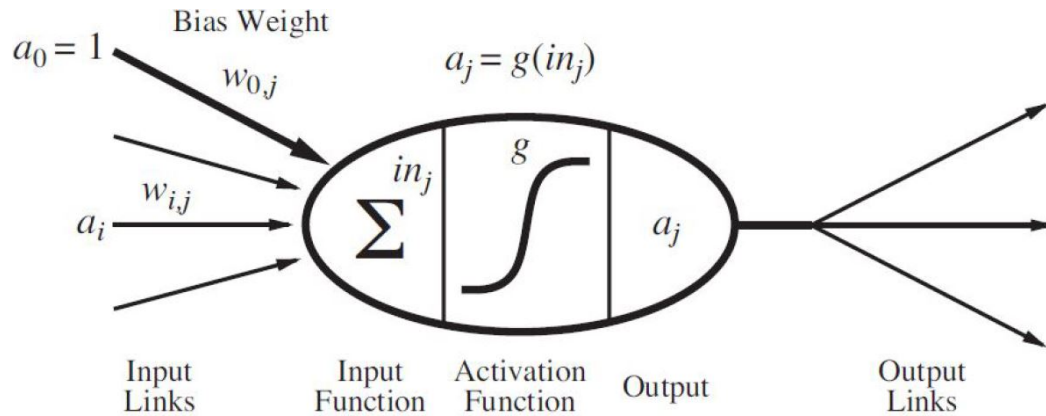


c) Superfície do modelo a partir do algoritmo de árvore de regressão.

Fonte: (HASTIE; TIBSHIRANI; FRIEDMAN, 2009)

Uma rede neural consiste basicamente em camadas de neurônios interconectados, com uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída, todas estas camadas compostas por nós, e sua modelagem é determinada por sua topologia e pelas propriedades dos neurônios (RUSSELL; NORVIG, 2010). Uma das arquiteturas mais populares de redes neurais artificiais é o perceptron multicamadas (do inglês *Multilayer Perceptron* - MLP), que será melhor detalhada a seguir.

Figura 11 – Modelo matemático simples de um neurônio.



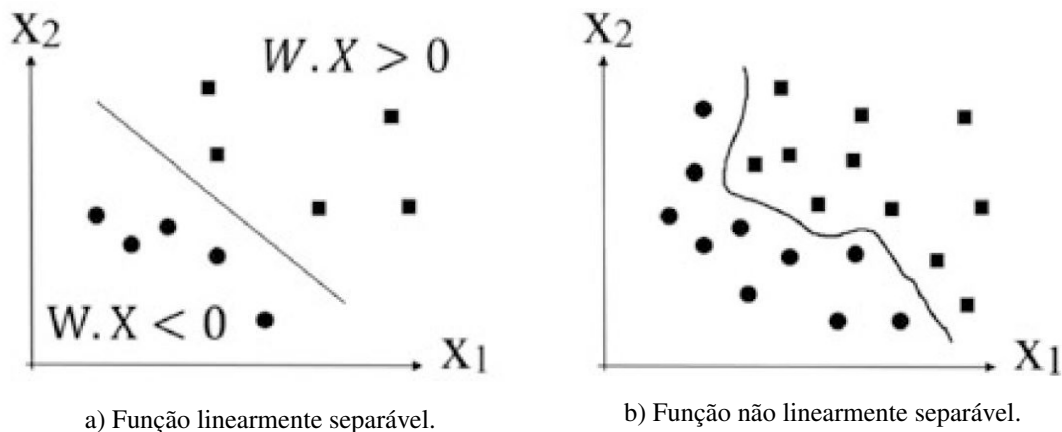
Fonte: (RUSSELL; NORVIG, 2010)

5.4.1 Perceptron multicamadas (MLP)

Uma limitação do modelo de neurônio único é sua capacidade de aprender apenas funções linearmente separáveis, ou seja, problemas que possam ser classificados apenas com a implementação de uma linha reta (TAUD; MAS, 2018). Na figura 12a é possível observar um gráfico de duas dimensões, com um exemplo de uma função linearmente separável. Entretanto há situações em que problemas que funções lineares não são capazes de resolver, e modelos mais complexos são necessários, como a curva hipotética vista na figura 12b. Contudo, uma arquitetura com múltiplas camadas de neurônios, como o perceptron multicamadas (MLP), é capaz de aprender problemas complexos, e treinar modelos para classificação ou regressão de problemas não lineares.

A arquitetura do perceptron multicamadas é composto basicamente por uma camada de entrada, com o mesmo número de neurônios aos atributos introduzidos na rede (TAUD; MAS, 2018). Os neurônios de entrada são conectados a uma segunda camada, denominada camada oculta, que terá uma quantidade de neurônios de acordo com a complexidade do problema que a rede deverá resolver, e também será totalmente conectada a camada de saída. Na figura 13 é

Figura 12 – Modelagem para funções linearmente e não linearmente separáveis.

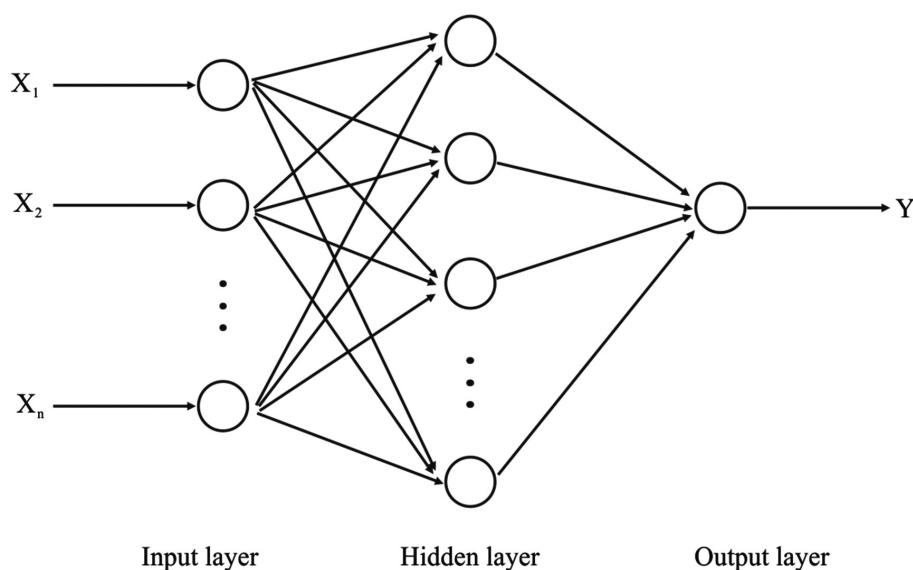


Fonte: (TAUD; MAS, 2018)

possível observar o esquema básico de uma rede neural perceptron multicamadas. O treinamento desse modelo é feito inserindo dados nos neurônios de entrada, e comparando a saída prevista pela rede com a saída real, e calculando a diferença como o erro. O erro então é propagado de volta pela rede, e é modificado o peso de cada conexão entre os nós, processo denominado de retropropagação, que é repetido o número necessário de vezes até que o erro diminua, e a rede seja ajustada para fornecer uma precisão razoável (ROY; CHOWDHURY, 2021).

Devido a possibilidade de modelagem de funções não lineares complexas, com múltiplas entradas e saídas, implementações de localização com utilização de algoritmos de aprendizado com redes neurais encontram desempenhos satisfatórios, tanto em condições com linha de visada livre (LOS), quando com ambientes com obstáculos (NLOS - *Non Line of Sight*) (ROY; CHOWDHURY, 2021).

Figura 13 – Esquema de uma rede neural perceptron multicamadas.



Fonte: (FATH; POURANFARD; FOROUGHIZADEH, 2018)

5.5 Algoritmos de agrupamento de dados

Algoritmos de agrupamento de dados são técnicas de Aprendizado de Máquina não supervisionado, ou seja, que não exige dados previamente rotulados ou classificados, e tem como principal função agrupar as instâncias de um determinado conjunto de dados a partir de algum parâmetro de similaridade entre os dados (ROY; CHOWDHURY, 2021). Um dos algoritmos clássicos para o agrupamento de dados é o *K-means*, que a partir da definição da quantidade de grupos no momento inicial da implementação, através do parâmetro K , irá definir a forma como o algoritmo irá agrupar os dados focando em uma otimização a partir da escolha do melhor K , que na maioria das vezes não é conhecido a priori (ROY; CHOWDHURY, 2021).

O processo de agrupamento é iniciado com um conjunto de dados $X = \{x_1, \dots, x_n\}$, onde n é a quantidade total de instâncias, e se deseja agrupar em K clusters (agrupamento de dados), em um conjunto $C = \{c_1, \dots, c_k\}$. O algoritmo irá inicialmente de forma aleatória definir pontos centrais para cada um dos clusters (também chamado de centróide) m_k , e em seguida passar para a etapa de otimização, em que o objetivo é minimizar o erro total $E(C_k)$ (equação 13), referente a distância euclidiana de cada instância x_i em relação ao centróide m_k de cada cluster, e assim o agrupamento

será otimizado alterando até encontrar o conjunto de centróides que produzir o menor erro de distância total para os K clusters (JAIN, 2010).

$$E(C) = \sum_{k=1}^K \sum_{x_i \in C_k} ||x_i - m_k||^2 \quad (13)$$

O uso de algoritmos de agrupamento de dados permite estratégias para redução de esforços na criação de bases de dados de *fingerprint*, como a etapa de análise de cena, onde cada posição coletada deve ser rotulada com sua localização (ROY; CHOWDHURY, 2021).

6 Trabalhos Correlatos

Diversos trabalhos têm sido publicados sobre sistemas de localização com dispositivos sem fio em ambientes internos, que podem utilizar diferentes tipos de tecnologia, métricas e técnicas para realizar a tarefa de estimar a posição desconhecida de algum dispositivo, e proporcionam bases para comparação de desempenho dessas abordagens. Nesse sentido, com relação as redes de sensores sem fio citadas neste trabalho, como Wi-Fi, UWB, *Zigbee*, e *Bluetooth Low Energy* (tecnologia escolhida para implementação deste trabalho) essa revisão apresentará trabalhos que trazem a técnica estado da arte para localização em ambientes internos, algoritmos de Aprendizado de Máquina com base na abordagem de *fingerprint*.

A abordagem de análise de cena ou *fingerprint* utiliza conceitos e procedimentos básicos de algoritmos de Aprendizado de Máquina, onde, a partir de uma base de dados com informações que representam o padrão de comportamento de uma determinada variável de entrada, seja possível treinar e modelar um algoritmo para calcular a localização estimada de um elemento na etapa seguinte de testes, com base em novos dados de entrada. Nesse sentido, diversos algoritmos de Aprendizado de Máquina têm sido implementados para sistemas de localização em ambientes fechados com a abordagem de *fingerprint*, como redes neurais, árvores de regressão, K-vizinhos mais próximos (KNN), entre outras, assim como diversas tecnologias e métricas para distância (ZAFARI; GKELIAS; LEUNG, 2019). Um resumo de alguns trabalhos nessa área, com seus algoritmos e tecnologias utilizadas pode ser visto na tabela 2.

Tabela 2 – Trabalhos na área de localização em ambientes internos com redes de sensores sem fio.

Trabalho	BLE	UWB	Wi-Fi	Zigbee	Métrica	Algoritmos aplicados
(AHMADI; BOUALLEGUE, 2015)				✓	RSSI	Árvore de regressão, mínimos quadrados, MLP
(BELMONTE-HERNÁNDEZ et al., 2019)	✓		✓	✓	RSSI	<i>Deep Learning</i> , Filtro de partículas
(BREGAR; HROVAT; MOHORCIC, 2016)		✓			CIR	MLP
(CHUENURAJIT; PHIMMASEAN; CHERNTANOMWONG, 2013)				✓	RSSI	KNN
(DJOSIC et al., 2021)		✓			TOF	Trilateração, KNN
(JONDHALE et al., 2020)					RSSI	GRNN, MLP, RBFN, FFNT, trilateração
(MOHAMMADI et al., 2017)	✓				RSSI	DRL semisupervisionado
(PENG et al., 2017)					RSSI	SWBN
(SADOWSKI; SPACHOS; PLATANIOTIS, 2020)	✓		✓	✓	RSSI	Trilateração, KNN, Naive Bayes
(XIAO et al., 2017)	✓				RSSI	Denoising autoencoder, KNN

Fonte: Autor.

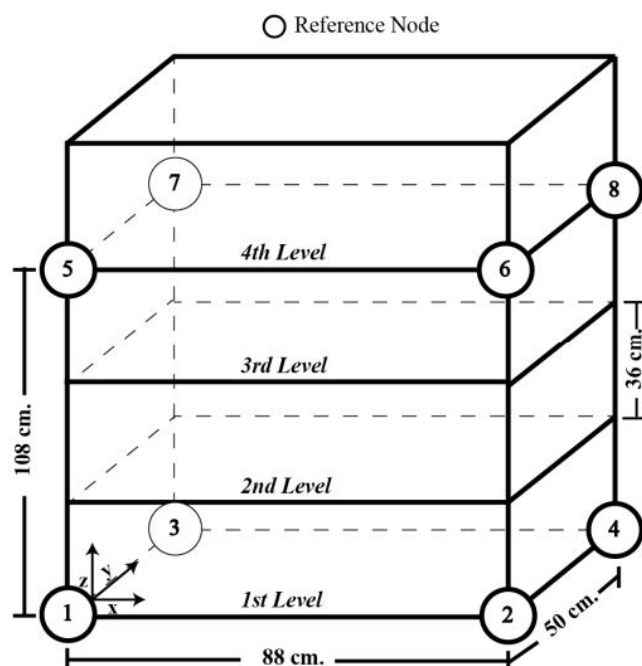
Um dos primeiros trabalhos desenvolvidos com a técnica de *fingerprint* para localização 3D com redes de sensores sem fio, o artigo de (CHUENURAJIT; PHIMMASEAN; CHERNTANOMWONG, 2013) utiliza o padrão de tecnologia *Zigbee* baseado no RSSI (índice de potência de sinal recebido). Como algoritmo é utilizada a técnica de KNN utilizando

como métrica a distância euclidiana entre os dados de entrada e as instâncias presentes na base de dados, classificando como vizinhos os atributos da base mais próximos. Neste trabalho os autores implementam âncoras em uma prateleira de livros com 4 níveis de altura, conforme se observa na figura 14. Devido a seus bons resultados, e simplicidade de implementação, a técnica de KNN têm sido utilizada amplamente para resultados comparativos em trabalhos de localização utilizando *fingerprint*.

O trabalho de Sadowski, Spachos e Plataniotis (2020) compara a técnica de localização em um ambiente 2D baseada em trilatação com dois algoritmos de análise de cena, KNN e Naive Bayes, utilizando 3 diferentes tecnologias de dispositivos sem fio, *Zigbee*, *Bluetooth Low Energy* e *WiFi*, e como atributo de entrada os níveis de potência de sinal detectados (RSSI). Os testes são feitos em 3 cenários bidimensionais com diferentes tamanhos e níveis de interferência. A respeito dos resultados, a melhor acurácia (menor erro quadrático médio entre a localização prevista e a real) é obtida com o modelo baseado no algoritmo de KNN, enquanto o pior desempenho é observado com a técnica de trilatação, devido à forte dependência dessa técnica dos métodos de medição de distância (conforme comentado na seção 4.2), neste caso o modelo de perda de sinal de RSSI, que em um cenário com obstáculos ou interferências tem um comportamento instável. Entretanto, em relação aos resultados obtidos neste trabalho, é importante ressaltar que algoritmos de localização baseados em Aprendizado de Máquina precisam da construção de bases de dados para treinamento e consulta durante os testes, gerando maiores esforços para sua implementação e custos computacionais superiores aos da trilatação, que devem ser considerados na avaliação de viabilidade da solução (SADOWSKI; SPACHOS; PLATANIOTIS, 2020).

O algoritmo de árvore de regressão é proposto para localização em ambientes internos no trabalho de Ahmadi (2015). Este trabalho utiliza uma base de dados de localização a partir de dispositivos com a tecnologia *Zigbee*, utilizando como métrica o RSSI detectado das âncoras. No artigo, é descrita a modelagem da árvore, com uma iteração entre todas as divisões binárias possíveis para cada folha, com o erro sendo avaliado pelo erro quadrático médio (do

Figura 14 – Desenho do setup experimental para localização 3D.



Fonte: (CHUENURAJIT; PHIMMASEAN; CHERNTANOMWONG, 2013)

inglês *Mean Square Error* - MSE) referente a localização estimada e real da base de dados. Como em outros trabalhos, os resultados são comparados a outros algoritmos da literatura, neste caso aos métodos de mínimos quadrados, e perceptron multicamadas (MLP), e nos testes comparativos os autores obtêm melhores resultados de acurácia e menor tempo de processamento com a árvore de regressão em comparação com os outros dois algoritmos, principalmente quando a quantidade de âncoras é reduzida no ambiente.

O trabalho de Peng et al. (2017) implementa o método SWBN (*spectral clustering and weighted backpropagation neural network*) para melhorar a acurácia de localização 3D em ambientes internos com o uso de RSSI. A proposta do processo de agrupamento de dados visa, com base nos pontos de referência capturados na fase *offline* do *fingerprint*, dividir a massa de dados em n grupos, baseados na correlação e similaridade de cossenos dos vetores de RSSI das âncoras (ou access points), no caso, no ângulo entre esses vetores no espaço, atribuindo importância às diferenças refletidas na direção dos mesmos, e não em sua posição, característica que o diferencia de outros algoritmos implementados na literatura. O trabalho não detalha que tipo de tecnologia sem fio é utilizada.

O trabalho de Mohammadi et al. (2017) segue uma abordagem para localização em ambientes internos baseada em coleta de dados de RSSI de 13 âncoras comunicando-se com o protocolo *Bluetooth Low Energy*, utilizando como ambiente de testes a biblioteca Waldo na *Western Michigan University*, um ambiente real com pilares, equipamentos, circulação de pessoas, enfim, variáveis que podem deteriorar os sinais dos *beacons*. A proposta utiliza um modelo de aprendizado profundo por reforço semi-supervisionado, baseando-se no argumento da necessidade de treinar algoritmos com dados não rotulados, em razão do esforço realizado na criação de bases de dados de *fingerprint*.

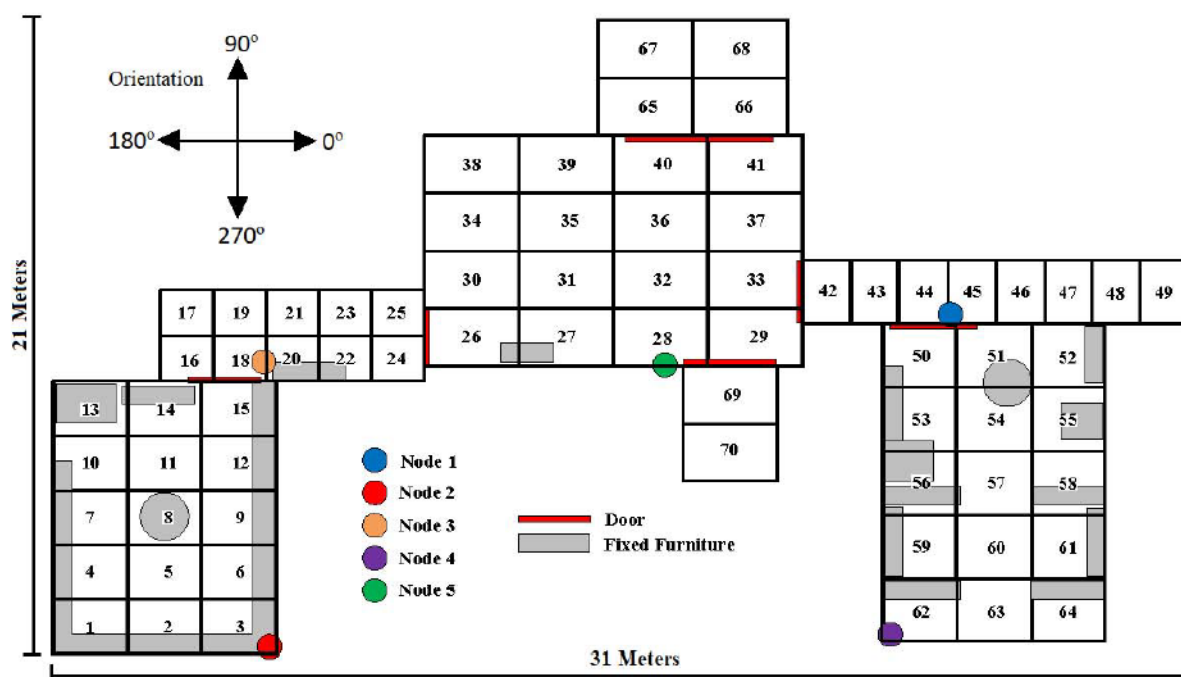
Neste trabalho, o algoritmo de aprendizado profundo por reforço procura treinar um modelo capaz de aprender as melhores políticas de posicionamento para predição de uma localização desconhecida a partir de dados rotulados e não rotulados do ambiente, e obter melhores resultados em comparação com o uso de um conjunto de dados apenas com instâncias rotuladas em um modelo totalmente supervisionado. Os resultados da abordagem mostram uma melhoria de 23% na precisão da localização com o modelo semi-supervisionado proposto em comparação aos modelos supervisionados (MOHAMMADI et al., 2017). Vale ressaltar que este trabalho apresenta mais uma proposta de algoritmo de navegação do que de localização estática, devido a necessidade de diferença temporal para abordagens com aprendizagem por reforço.

O trabalho de Belmonte-Hernandez et al. (2019) faz uma análise dos parâmetros mais influentes na construção de um *fingerprint* e implementa uma abordagem com *deep learning* combinado com métodos estatísticos para estimar localização em ambientes internos. Esse trabalho utilizou 3 dispositivos de redes de sensores sem fio (*Zigbee*, *Bluetooth Low Energy* e Wi-Fi) mas também uma unidade de medida inercial (do inglês *Inertial measurement unit* - IMU) para monitoramento da orientação e aceleração do elemento móvel. Na figura 15 pode-se observar o layout do ambiente de testes e a divisão de células para coleta dos dados de *fingerprint*. Foram definidas dentro do ambiente 70 células, pontos de medição onde cada registro tem como atributos de entradas os RSSI detectados de 5 âncoras (destacadas em cores diferentes na figura 15), além de dados de aceleração e orientação da IMU, rotulados com a localização da célula. Os atributos de entrada e rótulos foram usados para treinamento de uma rede neural profunda. Como para essa proposta também é considerada a movimentação do elemento (colhida com o sensor IMU), além do uso de redes neurais também são aplicadas abordagens estatísticas como filtros gaussianos e de partículas para melhora da performance do rastreamento (BELMONTE-HERNÁNDEZ et al., 2019).

Outras métricas de distância ou qualidade do sinal podem ser utilizadas para aplicações de localização. No trabalho de Bregar, Hrovat e Mohorcic (2016) é utilizado como variável de entrada a resposta ao impulso do canal (do inglês *Channel Impulse Response* - CIR), dado referente a amplitude e fase do canal em no recebimento de um determinado sinal. Esse atributo é utilizado juntamente com uma rede neural perceptron multicamadas (MLP) para classificação da condição de sinais recebidos como com linha de visada direta (LOS) ou sem linha de visada direta (NLOS). Tecnologias como *Ultra Wideband* (UWB) permitem também a utilização de métricas de distância baseadas em tempo, como no

trabalho de (DJOSIC et al., 2021) que utiliza a técnica de medição de tempo de diferença de chegada (TDOA) para um abordagem que mistura trilateração e KNN para localização em ambientes fechados.

Figura 15 – Planta baixa do ambiente de testes, com células e posição das âncoras.



Fonte: (BELMONTE-HERNÁNDEZ et al., 2019).

Técnicas tradicionais de localização com redes de sensores sem fio como a trilateração necessitam prioritariamente da acurácia de métodos de medição de distância, fazendo com que oscilações, ruídos e outras anormalidades de sinais afetem sua performance. Em contrapartida, abordagens com rede neurais artificiais por padrão não necessitam de um conhecimento prévio do comportamento ou modelo de sinais dentro de ambientes fechados e podem ser utilizadas em aplicações de localização mesmo em condições de falta de linha de visada direta (JONDHALE et al., 2020). Nesse sentido, o trabalho de Jondhale et al. (2020) avalia o desempenho de algumas arquiteturas de redes neurais em relação a técnica de trilateração para localização indoor. As arquiteturas empregadas foram: rede neural de regressão generalizada (GRNN), perceptron multicamadas (MLP), rede de função de base radial (RBFN) e rede neural *feedforward* (FFNT). Para esse trabalho nenhum dispositivo sem fio específico foi utilizado, mas sim medições RSSI são geradas usando o modelo específico do software MATLAB®.

Aplicações de localização em ambientes internos com uso do nível de sinal detectado (RSSI) podem sofrer com problemas de ruído na propagação de sinal ou multipercursos (quando sinais de rádio de um mesmo transmissor alcançam a antena receptora por mais de caminho), entretanto, resultados obtidos em (JONDHALE et al., 2020) demonstram que o uso de redes neurais para essa abordagem podem trazer resultados de acurácia em comparação com a técnica de trilateração, sendo que a que apresentou menor erro médio a rede neural de regressão generalizada (GRNN).

Ainda no contexto do impacto de perturbações e flutuações nos níveis de sinal detectado, no trabalho de Xiao et al. (2017), é proposto o uso de um modelo de rede neural chamado *denoising autoencoder* (para eliminação de ruído), que tem uma camada de entrada que faz um processo chamado de "corrupção estocástica", que basicamente escolhe entre os valores do vetor de entrada (nível RSSI das âncoras) e os anula multiplicando-os por 0, realizando uma série de iterações de treinamento para verificar quais entradas podem ser anuladas de forma a manter ou aumentar

o desempenho de localização. Na proposta, é feito o treinamento de uma rede eliminadora de ruído para cada local de referência da base de dados de *fingerprint* para localização 3D em ambientes internos. Esse processo tem o objetivo de caracterizar a ocorrência de perturbações na coleta de dados durante o treinamento.

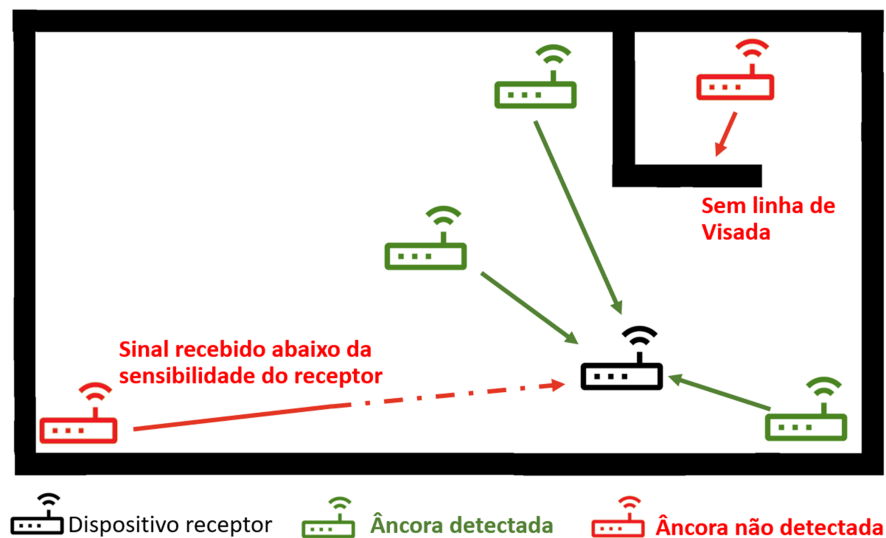
Na etapa de testes proposta por Xiao et al. (2017), cada novo vetor de RSSI's coletados de um ponto de localização desconhecida é comparado aos registros dos pontos de referência da base de dados (já processados pela rede eliminadora de ruídos) e um algoritmo de KNN é utilizado para selecionar os k vizinhos mais próximos, e através da média ponderada dos locais de referência destes calcular a localização estimada. O trabalho traz uma interessante contribuição com relação a possibilidades do uso de redes neurais para tratamento de perturbações comuns a medição de RSSI.

Esta dissertação considera o trabalho de Xiao et al. (2017) como uma base importante tanto para a sua hipótese quanto para sua abordagem proposta. Com relação a hipótese, a verificação que dentre os elementos de entrada existem atributos que não contribuem para uma boa performance dos algoritmos de localização, e até pelo contrário, podem ser considerados ruídos de medição e sua eliminação pode melhorar sua acurácia, nesse sentido é necessária uma abordagem para tratar esses parâmetros. A proposta de Xiao et al. (2017) propõe um tratamento para os atributos de entrada para que flutuações e ruídos sejam anulados, assim como esta dissertação, que com a abordagem de **filtro de âncoras** pretende utilizar apenas leituras referentes aos maiores níveis de sinal detectadas, associados a localização das respectivas âncoras.

7 A abordagem do filtro de âncoras

Conforme apresentado no capítulo 6, algoritmos de Aprendizado de Máquina têm sido aplicados em ambientes internos com o intuito de estimar a posição de dispositivos a partir dos sinais das entradas das âncoras. Considerando as discussões realizadas na seção 3.1, particularmente o modelo de comportamento logarítmico decrescente da perda de potência de sinal (RSSI) $\times$ distância física, percebe-se que à medida que o espaço entre o dispositivo emissor e o receptor aumenta, há uma degradação significativa na acurácia da medição de distância. Essa degradação da potência detectada também ocorre em condições de falta de linha de visada direta (NLOS) ou na presença de obstáculos entre o emissor e o receptor. Essa situação é destacada esquematicamente na figura 16.

Figura 16 – Condições de detecção e não detecção de âncoras.



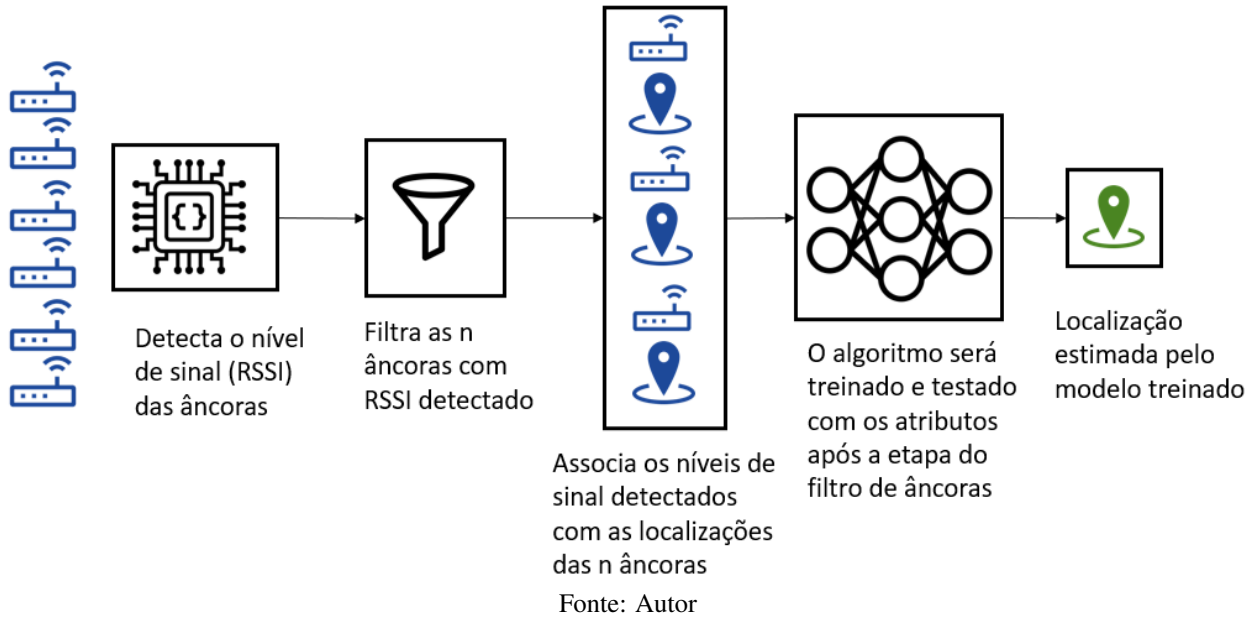
Fonte: Autor

Nesse cenário, tomando como parâmetro de entrada os níveis de potência de sinal (RSSI) recebidos das âncoras no ambiente, é razoável assumir como **hipótese** que as redes com sinais de maior potência em cada ponto do ambiente sejam mais relevantes para os algoritmos de localização. Em outros termos, o presente trabalho busca investigar se utilizando apenas parte das âncoras detectadas pelo dispositivo receptor – ou seja, adotando-se uma estratégia doravante denominada **filtro de âncoras** – os algoritmos de localização poderiam obter resultados similares utilizando menos atributos de entrada. Ainda, sob outro prisma, a possibilidade de algoritmos de Aprendizado de Máquina trabalharem com um número reduzido de dimensões de entrada pode ter implicações tais como a redução da necessidade de tratamento prévio de dados (para casos de sinais não detectados, por exemplo) e da complexidade do treinamento desses algoritmos.

A figura 17 sintetiza a abordagem proposta. A partir de um número de dimensões n escolhido para o filtro, serão selecionadas dentre as âncoras detectadas aquelas com os n maiores níveis de potência. Importante ressaltar que a utilização da estratégia do filtro de âncoras implica em alterações na dinâmica do sistema tanto durante o aprendizado – com diferentes implicações a depender do algoritmo de localização utilizado – quanto em tempo de execução. A seção 7.1, a seguir, aborda com mais detalhes o algoritmo do filtro de âncoras utilizado para adequação da base de

dados. A seção 7.2 destaca os procedimentos relacionados aos algoritmos de Aprendizado de Máquina que serão abordados no decorrer do trabalho.

Figura 17 – Abordagem proposta para o filtro de âncoras.



7.1 Algoritmo para adequação da base de dados para a proposta (filtro de âncoras)

O algoritmo 1 detalha o procedimento proposto para adaptar a base de dados utilizando a estratégia filtro de âncoras. O algoritmo inicialmente recebe a base de dados de *fingerprint*, as posições das âncoras fixadas no ambiente e o número de dimensões n , que define a quantidade de âncoras que serão consideradas. O algoritmo realiza então uma ordenação dos registros de acordo com a potência detectada, classificando do maior para o menor nível de potência. Após a ordenação, as âncoras com as n maiores potências são selecionadas, e o número de dimensões de entrada é então reduzido a esses valores. Desta forma, uma instância f da base de dados anteriormente organizada como $f = \{r, p\}$, onde r contém os registros $r = \{r_1, r_2, \dots, r_m\}$ das m âncoras detectadas em uma posição p , passará a ter o registro $j = \{j_1, j_2, \dots, j_n\}$, referente as n maiores potências detectadas pelo receptor, e cada item do registro apontará a potência detectada e localização da âncora emissora $j_n = \{r_n, x_n, y_n, z_n\}$. Cada instância redimensionada é adicionada como uma nova entrada em uma nova base de dados, que ao final da estrutura de repetição gera o arquivo da base de dados adaptada. Relevante observar que este algoritmo destina-se prioritariamente à adaptação da base de dados para o treinamento dos algoritmos de localização. De forma análoga, em tempo de execução, apenas as leituras das n âncoras de maior potência serão consideradas.

7.2 Algoritmos de Aprendizado de Máquina para localização

A abordagem de filtro de âncoras proposta realizará uma redução da dimensionalidade dos atributos de entrada do sistema de localização 3D em ambientes internos. Essa redução, evidentemente, terá impactos nos algoritmos de Aprendizado de Máquina selecionados para estimar a posição do dispositivo no ambiente. Nesse cenário, o presente trabalho propõe investigar o impacto desta estratégia sobre 3 (três) diferentes algoritmos de Aprendizado de Máquina:

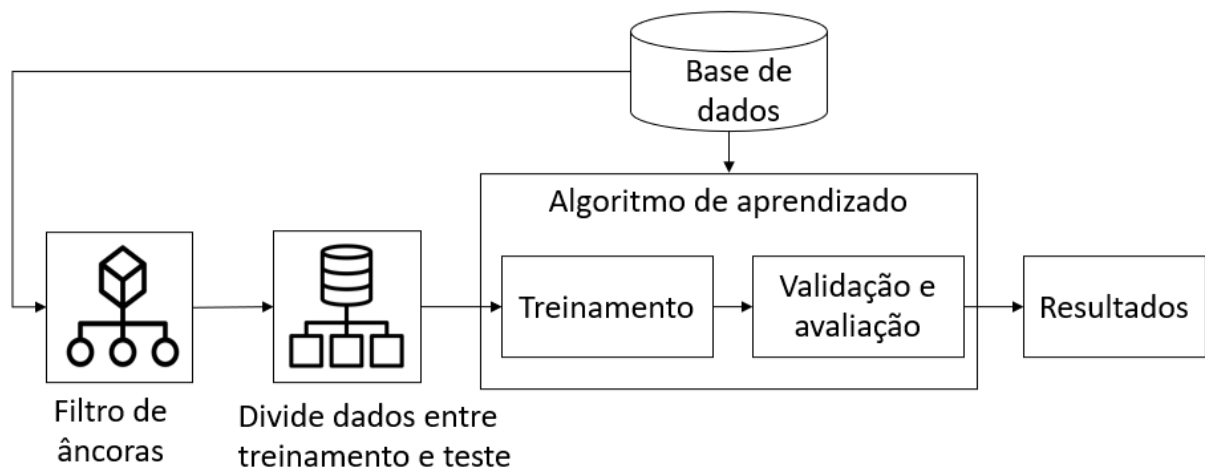
Algoritmo 1: Algoritmo para filtro de âncoras para adequação da base de dados.

1. Carrega base de dados de *fingerprint*;
 2. Carrega informações de localizações das âncoras;
 3. $n \leftarrow$ Define a quantidade de âncoras para filtro, referente aos maiores níveis de potência detectada dentre os registros;
 4. **for** cada registro da base de dados **do**
 - 4.1. **Registros de RSSI** $\leftarrow$ Ordena registros RSSI de âncoras de acordo com o nível de potência detectada;
 - 4.2. **Registros de RSSI** $\leftarrow$ Seleciona n primeiros registros de RSSI;
 - 4.3. **Nova instância** $\leftarrow$ Associa a posição da âncora com seu respectivo RSSI da lista **Registros de RSSI**;
 - 4.4. Adiciona **Nova instância** para a base de dados adaptada;
 - end**
 5. Cria arquivo de base de dados para a abordagem de filtro de âncoras;
-

1. K-vizinhos mais próximos (KNN);
2. Árvore de regressão, e
3. Rede neural Artificial (RNA) do tipo perceptron multicamadas (MLP).

O fluxo de trabalho para essa investigação é apresentado esquematicamente na figura 18. O processo tem início com o carregamento da base de dados de *fingerprint*, que contém os dados coletados em campo, com os níveis de RSSI detectados associados a posição do receptor. Para adaptar os dados de entrada (níveis RSSI das âncoras) ao formato da proposta de filtro de âncoras, as instâncias da base de dados são submetidas ao algoritmo descrito na seção 7.1. A base de dados já organizada passa então por um processo para particionar os conjuntos de treinamento, validação e testes, para utilização no treinamento dos algoritmos previamente definidos. Os materiais e métodos utilizados nos testes para verificação do desempenho dos algoritmos de Aprendizagem de Máquina serão melhor detalhados nas próximas seções.

Figura 18 – Fluxo de trabalho com os algoritmos de Aprendizagem de Máquina.



Fonte: Autor.

8 Criação da base de dados para Fingerprint

Neste capítulo serão descritas as etapas e os procedimentos realizados para a criação da base de dados de *fingerprint* para algoritmos de localização 3D em ambientes internos. Os seguintes tópicos serão abordados: *i)* pesquisa de bases de dados existentes; *ii)* escolha do ambiente para testes; *iii)* definição de posicionamento das âncoras e pontos de medição; *iv)* suporte utilizado para medição; *v)* procedimento para a coleta de dados; *vi)* algoritmos para coleta de dados, e *vii)* testes preliminares.

8.1 Pesquisa de base de dados para localização em ambientes internos utilizando fingerprint

Uma pesquisa de trabalhos acadêmicos foi inicialmente realizada para verificação da existência de alguma base de dados de localização 3D em ambientes internos disponível para comparação de desempenho de algoritmos. Inicialmente se pesquisou no site *Google Scholar* por trabalhos publicados a partir de 2015, utilizando as palavras-chave: *3D indoor localization*, *3D indoor positioning*, *RSSI BLE Database*, *3D fingerprint*. Nos 100 primeiros resultados apresentados pela pesquisa utilizado cada palavra-chave não foi encontrado nenhum trabalho com base de dados disponível para outros pesquisadores, apesar de existirem publicações sobre algoritmos de localização tridimensional para ambientes internos com uso de base de dados para *fingerprint*. O mesmo critério de pesquisa foi utilizado nos sites dos journals: *IEEE (Sensors, Internet of Things, Access)*, *Springer*, *Elsevier* e *MDPI Sensors*.

Também foram pesquisados em sites repositórios de bases de dados: *Kaggle*, *UCI machine learning repository*, *github* e *IEEE Dataport*. Para a pesquisa foram utilizadas as mesmas palavras-chave citadas anteriormente. Na pesquisa apenas foram encontradas bases de dados para *fingerprint* 2D. O resultado dessa pesquisa demonstrou a necessidade da criação e disponibilização de uma base de dados tridimensional para algoritmos de análise de cena (*fingerprint*), de forma a dar suporte às análises necessárias no âmbito do presente trabalho, e também para uso e comparação de desempenho de algoritmos implementados por outros pesquisadores.

8.2 Escolha do ambiente

O laboratório do Grupo de Automação e Sistemas Integrados (GASI), com aproximadamente 150 m², foi escolhido como ambiente para o levantamento da base de dados tridimensional devido ao amplo espaço interno disponível para ensaios, existência de diversidade de elementos no ambiente (bancadas, motores, dispositivos eletrônicos, fios, cabos e correlatos), existência de pontos de alimentação em abundância para as âncoras, além da existência de aparelhos de ar condicionado, que permitem manter condições ambientais e de temperatura semelhantes por um período prolongado durante os períodos de testes e coleta de dados.

No contexto de uma implementação de localização 3D para ambientes internos, o local escolhido proporciona um espaço onde estão contidas condições de linha de visada livre (LOS) - no espaço livre acima das bancadas - e falta de linha de visada direta (NLOS) - abaixo da altura das bancadas, com adição de obstáculos como equipamentos e cadeiras. Estas características do local possibilitam testar a robustez dos algoritmos de localização implementados em condições favoráveis e adversas.

8.3 Definição do posicionamento das âncoras e pontos para medição

De posse da planta baixa do laboratório do GASI, foi realizada uma investigação dos possíveis pontos para instalação das âncoras. Cada dispositivo exige um ponto com alimentação, e uma boa visão do ambiente. Considerando o plano x,y as âncoras foram preferencialmente posicionadas nas extremidades do laboratório, próximas às paredes. Depois de definidas as posições x,y das âncoras, procurou-se explorar diferentes níveis no eixo z.

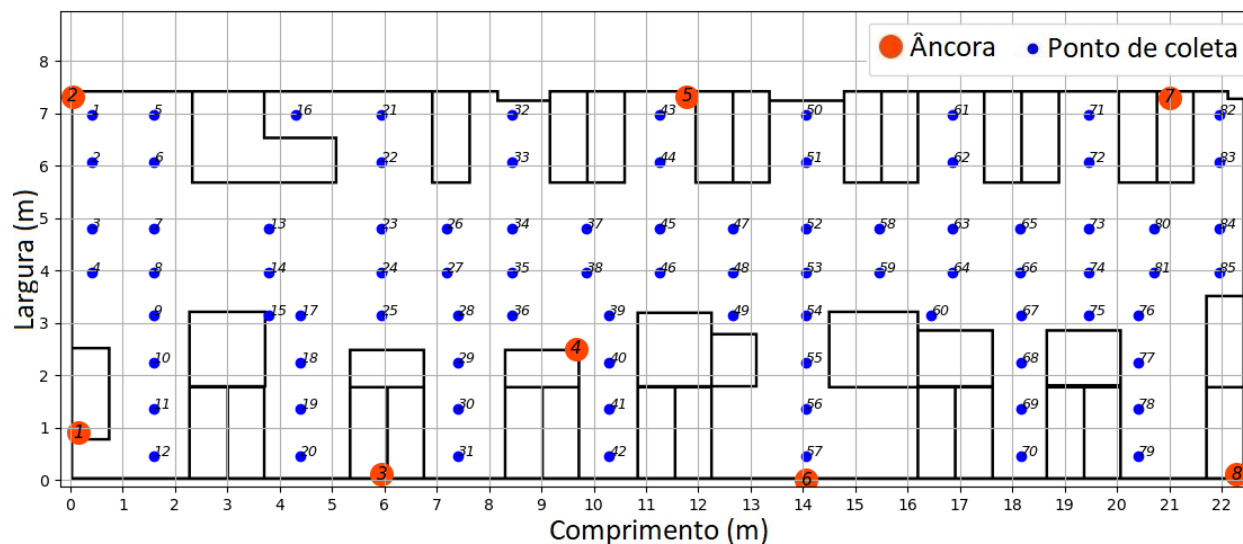
Em seguida, foram analisados os pontos para medição e coleta para a base de dados de *fingerprint*. A proposta inicial que norteou a definição dos pontos foi a intenção de gerar um *grid* 3D, com distribuição aproximadamente homogênea dos pontos nos 3 eixos. Entretanto, foi necessário alterar a posição de alguns dos pontos para a alocação do suporte para medição (descrito na próxima seção). Sendo assim foram descartados pontos acima de bancadas e equipamentos. Com base nessas diretrizes, foram definidos ao total as coordenadas x e y para 85 pontos de medição. A posição no plano x, y e z (altura) das âncoras pode ser observada na tabela 3. A posição dos pontos de coleta e das âncoras pode ser visualmente observada na figura 19.

Tabela 3 – Posição das âncoras.

Âncora	x	y	z
1	0.16	0.90	1.35
2	0.05	0.10	0.42
3	5.95	0.10	1.35
4	9.66	2.50	2.30
5	11.78	7.33	1.35
6	14.05	0.05	0.42
7	21.00	7.30	0.75
8	22.29	0.10	1.35

Fonte: Autor.

Figura 19 – Posição dos pontos de medição para *fingerprint* e das âncoras no laboratório do GASI.



Fonte: Autor.

8.4 Suporte para medição

Para realizar as medições em diferentes níveis de altura foi utilizado um suporte metálico com extensão de canos de PVC, regulável em 3 diferentes níveis: 1m, 2m e 2.90m (ligeiramente abaixo dos 3.00m, pé direito do ambiente). Nas figuras 20a e 20b se observa o suporte em 2 níveis diferentes para 2.90m e 1m.

Figura 20 – Suporte para medição.



a) Suporte de medição com 2.90m.



b) Suporte de medição com 1m.



c) Medição sem suporte (0m).

Fonte: Autor.

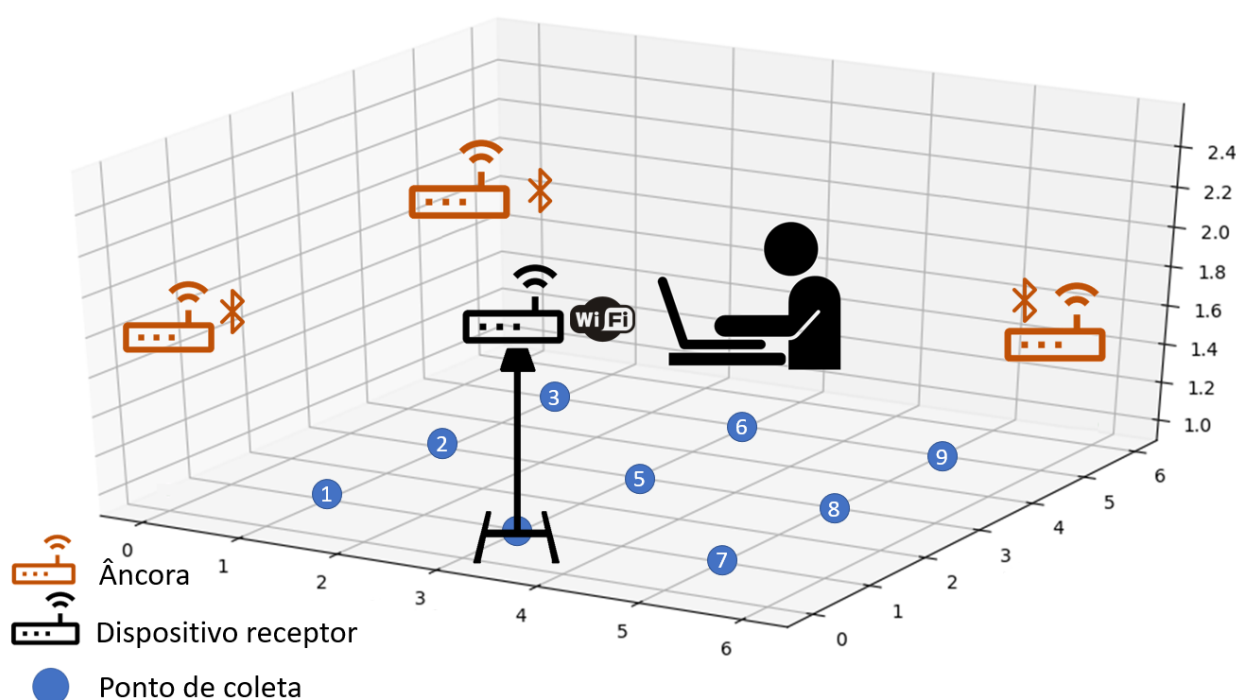
Além do suporte foi utilizado um *power bank* para alimentação 5V DC do dispositivo de coleta de dados durante os testes. Para a medição no nível do chão (0m) foi ao dispositivo de coleta alimentado pelo *power bank* foi posicionado

no chão nos pontos de medição (figura 20c). O dispositivo foi configurado (procedimento descrito a seguir) para ler e armazenar o nível de potência (RSSI) das redes detectadas no ambiente interno.

8.5 Procedimento da coleta

Após definidos os pontos de medição para a criação da base de dados para *fingerprint* no laboratório do GASI, foi realizada a marcação no piso das 85 posições possíveis. O processo teve início com o ajuste do suporte em sua maior altura (2.90m), com o dispositivo de coleta de dados fixado no topo da estrutura, e alimentado pelo *power bank*. O dispositivo foi configurado para fazer a coleta dos níveis de sinal recebidos (protocolo *Bluetooth Low Energy*) das âncoras e enviar os dados dessa coleta por Wi-Fi para o computador, responsável por agregar e registrar as informações recebidas. O *setup* do processo de coleta de dados pode ser observado esquematicamente na figura 21.

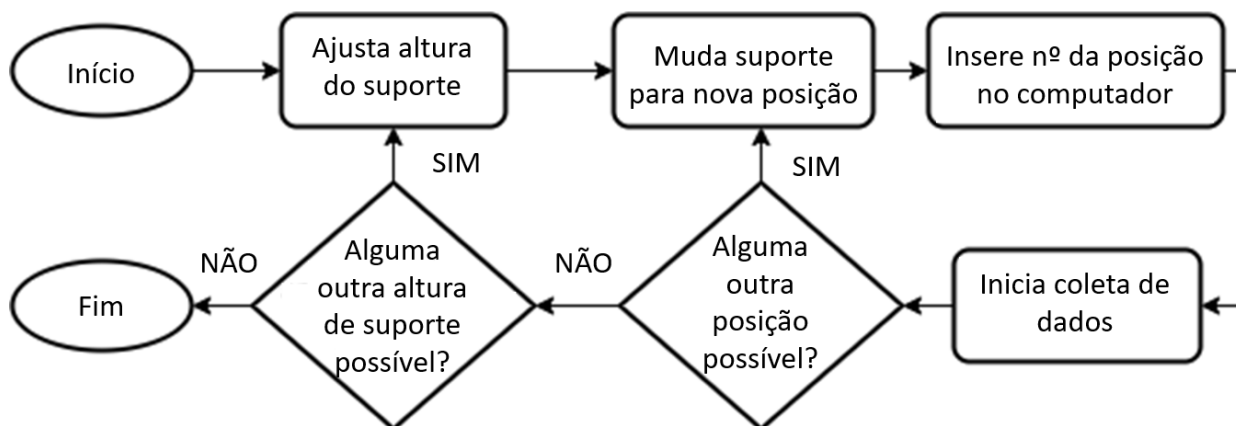
Figura 21 – Setup experimental para coleta e criação da base de dados.



Fonte: Autor.

O fluxo de trabalho para a coleta de dados de *fingerprint* é detalhado na forma de um fluxograma na figura 22. Para cada altura de coleta (0, 1m, 2m e 2.90m), o dispositivo de coleta foi posicionado em cada um dos 85 pontos de medição. O número da posição foi inserido no computador, que realizou a associação dos níveis de sinal recebidos de cada âncora com a localização atual. O algoritmo implementado para realizar a coleta dos dados das âncoras será detalhado na próxima seção. A aquisição foi realizada em um único turno de trabalho, durante aproximadamente 4 horas, com 5 amostragens de cada um dos 340 pontos selecionados, ou seja, 85 pontos em 4 alturas distintas (0m, 1m, 2m, 2.90m), totalizando 1.700 medições.

Figura 22 – Fluxograma do procedimento de coleta e criação da base de dados.



Fonte: Autor.

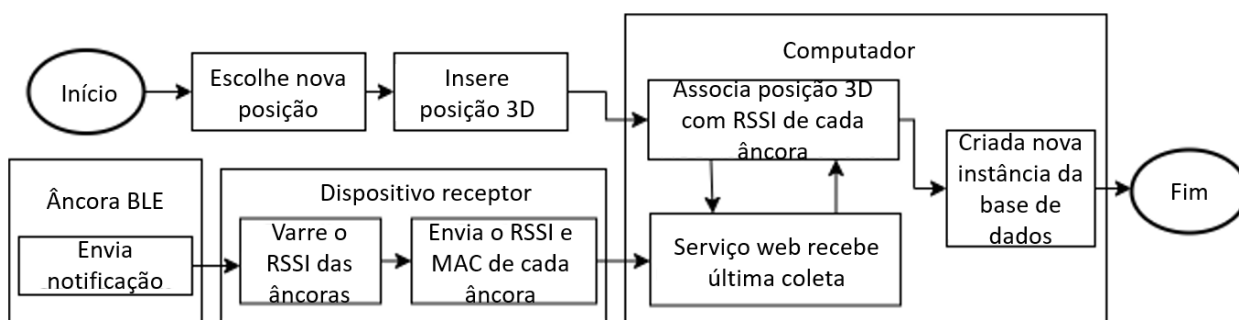
8.6 Algoritmos para coleta de dados

As âncoras foram configuradas para enviar notificações (*advertisements*) a intervalos regulares. O dispositivo móvel receptor foi configurado para ler essas notificações e enviá-las para um computador remoto para registro. Dessa forma, foi necessário o desenvolvimento de algoritmos embarcados em todos os tipos de dispositivos. A figura 23 apresenta o fluxograma das ações distribuídas pelos diferentes dispositivos. Os algoritmos e serviços envolvidos são:

1. **Âncora:** algoritmo para envio de notificações;
2. **Dispositivo móvel:** algoritmo para varredura de dispositivos BLE (*Bluetooth Low Energy*), coleta dos níveis de RSSI das âncoras e associação com o seu respectivo identificador, seu endereço MAC (*Media Access Control*);
3. **Computador:** configuração de servidor web HTTP (*Hyper Text Transfer Protocol*) para recebimento dos dados do cliente (dispositivo móvel); Associação dos dados recebidos com a posição corrente do dispositivo móvel e geração da base de dados.

Estes processos serão melhor detalhados nas próximas seções.

Figura 23 – Fluxo macro para o processo da criação da base de dados de fingerprint.



Fonte: Autor.

8.6.1 Algoritmo para envio de notificações pelas âncoras

O processo de envio de notificações é amplamente utilizado para aplicações no varejo, onde dispositivos BLE tipicamente fazem o envio de notificações para celulares de clientes com informações para publicidade na loja (ALLURWAR; NAWALE; PATEL, 2016). Um processo similar foi aqui adotado com envio regular dessas notificações a cada 1 segundo. O algoritmo utilizado nesse processo pode ser visto no algoritmo 2.

Algoritmo 2: Algoritmo para envio de notificações pelas âncoras

```

1. Configura notificações;
2. Configura intervalo de tempo para envio das notificações;
3. while True do
    | 3.1. Envia notificação;
    | 3.2. Aguarda intervalo;
end
```

8.6.2 Algoritmo para varredura do sinal das âncoras e envio dos dados de leitura

Para que o dispositivo móvel fosse capaz de executar a varredura, coleta de dados das âncoras disponíveis no ambiente e seu envio para o computador remoto, o código apresentado no algoritmo 3 foi desenvolvido.

Algoritmo 3: Algoritmo para varredura do sinal das âncoras e envio dos dados de leitura.

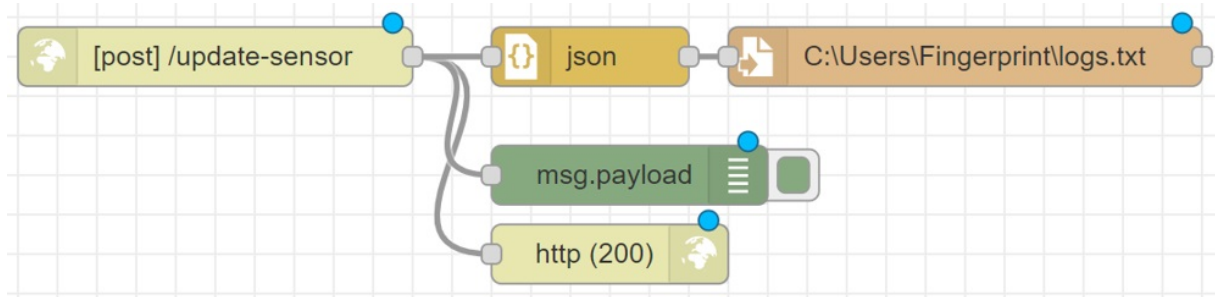
```

1. Configura Service Set Identifier (SSID) da rede Wi-Fi e senha;
2. Configura intervalo de tempo da varredura;
3. while True do
    | 3.1. Realiza varredura de dispositivos BLE;
    | 3.2. for cada dispositivo detectado do
        | 3.2.1. MAC  $\leftarrow$  armazena endereço MAC do dispositivo detectado;
        | 3.2.2. RSSI  $\leftarrow$  armazena RSSI do dispositivo detectado;
        | 3.2.3. Vetor  $\leftarrow$  Armazena endereço MAC e RSSI do dispositivo detectado;
    | end
    | 3.3. Envia Vetor para o servidor web via requisição HTTP POST;
end
```

8.6.3 Servidor web para envio dos dados coletados pelo dispositivo móvel

Para armazenar os dados recebidos dos dispositivos móveis foi utilizado um servidor *web* em um computador configurado com a ferramenta Node-RED (ver seção 9.1.3), *framework* para programação em fluxo baseada em eventos. O fluxo desenvolvido no Node-RED com os blocos de funções selecionados para criação do servidor web pode ser observado na figura 24. Inicialmente utilizou-se um nó para receber uma requisição POST vinda do dispositivo que faz a coleta de dados das âncoras, no caso o vetor com dados dos endereços MAC e respectivos RSSI durante a última varredura. Um segundo nó foi utilizado para converter o vetor recebido para um objeto JSON (*JavaScript Object Notation*), formato de padrão aberto para troca de dados entre sistemas. Para que os dados ficassem disponíveis para consulta do algoritmo responsável por organizar a base de dados, um terceiro nó foi utilizado para registrar o conteúdo do objeto JSON em um arquivo de texto logs (.txt). Foram utilizados dois outros nós: um para exibição em modo *debug* (verificação de cada mensagem trocada entre os blocos de função do Node-RED) dos conteúdos recebidos (msg.payload), e outro para fornecer uma resposta de status OK (200) para o dispositivo que enviou a requisição POST.

Figura 24 – Servidor web implementado na plataforma Node-RED.



Fonte: Autor.

8.6.4 Algoritmo para organização e criação da base de dados

A manipulação dos dados coletados para a criação da base de dados de *fingerprint* exigiu, além da varredura e envio de informações das âncoras no ambiente de testes, também um algoritmo para associação da posição do dispositivo de medição com os níveis de RSSI, que pode ser observado no algoritmo 4. Como a localização do dispositivo de medição tipicamente é alterada diversas vezes pelo usuário entre todas as posições possíveis durante o período de levantamento dos dados, é necessário receber o número da posição corrente de forma manual. Após recebida a posição atual de coleta, este algoritmo consulta o arquivo *logs.txt*, onde obterá os dados referentes às âncoras presentes no ambiente com os seus respectivos endereços MAC e RSSI. Em seguida é realizada a associação da posição corrente com os dados obtidos da última coleta, gerando-se uma nova linha para a base de dados de *fingerprint*.

Algoritmo 4: Algoritmo para organização e criação da base de dados.

```

1. Leituras ← Configura quantidade de leituras por posição;
2. Intervalo ← Configura intervalo entre as leituras em segundos;
3. while True do
    3.1. Posição ← Usuário digita posição atual em uma caixa de texto;
    3.2. for  $t \leftarrow 0$  to Leituras do
        3.2.1. Coleta informações da última varredura de logs.txt;
        3.2.2. for cada registro âncora nas informações de varredura do
            3.2.2.1. if registro de RSSI está vazio then
                3.2.2.1.1. RSSI ← -200;
            end
        end
        3.2.3. Adiciona linha ao banco de dados com a posição atual + registro de RSSI da âncora;
        3.2.4. Espera intervalo;
        3.2.5.  $t \leftarrow t + 1$ 
    end
end

```

Dessa forma a base de dados foi estruturada conforme demonstrado na tabela 4, onde as colunas são constituídas pelas coordenadas $[x, y, z]$ de cada posição do dispositivo receptor, além das n âncoras, identificadas pelo seu endereço MAC. Por outro lado, as linhas trazem as informações das m instâncias coletadas, com as respectivas posições e RSSI detectados de cada âncora.

É importante ressaltar que em alguns ciclos da varredura feita pelo dispositivo de medição, algumas âncoras não foram identificadas por estarem com seu sinal abaixo da sensibilidade mínima do radio BLE do dispositivo embarcado ESP32, de -97dBm (ESPRESSIF, 2021). Para que a base não contivesse registros com espaços vazios devido a essa

ausência de detecção da âncora em determinada instância, de forma similar a outros trabalhos (MOHAMMADI et al., 2017), foi adotado um critério de atribuir o valor específico desprezível de -200 dBm às entradas ausentes.

Tabela 4 – Estrutura da base de dados de localização 3D.

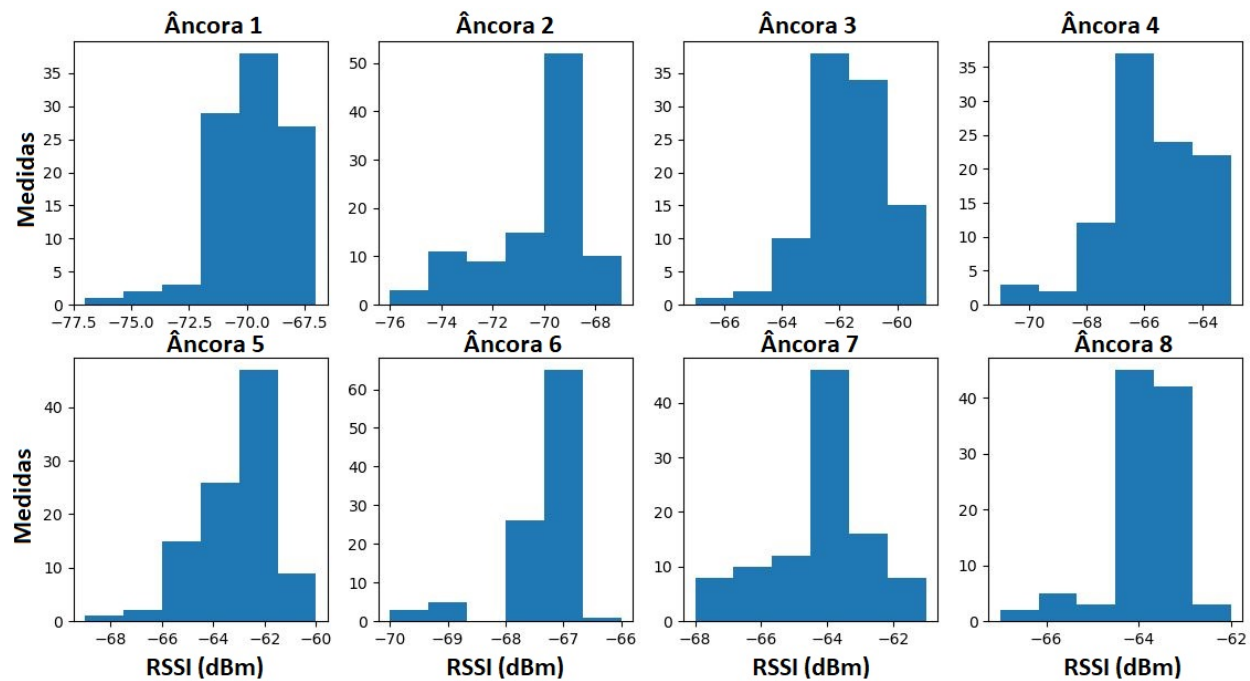
$\mathbf{x}$	$\mathbf{y}$	$\mathbf{z}$	A_1	A_2	...	A_n
x_1	y_1	z_1	$rss\hat{i}_{11}$	$rss\hat{i}_{21}$	...	$rss\hat{i}_{n1}$
x_2	y_2	z_2	$rss\hat{i}_{12}$	$rss\hat{i}_{22}$	...	$rss\hat{i}_{n2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_m	y_m	z_m	$rss\hat{i}_{1m}$	$rss\hat{i}_{2m}$	...	$rss\hat{i}_{nm}$

Fonte: Autor.

8.7 Testes preliminares à criação da base de dados de fingerprint

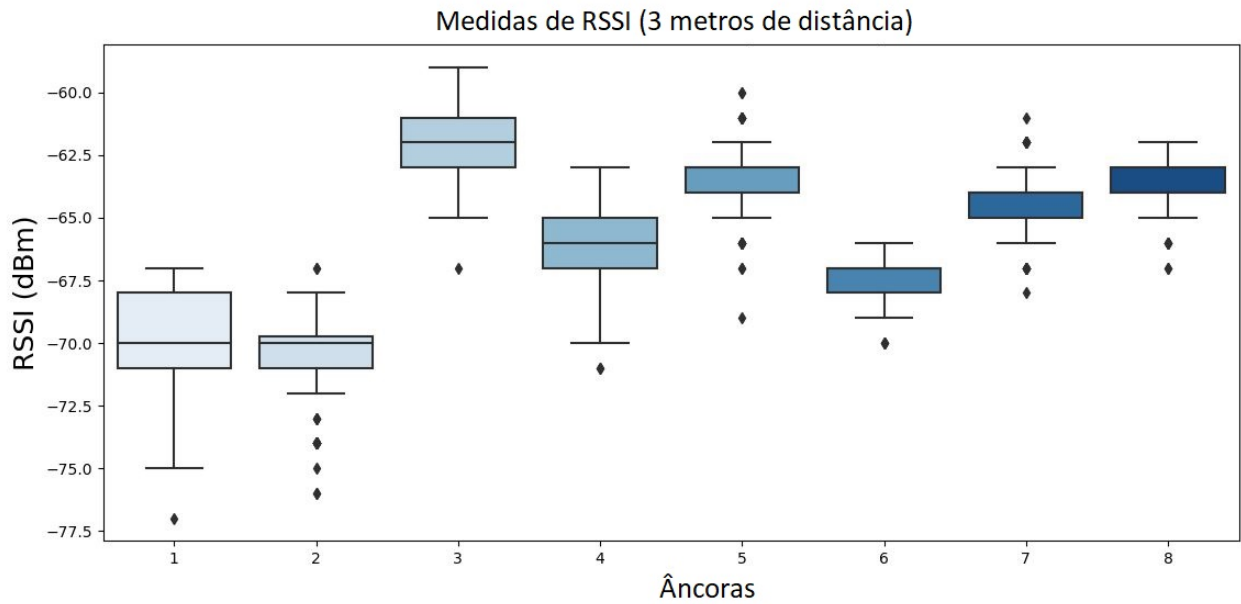
A variação RSSI é uma das questões mais relevantes em sistemas de posicionamento interno baseados em RSSI (MIZI; REGGIANI, 2016). Para uma análise da variância das medidas de RSSI dos dispositivos âncora (configurados conforme descrito na seção 8.6.1), foi executado um teste prévio utilizando o dispositivo de medição e um computador (e os algoritmos descritos em 8.6.2, 8.6.3 e 8.6.4). Nesse teste preliminar foram realizadas 100 coletas de RSSI de cada um dos dispositivos utilizados como âncora, sempre a uma distância fixa de 3 metros do dispositivo empregado para a medição (o dispositivo móvel que se deseja estimar a localização), adotando-se um intervalo de 1 segundo entre cada medição. O resultado gráfico da série de medições pode ser observado na forma de histogramas demonstrados na figura 25 e por meio de uma análise de *boxplot* na figura 26.

Figura 25 – Histogramas das medidas de RSSI dos dispositivos âncoras com distância de 3 metros.



Fonte: Autor.

Figura 26 – Gráfico de boxplot das medidas de RSSI dos dispositivos âncoras com distância de 3 metros.



Fonte: Autor.

Através da observação dos histogramas gerados a partir dos dados dos testes, verificou-se um comportamento próximo ao de uma curva normal nas medições de RSSI, observada a presença de medidas que se diferenciam de forma acentuada das demais, os chamados *outliers*. A respeito dos *outliers*, é possível observar esse evento na figura 25 nos histogramas das âncoras 6. No gráfico de *boxplot* na figura 26 também a presença de *outliers* na medição de todas as âncoras, fato que pode ser atribuído a flutuação da medição de RSSI no decorrer do tempo. Para reduzir a possibilidade da medição de algum *outlier* durante a coleta de dados, e também para possibilitar o uso de métodos estatísticos para reduzir o impacto do erro nas medições, optou-se por realizar múltiplas coletas em cada posição durante o processo de coleta de dados no ambiente escolhido.

De forma empírica, optou-se como procedimento pela adoção de uma sistemática de realizar 5 medições por posição, e um intervalo de 2 segundos entre as coletas. A quantidade de coletas e intervalos escolhidos tomaram como base o tempo corrido necessário para aquisição dos dados, de 10 segundos por posição nos 340 pontos previamente definidos (85 posições em 4 alturas diferentes), com um tempo de aproximadamente 57 minutos corridos, apenas relativos ao período de coleta, desconsiderando intervalos entre as mudanças de localização, manipulação do computador, ajuste do suporte, dentre outros. Uma das preocupações foi a de que o processo de coleta, com imprevistos, retrabalhos e ajustes necessários, estivesse dentro de um turno (período) de trabalho, com o mínimo de interrupção, de forma a manter aproximadamente as mesmas condições ambientais e físicas do local durante todo o tempo de desenvolvimento dos testes (temperatura, umidade, etc.).

Observando o comportamento próximo ao de uma curva normal das medidas do RSSI detectado das âncoras durante os testes preliminares, e com o intuito de reduzir a possibilidade de que *outliers* coletados na base de dados sejam utilizados na implementação dos algoritmos, optou-se por utilizar a técnica da mediana para que, as 5 coletas realizadas por posição – que na base de dados totalizam 1700 instâncias – sejam reduzidas para apenas 1 registro por posição, referente à mediana das 5 coletas – totalizando 340 instâncias na base de dados após o filtro da mediana.

9 Materiais e Métodos

O presente capítulo descreve os materiais e métodos adotados para a investigação da abordagem do **filtro de âncoras**. Esta seção encontra-se organizada da seguinte forma: *i) Materiais:* descreve os principais materiais utilizados (dispositivos de IoT, tecnologias de redes, ambientes computacionais, *hardware* e *software* utilizados nas âncoras e dispositivos móveis); *ii) Métodos:* apresenta as metodologias adotadas (métricas utilizadas para avaliação dos resultados, procedimentos experimentais empregados, detalhes de configuração e avaliação dos algoritmos de Aprendizado de Máquina e correlatos).

9.1 Materiais

9.1.1 Dispositivo IoT

Para implementação do projeto foi utilizada a placa de desenvolvimento DOIT ESP32 DEVKIT V1, baseada no módulo microcontrolador ESP-WROOM-32 da fabricante multinacional de semicondutores Espressif ([ZERYNTH, 2021](#)). O microcontrolador utilizado é considerado um sistema integrado complexo (*System on Chip* - SOC), devido a quantidade de recursos disponíveis, que permitem um sistema autônomo completo, memória *flash*, comunicação USB e conectividade sem fio, empregado em projetos de dispositivos móveis, eletrônicos vestíveis e aplicativos IoT em geral ([ESPRESSIF, 2021](#)).

O suporte a conectividade Wi-Fi permite conexão direta com a internet por meio de um roteador, enquanto com a utilização do *Bluetooth Low Energy* possibilita que a placa de desenvolvimento atue como dispositivo atue como um sensor para captura e envio de dados, ou como *beacon* para sua aplicações de detecção e localização ([ESPRESSIF, 2021](#)). O módulo possui modos de consumo para diferentes fins, de uma corrente nominal de 80 mA até um consumo em repouso inferior a $5\mu\text{A}$, para aplicações que dependem de um gasto de energia ultrabaixo, visando produtos alimentados a bateria com longa duração ([ESPRESSIF, 2021](#)). A placa de desenvolvimento DOIT ESP32 DEVKIT V1 pode ser verificada na figura 27, e as especificações mais importantes do módulo principais características são listadas a seguir:

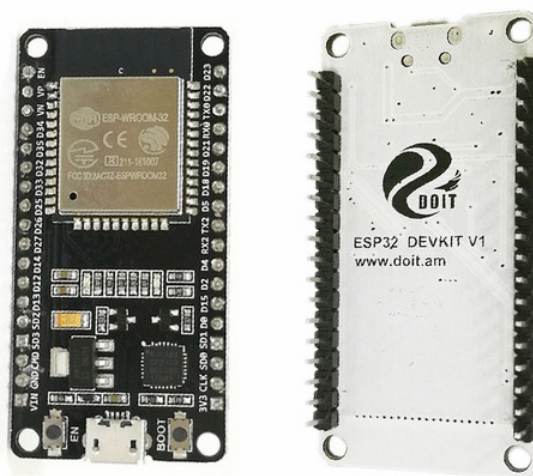
- Protocolo Wi-Fi 802.11 b/g/n (802.11n - 150 Mbps)
- Protocolo *Bluetooth* v4.2 BR/EDR e *Bluetooth Low Energy*
- Sensibilidade do rádio BLE (-97 dBm)
- Corrente operacional média - 80mA, Sleep Mode $<5\mu\text{A}$
- Frequência de *clock* 240MHz, SPI Flash 4MB, RAM 320 kB

Foram utilizados neste projeto 9 placas de desenvolvimentos iguais, 8 delas como dispositivos fixos de referência, as âncoras, e 1 para coleta dos dados.

9.1.2 Hardware para organização dos dados e implementação de algoritmos

Para armazenamento dos dados coletados durante a etapa de *fingerprint*, treinamento e execução dos algoritmos foi utilizado um *laptop* com as seguintes especificações:

Figura 27 – Plataforma de desenvolvimento - DOIT ESP32 DEVKIT V1.



Fonte: (ZERYNTH, 2021).

- CPU: Intel® Core™ i5-8250U CPU - 1,80 GHz
- Memória RAM - 8 GB

9.1.3 Ferramentas e linguagens de programação

Conforme apresentado na seção 9.1.1, o elemento responsável tanto pelo processo de envio de dados de identificação, tarefa básica das âncoras, quanto pela coleta dos dados em campo, é uma placa de desenvolvimento baseada no módulo microcontrolador ESP-WROOM32. Para realização destas funções, esse dispositivo requer uma programação com algoritmos especificamente desenvolvidos para essas tarefas. Uma linguagem usualmente adotada no âmbito dos sistemas embarcados é o C++, que possibilita o uso de estruturas de programação orientada a objetos, conferindo maior capacidade de abstração e de reusabilidade de código (SCHNEEBELI, 2007). O C++ foi a linguagem adotada no âmbito do presente trabalho, com ambiente de desenvolvimento integrado (*Integrated Development Environment - IDE*) disponibilizado pela plataforma de código aberto Arduino (ARDUINO, 2021).

Para recebimento dos dados coletados foi configurado um servidor *web* utilizando a ferramenta Node-RED, um *framework* para programação em fluxo baseada em eventos, popular em aplicações de *Internet of Things*, conexões de dispositivos de *hardware*, serviços *web*, entre outros (NODE-RED, 2021). A ferramenta Node-RED dispõe de uma plataforma de criação de alto nível, com um editor baseado em navegador onde as aplicações são criadas através da ligação de nós (ou blocos) com funções específicas, como HTTP, TCP, MQTT, *websocket* e outras (NODE-RED, 2021).

A linguagem Python foi utilizada para criação e manipulação de bases de dados, particularmente fazendo uso da biblioteca de código aberto Pandas (PANDAS, 2021), que fornece soluções para o trabalho com dados estruturados, tabulares e multidimensionais em documentos de extensões diversas (csv, txt, xls, entre outras). Outra biblioteca utilizada foi o Numpy (NUMPY, 2021) para a realização de cálculos e organização de vetores e matrizes multidimensionais. Para a implementação dos algoritmos de Aprendizado de Máquina foi utilizada a biblioteca de código aberto para linguagem Python scikit-learn (SCIKIT-LEARN, 2021). Essa biblioteca inclui diversos algoritmos de classificação, regressão, agrupamento, redução de dimensionalidade, comparação e validação de modelos, entre outros.

9.2 Métodos

9.2.1 Métricas para avaliação

9.2.1.1 Raiz quadrada do erro médio - RMSE

Uma métrica básica utilizada para erro em aplicações de localização é a raiz quadrada do erro médio (em inglês *Root Mean Square Error* - RMSE). O RMSE pode ser aplicado para comparação entre os valores reais observados e predições ou hipóteses, no caso de aplicações de localização podem ser desde as distâncias de pontos de referência, ou coordenadas cartesianas (x, y) . Para aplicações de localização 3D, como a proposta deste trabalho o cálculo do RMSE será feito conforme a equação 14, em que as coordenadas de localização reais serão representadas por (x, y, z) e as estimadas pelos algoritmos utilizados como $(\hat{x}, \hat{y}, \hat{z})$. Para uma análise mais detalhada, a métrica de RMSE também será utilizada para verificação do erro por eixo.

$$Erro = \sqrt{\frac{\sum_{i=1}^n (x(i) - \hat{x}(i))^2 + (y(i) - \hat{y}(i))^2 + (z(i) - \hat{z}(i))^2}{n}} \quad (14)$$

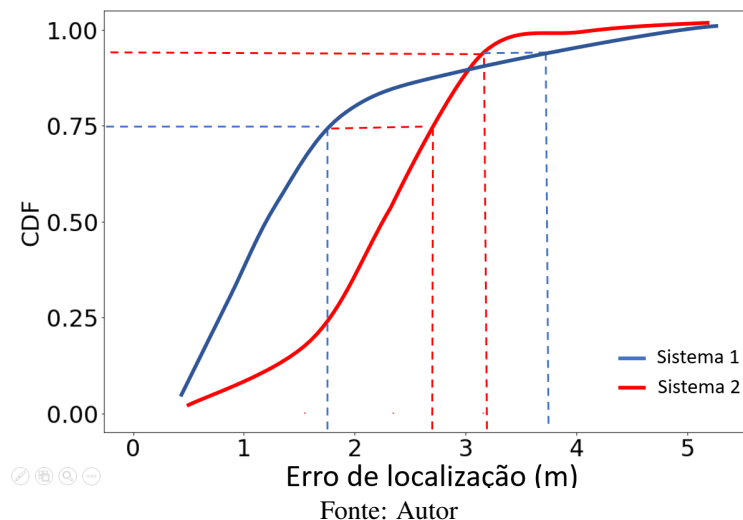
9.2.1.2 Função de distribuição acumulada - CDF

No contexto da estatística, a distribuição de probabilidade relaciona um determinado valor de uma variável investigada com a sua probabilidade de ocorrência. A função de distribuição acumulada (*Cumulative Distribution Function* - CDF) por sua vez trata da probabilidade da variável estudada ser igual, ou menor a um determinado valor. Aplicações de localização utilizam como métrica a função de distribuição acumulada para investigar e comparar a performance de diferentes abordagens, fixando alguma faixa de erro e comparando a probabilidade acumulada de cada algoritmo, ou comparando o valor de erro para determinadas parcelas da função acumulada de probabilidade, particionadas em percentis ou quartis (25%). Enquanto a métrica de RMSE trata apenas da acurácia de um sistema de localização, a CDF possibilita uma análise do incremento do erro a partir da função acumulada, permitindo uma avaliação da precisão do sistema.

Na comparação entre duas técnicas de localização, se a acurácia dos sistemas é a mesma ou muito próxima (obtida pelo erro médio), com a utilização da CDF de erro de ambas, é preferido o sistema em que a curva atinge valores de alta probabilidade com um menor aumento do erro, uma vez que os seus desvios de distâncias estão concentrados em valores menores (LIU et al., 2007). De forma prática, a análise através da CDF é descrita pelo formato de percentis (medidas que dividem a amostra em 100 partes) ou quartis (amostra dividida em 4 partes), e a partir dela é possível comparar a precisão em intervalos da curva cumulativa do erro de localização.

A título de exemplo, consideremos a curva de um determinado sistema que denominaremos de **sistema 1** (figura 28, curva em azul), o qual representa o erro acumulado do mesmo. No gráfico observa-se que 75% das instâncias tem um erro de localização menor ou igual à 1.8 m (o CDF do erro de distância de 2.3 m é 0.9) e 95% menor ou igual à 3.8 m. Um segundo sistema de localização (**sistema 2**, curva em vermelho) tem, considerando o seu erro acumulado, 75% das instâncias com erro menor ou igual à 2.7m e 95% menor ou igual à 3.2m. Utilizando a métrica da CDF de erro de localização é possível optar pelo primeiro sistema (curva em azul) devido ao seu erro de distância estar mais concentrado em valores menores (LIU et al., 2007). Para avaliação da precisão do sistema será adotado como critério a verificação da acurácia do sistema no segundo e terceiro quartis, referentes a 50% e 75% da função de erro acumulado.

Figura 28 – Função de distribuição acumulada (CDF) de erro para 2 sistemas de localização.



9.2.2 Procedimento Experimental

A partir da abordagem proposta foram elaborados 3 (três) experimentos distintos utilizando algoritmos supervisionados do Aprendizado de Máquina para a tarefa de localização em ambientes internos:

1. **EXP-01:** K-vizinhos mais próximos (KNN);
2. **EXP-02:** Árvore de regressão, e
3. **EXP-03:** Rede Neural Artificial (RNA) do tipo perceptron multicamadas (MLP).

As seções a seguir apresentam em detalhes os procedimentos adotados em cada caso, destacando a arquitetura e parametrização dos algoritmos. Além dos algoritmos propriamente ditos, os experimentos buscaram investigar o impacto da utilização da estratégia do **filtro de âncoras** em cada caso, considerando-se o ambiente experimental (descrito na seção 8) que apresentou disponibilidade simultânea de até 8 âncoras.

9.2.2.1 EXP-01: K-vizinhos mais próximos (KNN)

Dados. A base de dados levantada conforme os procedimentos descritos no capítulo 8 foi particionada em conjuntos de treinamento e testes, adotando a seguinte estratégia: os exemplos disponíveis foram aleatoriamente selecionados para cada conjunto obedecendo à proporção 3:1 para treinamento e teste. **Tarefa.** O algoritmo do KNN é tipicamente utilizado em aplicações de classificação, ou seja, para agrupar uma instância aos vizinhos mais similares a ela observando os k vizinhos mais próximos. Na presente aplicação de localização, o algoritmo foi configurado para atuar como ferramenta de regressão. **Entradas.** O sistema foi configurado com $4 \times n$ entradas, com n sendo o número de âncoras consideradas, com as 4 entradas em cada caso recebendo os valores de potência (RSSI), e posição (x, y, z) de cada âncora. **Saídas.** O sistema foi configurado com 3 saídas indicativas da posição estimada do objeto (x,y,z). As saídas do algoritmo foram configuradas para assumir a média ponderada das coordenadas de posicionamento dos k vizinhos mais próximos, conforme apresentado na seção 5.1. A ponderação foi proporcional à distância euclidiana entre as instâncias. **Treinamento.** O KNN não possui fase de treinamento, uma vez que não elabora modelo. Os dados disponíveis para treinamento foram armazenados para uso quando da apresentação de novas instâncias. **Parâmetros**

observados. O experimento foi repetido variando-se: a) o parâmetro k para investigar o desempenho do algoritmo KNN na tarefa de localização em função da quantidade de vizinhos adotada (de 1 a 14 vizinhos); b) o parâmetro n para investigar o desempenho do algoritmo de localização em função do número de âncoras utilizado, de acordo com a abordagem do filtro de âncoras, ou seja, considerando-se apenas as n âncoras de maior potência (RSSI), com variação de 1 a 8 âncoras (número máximo disponível). **Validação.** Após a conclusão do treinamento, as instâncias de teste foram submetidas ao algoritmo, observando-se o RMSE em cada caso.

O algoritmo aqui apresentado foi comparado com a abordagem clássica para a tarefa da localização conforme visto no trabalho de (SADOWSKI; SPACHOS; PLATANIOTIS, 2020), na qual o KNN é configurado com 8 dimensões de entrada, que recebem diretamente as potências (RSSI) das 8 âncoras.

9.2.2.2 EXP-02: Árvore de regressão

Dados. A base de dados levantada conforme os procedimentos descritos no capítulo 8 foi particionada em conjuntos de treinamento e testes, adotando a seguinte estratégia: os exemplos disponíveis foram aleatoriamente selecionados para cada conjunto obedecendo à proporção 3:1 para treinamento e teste. **Tarefa.** Para desempenhar a tarefa de regressão, o parâmetro necessário para determinar a arquitetura da árvore de regressão é a profundidade da árvore, ou seja, o comprimento do caminho mais longo referente aos níveis desde a raiz até a última folha. Assim sendo, foram observados o melhor parâmetro para a árvore de regressão para a tarefa de localização. **Entradas.** O sistema foi configurado com $4 \times n$ entradas, com n sendo o número de âncoras consideradas, com as 4 entradas em cada caso recebendo os valores de potência (RSSI), e posição (x, y, z) de cada âncora. **Saídas.** O sistema foi configurado com 3 saídas indicativas da posição estimada do objeto (x,y,z). Como critério para medir a qualidade de uma determinada divisão da árvore foi utilizado o erro quadrático médio, escolhendo a divisão com menor erro de localização. **Parâmetros observados.** O experimento foi repetido variando-se: a) o parâmetro d para investigar o desempenho do algoritmo de árvore de regressão na tarefa de localização em função da sua profundidade (de 1 a 10 níveis); b) o parâmetro n para investigar o desempenho do algoritmo de localização em função do número de âncoras utilizado, de acordo com a abordagem do filtro de âncoras, ou seja, considerando-se apenas as n âncoras de maior potência (RSSI), com variação de 1 a 8 âncoras (número máximo disponível). **Validação.** Após a conclusão do treinamento, as instâncias de teste foram submetidas ao algoritmo, observando-se o RMSE em cada caso.

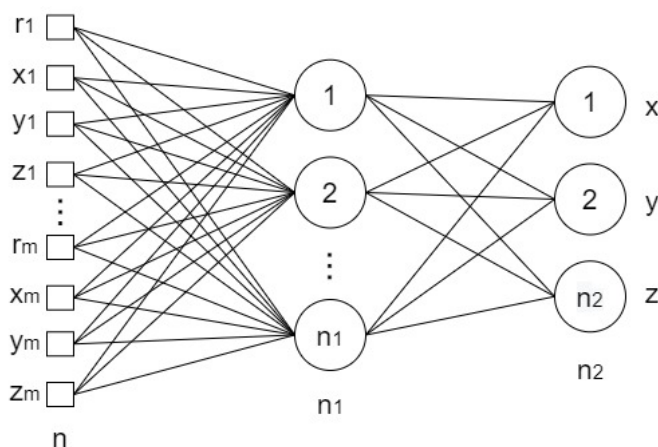
O algoritmo aqui apresentado foi comparado com a abordagem clássica para a tarefa da localização, conforme visto no trabalho de Ahmadi e Bouallegue (2015), na qual a árvore de regressão é configurada com 8 dimensões de entrada, que recebem diretamente as potências (RSSI) das 8 âncoras.

9.2.2.3 EXP-03: Rede Neural Artificial (RNA)

Dados. A base de dados levantada conforme os procedimentos descritos no capítulo 8 foi particionada em conjuntos de treinamento e testes, adotando a seguinte estratégia: os exemplos disponíveis foram aleatoriamente selecionados para cada conjunto obedecendo à proporção 3:1 para treinamento e teste. **Tarefa.** A RNA foi configurada também para a tarefa da regressão, com saídas numéricas representando a posição estimada do dispositivo no ambiente. **Arquitetura.** A RNA utilizada foi do tipo MLP configurada com camada de entrada, uma única camada oculta e camada de saída. A arquitetura da rede implementada é apresentada esquematicamente na figura 29. **Camada de entrada.** A camada de entrada foi configurada com $4 \times n$ neurônios, com n sendo o número de âncoras a serem consideradas, com as 4 entradas em cada caso recebendo os valores de potência (RSSI) e posição (x,y,z) de cada âncora. Os neurônios utilizaram função de ativação TANH. **Camada oculta.** A rede foi configurada com apenas uma camada oculta com n_1 neurônios, e o desempenho da rede foi avaliado para diferentes valores de n_1 . Os neurônios utilizaram função de ativação TANH. **Camada de saída.** A rede foi configurada com 3 neurônios de saída para estimação da localização do dispositivo no

ambiente (x,y,z) . Os neurônios dessa camada utilizaram função de ativação linear para atenderem à tarefa da regressão. **Treinamento.** A rede neural foi treinada com o algoritmo da retropropagação do erro. **Taxa de aprendizado.** A taxa de aprendizado foi constante e igual a 0,001. **Critério de parada.** Número máximo de iterações igual a 10.000. **Parâmetros observados.** O experimento foi repetido variando-se: a) o parâmetro n_1 para investigar o desempenho da RNA com a variação do número de neurônios na camada oculta (de 10 a 64 neurônios); b) o parâmetro n para investigar o desempenho do algoritmo de localização em função do número de âncoras utilizado, de acordo com a abordagem do filtro de âncoras, ou seja, considerando-se apenas as n âncoras de maior potência (RSSI), com variação de 1 a 8 âncoras (número máximo disponível). **Validação.** Após a conclusão do treinamento, as instâncias de teste foram submetidas ao algoritmo, observando-se o RMSE em cada caso.

Figura 29 – Arquitetura da rede neural perceptron multicamadas utilizada.



Fonte: Autor.

O algoritmo aqui apresentado foi comparado com a abordagem clássica para a tarefa da localização, na qual a RNA é configurada com 8 neurônios de entrada, que recebem diretamente as potências (RSSI) das 8 âncoras.

9.3 Comparação das abordagens

Como forma de promover uma análise comparativa entre os algoritmos, uma avaliação dos mesmos utilizando a função de distribuição acumulada (CDF), abordada na seção 9.2.1.2 será realizada. Considerando que no ambiente onde os experimentos foram realizados, conforme descrito na seção 8.4, a maioria dos obstáculos presentes é composta por bancadas com altura de 1.35m, é possível supor que as medições abaixo desse nível – falta de linha de visata (NLOS) – possam estar sujeitas a condições mais adversas do que as medições realizadas acima da altura das bancadas, em situação de linha de visada direta (LOS). Dessa forma, e considerando que não foram encontradas na literatura outras análises em 3D, algumas investigações foram conduzidas também observando-se o desempenho das abordagens em diferentes níveis de altura (eixo z).

10 Resultados e Discussão

O presente capítulo destaca os resultados obtidos para os diferentes experimentos realizados no contexto da localização em ambientes internos – referente aos procedimentos detalhados na seção 9.2.2 – e apresenta as discussões pertinentes.

10.1 EXP-01: K-vizinhos próximos (KNN)

As figuras 30 e 31 apresentam respectivamente o desempenho do algoritmo para as abordagens filtro de âncoras e clássica.

Figura 30 – Raiz do Erro Quadrático Médio (RMSE) x número k de vizinhos (KNN) - Filtro de âncoras.

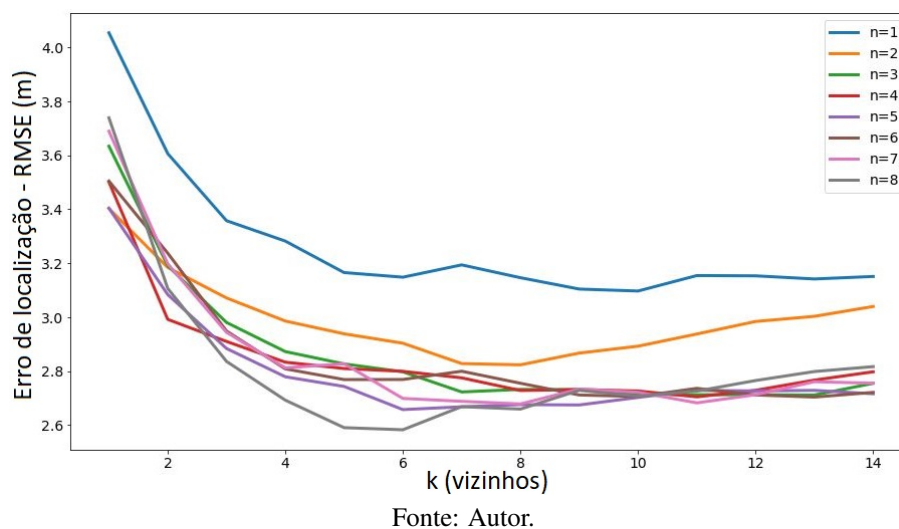
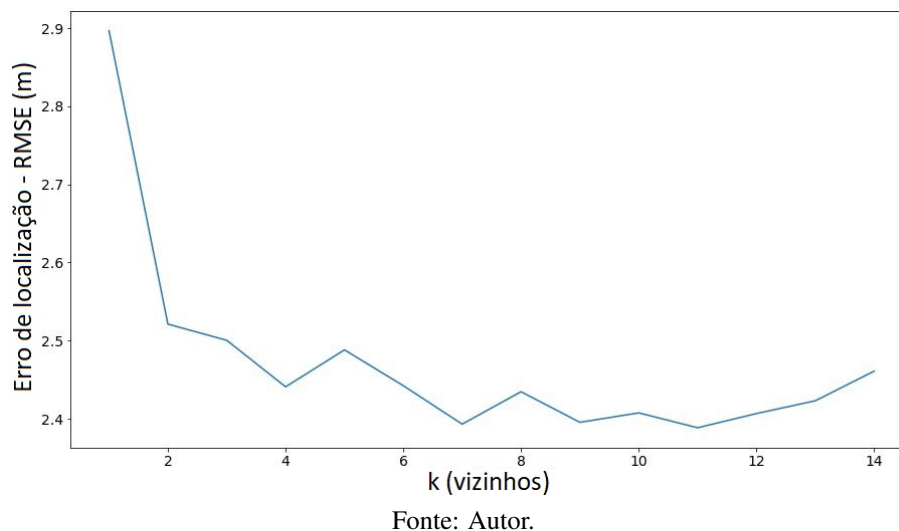


Figura 31 – Raiz do Erro Quadrático Médio (RMSE) x número k de vizinhos (KNN) - Abordagem clássica.

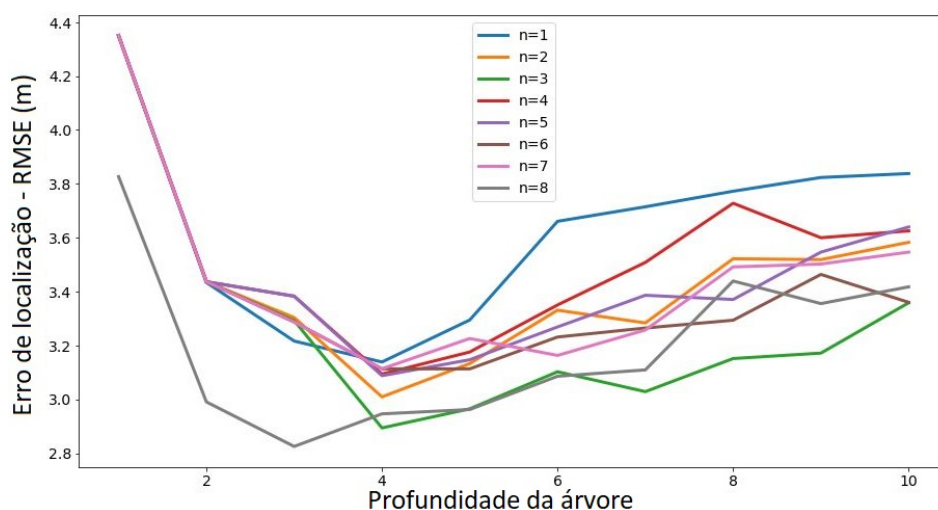


As curvas apresentadas no gráfico do filtro de âncoras destacam o erro de localização 3D médio (RMSE) em função do número de vizinhos para os 8 possíveis cenários de filtro. Com o filtro de $n = 8$ foi encontrado o menor erro médio (RMSE=2.58m) com um $k = 6$ vizinhos, valor que será adotado para os testes comparativos. Com o uso do algoritmo de KNN na abordagem clássica, o menor resultado de erro de localização média foi encontrado com um número de vizinhos $k = 7$, obtendo um RMSE de 2.39m. Relevante destacar nesse cenário que para $n \geq 3$ âncoras, o erro obtido pelo algoritmo do filtro de âncoras é sempre bastante semelhante, o que sugere esse como um número mínimo de âncoras para um bom resultado da abordagem. Utilizando-se, então, apenas 3 (das 8) âncoras para realizar o processo de localização, o erro da abordagem foi apenas ligeiramente maior do que quando todas foram utilizadas.

10.2 EXP-02: Árvore de regressão

Os gráficos apresentados nas figuras 32 e 33 trazem respectivamente o erro de localização 3D em função do número de níveis definido para a árvore de regressão para a abordagem do filtro de âncoras e para a abordagem clássica.

Figura 32 – Raiz do Erro Quadrático Médio (RMSE) x profundidade da árvore de regressão - Filtro de âncoras.



Fonte: Autor.

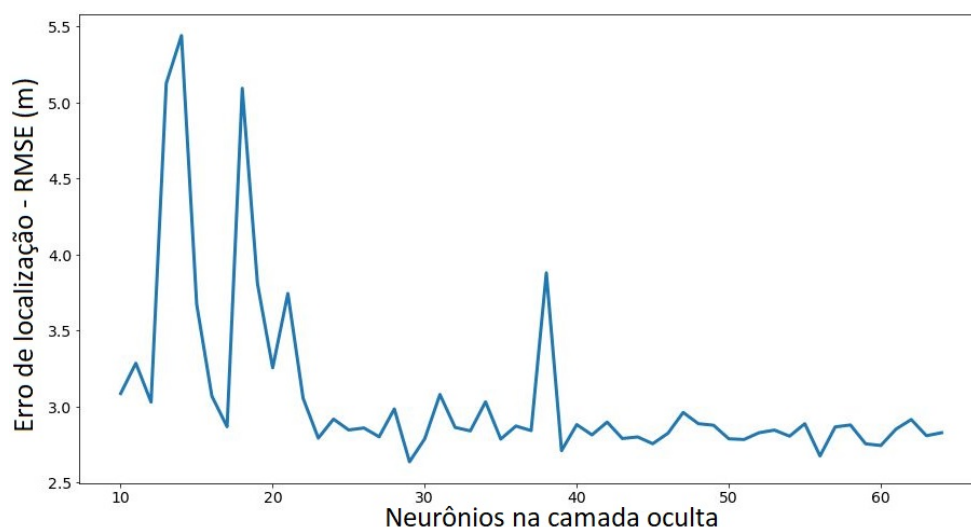
Para a abordagem do filtro de âncoras, o menor erro de localização médio é encontrado com uma profundidade de 3 níveis, e um filtro de $n = 8$, obtendo um RMSE de 2.82m, enquanto na utilização do algoritmo de árvore de regressão com as entradas a partir das coletas de RSSI da base de dados (figura 33), o melhor desempenho é obtido com a mesma quantidade de níveis (3), encontrando um erro médio de localização 3D de 3.21m. Também relevante observar que a abordagem de filtro de âncoras encontra erros menores do que a abordagem clássica para filtros com $n = 2, 3, 4, 5, 6, 7$, além da configuração de 3 níveis para $n = 8$, citada anteriormente, que teve o menor erro médio, sendo que, as outras arquiteturas de árvore de regressão encontram melhores desempenhos com 4 níveis de profundidade.

10.3 EXP-03: Rede Neural Artificial (RNA)

As figuras 34 e 35 apresentam os gráficos obtidos respectivamente para a abordagem do filtro de âncoras e clássica.

As figuras 34 e 35 destacam em cada caso o erro de localização 3D médio em função do número de neurônios na camada oculta da rede MLP. Tipicamente, para um número reduzido de neurônios na camada oculta a rede apresenta

Figura 35 – Raiz do Erro Quadrático Médio (RMSE) x neurônios na camada oculta da rede MLP - Abordagem clássica.



Fonte: Autor.

10.4 Influência da altura

A tabela 5 apresenta os erros de localização obtidos considerando-se a divisão das alturas em níveis **inferior** ($0m \leq z \leq 1m$) e **superior** ($2m \leq z \leq 3m$), além do caso **geral** ($0m \leq z \leq 3m$).

Tabela 5 – Resumo dos erros de localização encontrados nos cenário propostos.

Algoritmo	Erro (m)											
	Parte inferior				Parte superior				Geral			
	x	y	z	RMSE	x	y	z	RMSE	x	y	z	RMSE
MLP (filtro)	1.70	1.55	0.40	2.62	1.83	1.67	0.45	2.84	1.35	1.48	0.94	2.49
Árvore (filtro)	2.13	1.37	0.40	2.89	2.30	1.39	0.45	3.00	1.79	1.49	0.95	2.82
KNN (filtro)	1.90	1.47	0.32	2.69	1.77	1.53	0.42	2.60	1.64	1.65	0.72	2.75
MLP	2.17	1.47	0.43	2.93	1.73	1.50	0.41	2.57	1.49	1.49	0.94	2.62
Árvore	2.06	1.48	0.46	2.90	2.35	1.44	0.45	3.13	2.17	1.55	0.94	3.21
KNN	1.60	1.50	0.25	2.55	1.32	1.43	0.39	2.20	1.34	1.47	0.71	2.39
	1.92	1.47	0.37	2.76	1.88	1.49	0.42	2.72	1.63	1.52	0.86	2.71

Fonte: Autor.

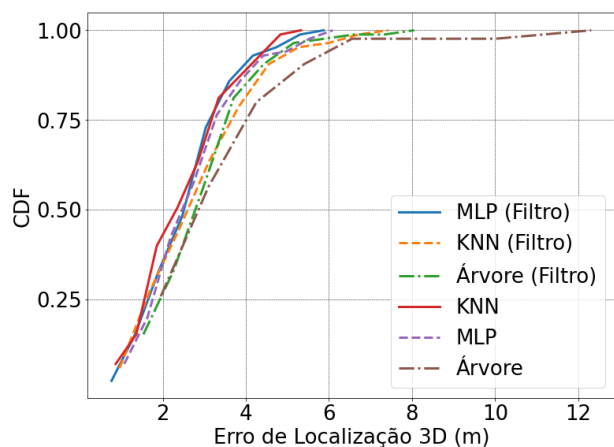
Considerando os RMSE médios obtidos pelos diferentes algoritmos em cada uma das diferentes faixas (inferior, superior e geral), não fica evidenciada nenhuma influência significativa das alturas no desempenho geral dos algoritmos de localização como poderia-se esperar em um primeiro momento em decorrência de possíveis interferências. O menor erro médio de localização foi obtido com o algoritmo KNN na abordagem clássica, com um RMSE de 2,39m, que também obteve os menores erros para os cenários de localização na parte inferior e superior do ambiente. O melhor desempenho de localização entre os algoritmos aplicados com a abordagem de filtro de âncoras foi obtido com o MLP com um erro geral de 2.49m.

10.5 Comparativo das abordagens

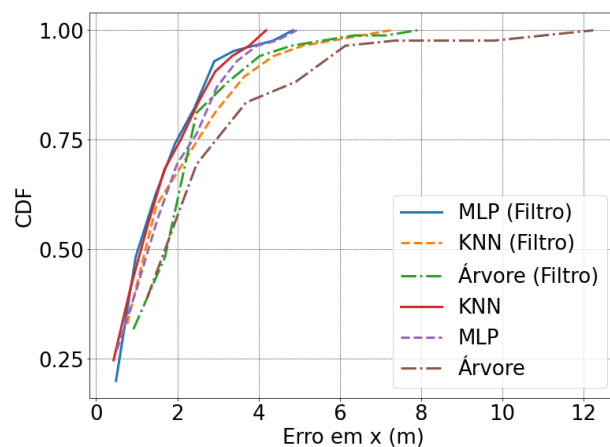
As figuras 36 e 37 apresentam os gráficos da função de distribuição acumulada (CDF) – discutida na seção 9.2.1.2 – para os diferentes algoritmos aplicados sobre os dados de todo o ambiente de testes e das partes superior e inferior do eixo z. Para as análises aqui apresentadas, adotamos como parâmetro o erro acumulado dos algoritmos no segundo e terceiro quartis, referentes a 50% e 75% da função de erro acumulado.

A figura 36a, por exemplo, permite observar os 2 algoritmos de menor RMSE, já destacados nas seções anteriores: o KNN (abordagem clássica) e o MLP (abordagem do filtro de âncoras). Observados pela métrica do CDF, verifica-se que o KNN obtém melhor desempenho (menor erro acumulado) no quartil de 50%, com 2.30m, comparado a 2.50m do MLP, e no quartil de 75% é o MLP que tem menor erro acumulado com 3.10m, frente a 3.18m do KNN. As demais curvas apresentam o erro acumulado também por eixo (x, y e z), corroborando os resultados obtidos nas seções anteriores.

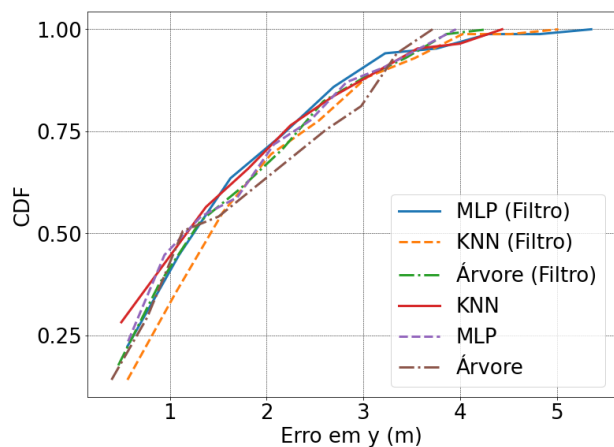
Figura 36 – Gráficos da função de distribuição acumulada (CDF) do erro de localização - parte 1.



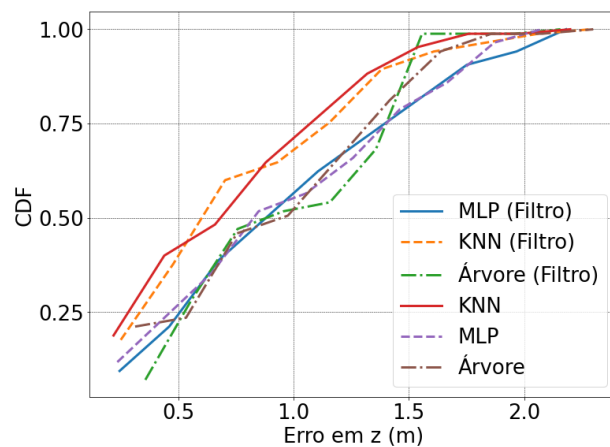
a) CDF - Erro de localização 3D- cenário geral.



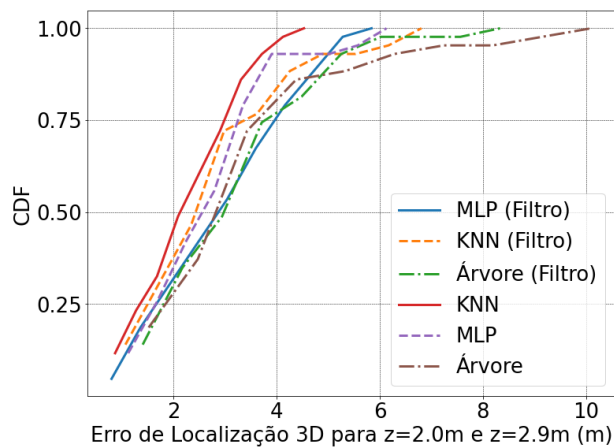
b) CDF - Erro em x - cenário geral.



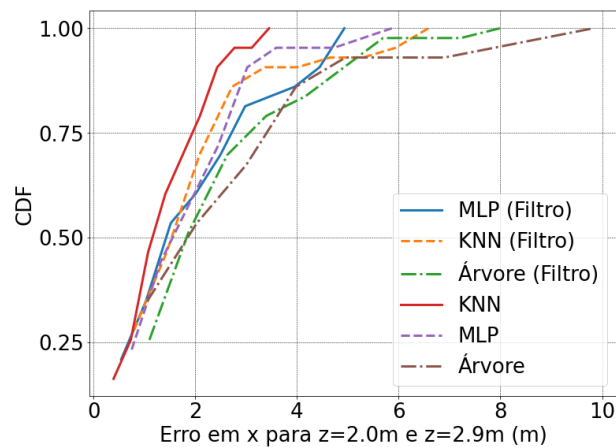
c) CDF - Erro em y - cenário geral.



d) CDF - Erro em z - cenário geral.



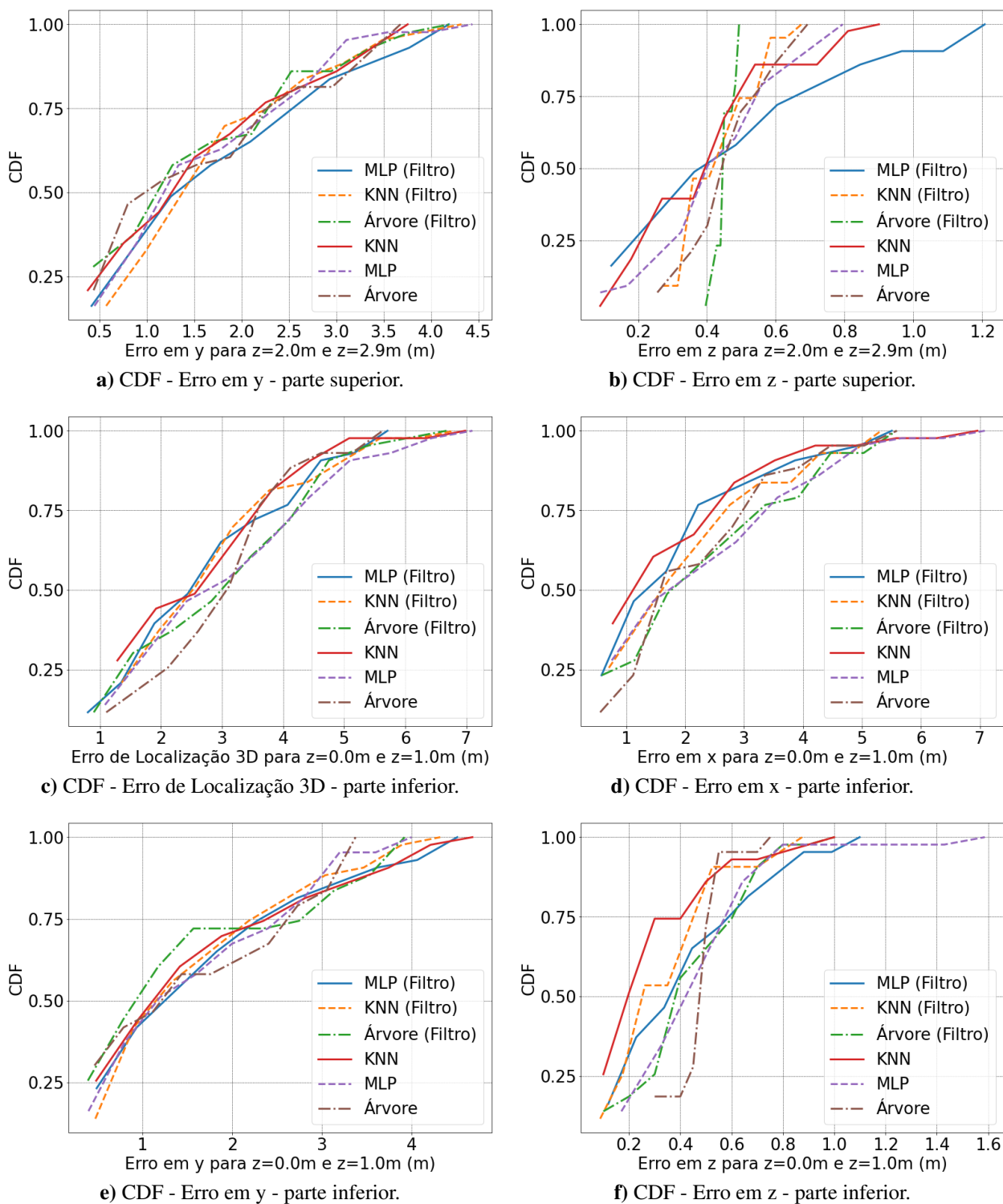
e) CDF - Erro de localização 3D- parte superior.



f) CDF - Erro em x - parte superior.

Fonte: Autor.

Figura 37 – Gráficos da função de distribuição acumulada (CDF) do erro de localização - parte 2.



Fonte: Autor.

11 Conclusões

O presente trabalho propôs uma estratégia denominada de **filtro de âncoras**, que pode ser aplicada de forma conjunta com algoritmos do âmbito do Aprendizado de Máquina, no contexto da localização 3D em ambientes internos com redes de sensores sem fio (IoT).

O trabalho inicialmente realizou uma revisão das principais tecnologias de redes de sensores sem fio para localização em ambientes internos, com uma análise da viabilidade de aplicação de cada uma, e uma comparação de desempenho entre as mesmas. As principais técnicas de medição de distância baseadas no uso de redes de sensores sem fio também foram discutidas. Verificou-se que o nível de potência recebido (RSSI), apesar das dificuldades encontradas em sua utilização devido a flutuações e multipercursos, ainda é extensivamente empregado em aplicações de localização em ambientes internos devido a sua simplicidade de uso, e disponibilidade desse recurso em praticamente todos os dispositivos sem fio.

A principal hipótese levantada no trabalho foi a de que, em abordagens baseadas em *fingerprint*, onde as leituras realizadas das âncoras no ambiente funcionam como atributos de entrada para os algoritmos de Aprendizado de Máquina, a utilização de apenas uma fração dessas leituras – ao invés de sua totalidade – já seria suficiente para garantir um bom desempenho dos sistemas de localização, podendo outra parcela dos dados recebidos (tipicamente os sinais das âncoras de menor potência RSSI) ser desconsiderada sem prejuízos significativos aos algoritmos.

Para dar suporte à investigação experimental desta hipótese, uma nova base de dados em um ambiente interno foi concebida e produzida nas dependências do Grupo de Automação e Sistemas Integráveis (GASI) do Instituto de Ciência e Tecnologia (ICT) do Campus de Sorocaba da Unesp. Nas revisões realizadas, nenhuma base de dados similar foi encontrada, tratando-se potencialmente, portanto, de um dos primeiros – senão o primeiro – *dataset* em três dimensões gerado para essa finalidade, com a coleta de sinais de 8 (oito) âncoras simultâneas em 85 pontos de medição em 4 alturas distintas nesse ambiente. A base de dados para este trabalho foi construída num local com cerca de 50% das instâncias do ambiente praticamente sem obstáculos e linha de visada livre (LOS) – correspondente aos pontos coletados com alturas de 2.0m e 2.9m – e a outra metade do total de registros com pontos coletados em ambiente com presença de bancadas, cadeiras e equipamentos, gerando condições de falta de linha de visada (NLOS) – pontos coletados com $z=0.0m$ e $z=1.0m$.

Considerando o ambiente escolhido e a base de dados levantada, a estratégia do **filtro de âncoras** foi investigada experimentalmente na tarefa da localização em ambientes internos em conjunto com três algoritmos de Aprendizado de Máquinas distintos: KNN, árvore de regressão e redes neurais artificiais (MLP). No contexto do cenário e ambiente escolhidos, os resultados obtidos – em termos do erro médio quadrático na tarefa de localização no ambiente interno – em dois dos casos (árvore de regressão e MLP) superaram aqueles obtidos utilizando os mesmos algoritmos com abordagens clássicas, ou seja, considerando como atributos de entrada os níveis de sinais coletados de todas as âncoras. No terceiro caso (KNN), embora o desempenho do algoritmo clássico na tarefa de localização não tenha sido superado, o desempenho obtido foi bastante próximo. Os experimentos, portanto, não apenas corroboraram a hipótese geral do trabalho mas apontaram na direção de que a utilização da estratégia do filtro de âncoras com apenas 3 (três) âncoras é, via de regra, capaz de levar os sistemas de localização a um desempenho satisfatório. Esse tipo de estratégia, portanto, poderia ser utilizado em ambientes de IoT até mesmo por dispositivos de menor complexidade tecnológica e/ou com menor consumo de energia.

No contexto da base de dados ambiente escolhido, cerca de 38% das instâncias tem pelo menos 1 âncora não detectada. O laboratório utilizado neste trabalho para levantamento da base de dados é composto por um ambiente único, sem paredes ou mais de um cômodo fechado, então é possível inferir que para ambientes mais complexos, de maio-

res dimensões a tendência é de que a taxa de não detecção de âncoras aumente. Dessa forma, a abordagem proposta proporciona um tratamento adequado para os atributos de entrada da base de dados de *fingerprint*, principalmente considerando a não detecção de âncoras, uma vez que os dispositivos que tenham o sinal deteriorado devido a obstáculos ou a distância entre os dispositivos serão desconsiderados por ter menor potência de sinal detectada.

Com base nos resultados obtidos a partir do tratamento dos atributos de entrada com a abordagem de **filtro de âncoras**, o melhor resultado é encontrado com a utilização do algoritmo de redes neurais (MLP), dessa forma, recomenda-se a aplicação deste algoritmo dentre as 3 técnicas de Aprendizado de Máquina escolhidas. Cabe ressaltar de que, apesar do melhor resultado de acurácia – menor erro médio de localização – entre todos os cenários ser encontrado com o algoritmo de KNN a partir da abordagem tradicional, o algoritmo de K-vizinhos mais próximos depende de uma consulta frequente da base de dados, diferentemente das outras técnicas, que treinam modelos de aprendizado. Essa característica do algoritmo de KNN pode se tornar uma dificuldade – em questões de processamento e armazenamento – conforme a base de dados de localização é ampliada, considerando ambientes maiores e mais complexos.

11.1 Contribuições

Dentre as principais contribuições do presente trabalho, destacamos:

- A criação de uma base de dados especificamente voltada para a pesquisa em cenários de *Internet of Things* para aplicações de localização 3D em ambientes internos. A base de dados utilizada neste trabalho pode ser acessada no seguinte *link*: <https://github.com/alefelipeoliveira/3D-indoor-localization-BLE-Database>;
- A proposição de uma estratégia para uso apenas das âncoras com sinais de maior potência nesse cenário;
- A proposição de uma estratégia para lidar com situações de âncora não detectada e/ou para evitar a inserção de dados arbitrários ou empíricos para condições onde uma das âncoras não é detectada;
- A identificação de que um nível satisfatório de qualidade na localização pode ser obtido por esses sistemas utilizando 3 (três) ou mais âncoras.

11.2 Trabalhos futuros

Dentre os trabalhos futuros, destacamos:

- A investigação da abordagem de **filtro de âncoras** e dos algoritmos utilizados neste trabalho em outros cenários e ambientes internos – com dimensões e configurações diferentes – permitiria analisar a capacidade de generalização e eficiência da abordagem e dos algoritmos;
- Um aumento da quantidade de âncoras instaladas no ambiente também incrementaria a quantidade de atributos de entrada dos algoritmos de localização, isso no caso da abordagem considerada tradicional – em que se consideram como atributos de entrada todas as âncoras presentes na base de dados de treinamento. Dessa forma, considera-se como trabalho futuro a investigação dos algoritmos em ambientes com outras quantidades de âncoras detectáveis, para analisar o desempenho da abordagem proposta;
- O cenário implementado neste trabalho trata da localização de um dispositivo receptor estático, no entanto, a implementação com um dispositivo em movimento possibilitaria investigar a robustez da abordagem **filtro de âncoras** para localização num cenário dinâmico. Neste contexto, a inclusão de outros tipos de sensores como sonares, acelerômetros ou ultrassônicos, que fornecem diferentes atributos de entrada, poderia proporcionar maior acuracidade de localização;

- Uma característica de implementações de localização baseadas em análise de cena (*fingerprint*) é o esforço e tempo necessário na criação da base de dados de treinamento. A comparação de resultados utilizando diferentes quantidades de registros desta mesma base de dados permitiria avaliar o quanto a redução de instâncias para treinamento afeta o desempenho dos algoritmos de localização;
- A investigação da utilização da proposta de **filtro de âncoras** em conjunto com outros algoritmos de Aprendizado de Máquina na tarefa da localização.

REFERÊNCIAS

- ABINADER, F. M.; SOUSA, V. A. de; CHOUDHURY, S.; CHAVES, F. S.; CAVALCANTE, A. M.; ALMEIDA, E. P.; VIEIRA, R. D.; TUOMAALA, E.; DOPPLER, K. Lte/wi-fi coexistence in 5 ghz ism spectrum: Issues, solutions and perspectives. *Wireless Personal Communications*, Springer, v. 99, n. 1, p. 403–430, 2018.
- AHMADI, H.; BOUALLEGUE, R. Rssi-based localization in wireless sensor networks using regression tree. In: IEEE. *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*. [S.l.], 2015. p. 1548–1553.
- ALLURWAR, N.; NAWALE, B.; PATEL, S. Beacon for proximity target marketing. *International Journal of Engineering and Computer Science*, v. 5, n. 5, 2016.
- ARDUINO. *Arduino*. 2021. <<https://www.arduino.cc/>>. Acesso em: abril de 2021.
- AYKAÇ, M.; ERÇELEBI, E.; ALDIN, N. B. Zigbee-based indoor localization system with the personal dynamic positioning method and modified particle filter estimation. *Analog Integrated Circuits and Signal Processing*, Springer, v. 92, n. 2, p. 263–279, 2017.
- BABA, A. I. Calibrating time of flight in two way ranging. In: IEEE. *2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks*. [S.l.], 2011. p. 393–397.
- BELMONTE-HERNÁNDEZ, A.; HERNÁNDEZ-PEÑALOZA, G.; GUTIÉRREZ, D. M.; ÁLVAREZ, F. Swiblux: Multi-sensor deep learning fingerprint for precise real-time indoor tracking. *IEEE Sensors Journal*, IEEE, v. 19, n. 9, p. 3473–3486, 2019.
- BENSKY, A. *Short-range wireless communication*. [S.l.]: Newnes, 2019.
- BIANCHI, V.; CIAMPOLINI, P.; MUNARI, I. D. Rssi-based indoor localization and identification for zigbee wireless sensor networks in smart homes. *IEEE Transactions on Instrumentation and Measurement*, IEEE, v. 68, n. 2, p. 566–575, 2018.
- BLAS, A. D.; IPIÑA, D. López-de. Improving trilateration for indoors localization using ble beacons. In: IEEE. *2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*. [S.l.], 2017. p. 1–6.
- BREGAR, K.; HROVAT, A.; MOHORCIC, M. Nlos channel detection with multilayer perceptron in low-rate personal area networks for indoor localization accuracy improvement. In: *Proceedings of the 8th Jožef Stefan International Postgraduate School Students' Conference, Ljubljana, Slovenia*. [S.l.: s.n.], 2016. v. 31.
- CALDERONI, L.; FERRARA, M.; FRANCO, A.; MAIO, D. Indoor localization in a hospital environment using random forest classifiers. *Expert Systems with Applications*, Elsevier, v. 42, n. 1, p. 125–134, 2015.
- CHOOSAKSAKUNWIBOON, S.; TERAUWONG, C.; SUTTISIRIKUL, S.; ANANTAVRASILP, I.; THIEMJARUS, S.; WISADSUD, S.; KAEMARUNGSI, K. A pre-processing technique for ble-based indoor localization. In: SINGAPORE THERAPEUTIC, ASSISTIVE & REHABILITATIVE TECHNOLOGIES (START) CENTRE. *Proceedings of the 12th International Convention on Rehabilitation Engineering and Assistive Technology*. [S.l.], 2018. p. 241–244.
- CHUENURAJIT, T.; PHIMMASEAN, S.; CHERNTANOMWONG, P. Robustness of 3d indoor localization based on fingerprint technique in wireless sensor networks. In: IEEE. *2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*. [S.l.], 2013. p. 1–6.
- DAI, P.; YANG, Y.; WANG, M.; YAN, R. Combination of dnn and improved knn for indoor location fingerprinting. *Wireless Communications and Mobile Computing*, Hindawi, v. 2019, 2019.

- DJOSIC, S.; STOJANOVIC, I.; JOVANOVIC, M.; NIKOLIC, T.; DJORDJEVIC, G. L. Fingerprinting-assisted uwb-based localization technique for complex indoor environments. *Expert Systems with Applications*, Elsevier, v. 167, p. 114188, 2021.
- DJUKNIC, G. M.; RICHTON, R. E. Geolocation and assisted gps. *Computer*, IEEE, v. 34, n. 2, p. 123–125, 2001.
- DODDIPALLI, S.; KOTHARI, A. Compact uwb antenna with integrated triple notch bands for wban applications. *IEEE Access*, IEEE, v. 7, p. 183–190, 2018.
- ESPRESSIF. *ESP-WROOM-DATASHEET*. 2021. <https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf>. Acesso em: março de 2021.
- FATH, A. H.; POURANFARD, A.; FOROUGHIZADEH, P. Development of an artificial neural network model for prediction of bubble point pressure of crude oils. *Petroleum*, Elsevier, v. 4, n. 3, p. 281–291, 2018.
- GUPTA, N. K. *Inside Bluetooth low energy*. [S.l.]: Artech House, 2016.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The elements of statistical learning: data mining, inference, and prediction*. [S.l.]: Springer Science & Business Media, 2009.
- HE, S.; CHAN, S.-H. G. Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys & Tutorials*, IEEE, v. 18, n. 1, p. 466–490, 2015.
- HIERTZ, G. R.; DENTENEER, D.; STIBOR, L.; ZANG, Y.; COSTA, X. P.; WALKER, B. The ieee 802.11 universe. *IEEE Communications Magazine*, IEEE, v. 48, n. 1, p. 62–70, 2010.
- JAIN, A. K. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, Elsevier, v. 31, n. 8, p. 651–666, 2010.
- JIANG, Q.; MA, Y.; LIU, K.; DOU, Z. A probabilistic radio map construction scheme for crowdsourcing-based fingerprinting localization. *IEEE Sensors Journal*, IEEE, v. 16, n. 10, p. 3764–3774, 2016.
- JONDALE, S. R.; SHUBAIR, R.; LABADE, R. P.; LLORET, J.; GUNJAL, P. R. Application of supervised learning approach for target localization in wireless sensor network. In: *Handbook of Wireless Sensor Networks: Issues and Challenges in Current Scenario's*. [S.l.]: Springer, 2020. p. 493–519.
- KAGGLE. *Kaggle: Your Machine Learning and Data Science Community*. 2021. <<https://www.kaggle.com/>>. Acesso em: março de 2021.
- KAZEEM, O. O.; AKINTADE, O. O.; KEHINDE, L. O. Comparative study of communication interfaces for sensors and actuators in the cloud of internet of things. *Int. J. Internet Things*, v. 6, n. 1, p. 9–13, 2017.
- KHALEL, A. M. *Position location techniques in wireless communication systems*. 2010.
- KHALID, N.; AKAN, O. B. Wideband thz communication channel measurements for 5g indoor wireless networks. In: IEEE. *2016 IEEE International Conference on Communications (ICC)*. [S.l.], 2016. p. 1–6.
- KLEPAL, M.; PESCH, D. et al. A bayesian approach for rf-based indoor localisation. In: IEEE. *2007 4th International Symposium on Wireless Communication Systems*. [S.l.], 2007. p. 133–137.
- LASHKARI, B.; REZAZADEH, J.; FARAHBAKHS, R.; SANDRASEGARAN, K. Crowdsourcing and sensing for indoor localization in iot: A review. *IEEE Sensors Journal*, IEEE, v. 19, n. 7, p. 2408–2434, 2018.
- LAZIK, P.; RAJAGOPAL, N.; SHIH, O.; SINOPOLI, B.; ROWE, A. Alps: A bluetooth and ultrasound platform for mapping and localization. In: *Proceedings of the 13th ACM conference on embedded networked sensor systems*. [S.l.: s.n.], 2015. p. 73–84.
- LEE, B.; WOO, D.-M.; PARK, M.-K.; KIM, S. Development of self-localizer using collaboration of trilateration and triangulation. In: IEEE. *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. [S.l.], 2014. p. 729–733.

- LEE, C.; IP, C.; PARK, T.; CHUNG, S. A bluetooth location-based indoor positioning system for asset tracking in warehouse. In: IEEE. *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. [S.l.], 2019. p. 1408–1412.
- LIU, H.; DARABI, H.; BANERJEE, P.; LIU, J. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, IEEE, v. 37, n. 6, p. 1067–1080, 2007.
- LIU, J. Survey of wireless based indoor localization technologies. *Department of Science & Engineering, Washington University*, 2014.
- LU, X.; KIM, I. H.; XHAFI, A.; ZHOU, J.; TSAI, K. Reaching 10-years of battery life for industrial iot wireless sensor networks. In: IEEE. *2017 Symposium on VLSI Circuits*. [S.l.], 2017. p. C66–C67.
- LU, Y.-S.; LAI, C.-F.; HU, C.-C.; HUANG, Y.-M.; GE, X.-H. Path loss exponent estimation for indoor wireless sensor positioning. *KSII Transactions on Internet & Information Systems*, v. 4, n. 3, 2010.
- MALETIC, N.; SARK, V.; EHRIG, M.; GUTIÉRREZ, J.; GRASS, E. Experimental evaluation of round-trip tof-based localization in the 60 ghz band. In: IEEE. *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. [S.l.], 2019. p. 1–6.
- MARKIT, I. The internet of things: a movement, not a market. *IHS Market*, 2017.
- MILLER, B. A.; BISDIKIAN, C. *Bluetooth revealed*. [S.l.]: Prentice Hall PTR, 2001.
- MILLER, M. *Discovering Bluetooth*. [S.l.]: Sybex, 2001.
- MIZMIZI, M.; REGGIANI, L. Design of rssi based fingerprinting with reduced quantization measures. In: IEEE. *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. [S.l.], 2016. p. 1–6.
- MOHAMMADI, M.; AL-FUQAHA, A.; GUIZANI, M.; OH, J.-S. Semisupervised deep reinforcement learning in support of iot and smart city services. *IEEE Internet of Things Journal*, IEEE, v. 5, n. 2, p. 624–635, 2017.
- MOTRONI, A.; NEPA, P.; MAGNAGO, V.; BUFFI, A.; TELLINI, B.; FONTANELLI, D.; MACII, D. Sar-based indoor localization of uhf-rfid tags via mobile robot. In: IEEE. *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. [S.l.], 2018. p. 1–8.
- NODE-RED. *Node-RED*. 2021. <<https://nodered.org/>>. Acesso em: janeiro de 2021.
- NUMPY. *NumPy*. 2021. <<https://numpy.org/>>. Acesso em: março de 2021.
- OLIVEIRA, K. V. de; CASTELLI, H. M. E.; MONTEBELLER, S. J.; AVANCINI, T. G. P. Wireless sensor network for smart agriculture using zigbee protocol. In: IEEE. *2017 IEEE First Summer School on Smart Cities (S3C)*. [S.l.], 2017. p. 61–66.
- PANDAS. *pandas: Python Data Analysis Library*. 2021. <<https://pandas.pydata.org/>>. Acesso em: abril de 2021.
- PECORARO, G.; DOMENICO, S. D.; CIANCA, E.; SANCTIS, M. D. Csi-based fingerprinting for indoor localization using lte signals. *EURASIP Journal on Advances in Signal Processing*, Springer, v. 2018, n. 1, p. 49, 2018.
- PENG, L.; LIU, J.; SHENG, M.; ZHANG, Y.; HOU, D.; ZHENG, Y.; LI, J. 3d indoor localization based on spectral clustering and weighted backpropagation neural networks. In: IEEE. *2017 IEEE/CIC International Conference on Communications in China (ICCC)*. [S.l.], 2017. p. 1–6.
- RAMYA, C. M.; SHANMUGARAJ, M.; PRABAKARAN, R. Study on zigbee technology. In: IEEE. *2011 3rd International Conference on Electronics Computer Technology*. [S.l.], 2011. v. 6, p. 297–301.
- RIDOLFI, M.; VELDE, S. Van de; STEENDAM, H.; POORTER, E. D. Wifi ad-hoc mesh network and mac protocol solution for uwb indoor localization systems. In: IEEE. *2016 Symposium on Communications and Vehicular Technologies (SCVT)*. [S.l.], 2016. p. 1–6.

- ROSE, K.; ELDRIDGE, S.; CHAPIN, L. The internet of things: An overview. *The internet society (ISOC)*, v. 80, p. 1–50, 2015.
- ROY, P.; CHOWDHURY, C. A survey of machine learning techniques for indoor localization and navigation systems. *Journal of Intelligent & Robotic Systems*, Springer, v. 101, n. 3, p. 1–34, 2021.
- RUSSELL, S.; NORVIG, P. Artificial intelligence: a modern approach. 2010.
- SADOWSKI, S.; SPACHOS, P. Rssi-based indoor localization with the internet of things. *IEEE Access*, IEEE, v. 6, p. 30149–30161, 2018.
- SADOWSKI, S.; SPACHOS, P.; PLATANIOTIS, K. N. Memoryless techniques and wireless technologies for indoor localization with the internet of things. *IEEE Internet of Things Journal*, IEEE, 2020.
- SALMAN, A.; EL-TAWAB, S.; YORIO, Z.; HILAL, A. Indoor localization using 802.11 wifi and iot edge nodes. In: IEEE. *2018 IEEE Global Conference on Internet of Things (GCIoT)*. [S.l.], 2018. p. 1–5.
- SCHNEEBELI, H. *C para microcontroladores*. [S.l.: s.n.], 2007.
- SCHÜSSEL, M. Angle of arrival estimation using wifi and smartphones. In: *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. [S.l.: s.n.], 2016. v. 4, p. 7.
- SCIKIT-LEARN. *scikit-learn: machine learning in Python*. 2021. <<https://scikit-learn.org/stable/>>. Acesso em: abril de 2021.
- SHAIKH, F. K.; ZEADALLY, S. Energy harvesting in wireless sensor networks: A comprehensive review. *Renewable and Sustainable Energy Reviews*, Elsevier, v. 55, p. 1041–1054, 2016.
- SHAO, C.; NIRJON, S.; FRAHM, J.-M. Years-long binary image broadcast using bluetooth low energy beacons. In: IEEE. *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*. [S.l.], 2016. p. 225–232.
- SHIH, C.-Y.; MARRÓN, P. J. Cola: Complexity-reduced trilateration approach for 3d localization in wireless sensor networks. In: IEEE. *2010 Fourth International Conference on Sensor Technologies and Applications*. [S.l.], 2010. p. 24–32.
- SIG, B. *Bluetooth SIG*. 2021. <<https://www.bluetooth.com/>>. Acesso em: fevereiro de 2021.
- TAUD, H.; MAS, J. Multilayer perceptron (mlp). In: *Geomatic Approaches for Modeling Land Change Scenarios*. [S.l.]: Springer, 2018. p. 451–455.
- THOMAS, F.; ROS, L. Revisiting trilateration for robot localization. *IEEE Transactions on robotics*, IEEE, v. 21, n. 1, p. 93–101, 2005.
- TI. *2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*. 2021. <<https://www.ti.com/lit/ds/symlink/cc2420.pdf>>. Acesso em: março de 2021.
- TJENSVOLD, J. M. Comparison of the ieee 802.11, 802.15. 1, 802.15. 4 and 802.15. 6 wireless standards. In: IEEE: *September*. [S.l.: s.n.], 2007. v. 18.
- UCI. *UCI Machine Learning Repository*. 2021. <<https://archive.ics.uci.edu/ml/index.php>>. Acesso em: abril de 2021.
- VÁŠČÁK, J.; SAVKO, I. Radio beacons in indoor navigation. In: IEEE. *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*. [S.l.], 2018. p. 283–288.
- WOOLLEY, M. Bluetooth direction finding: A technical overview. *Bluetooth. Accessed Sept*, v. 15, 2020.
- XIAO, C.; YANG, D.; CHEN, Z.; TAN, G. 3-d ble indoor localization based on denoising autoencoder. *IEEE Access*, IEEE, v. 5, p. 12751–12760, 2017.
- XUE, W.; HUA, X.; LI, Q.; YU, K.; QIU, W.; ZHOU, B.; CHENG, K. A new weighted algorithm based on the uneven spatial resolution of rssi for indoor localization. *IEEE Access*, IEEE, v. 6, p. 26588–26595, 2018.

YASSIN, A.; NASSER, Y.; AWAD, M.; AL-DUBAI, A.; LIU, R.; YUEN, C.; RAULEFS, R.; ABOUTANIOS, E. Recent advances in indoor localization: A survey on theoretical approaches and applications. *IEEE Communications Surveys & Tutorials*, IEEE, v. 19, n. 2, p. 1327–1346, 2016.

ZAFARI, F.; GKELIAS, A.; LEUNG, K. K. A survey of indoor localization systems and technologies. *IEEE Communications Surveys & Tutorials*, IEEE, v. 21, n. 3, p. 2568–2599, 2019.

ZEBRA. Zebra. 2021. <<https://www.zebra.com/br/pt/products/location-technologies/ultra-wideband.html>>. Acesso em: março de 2021.

ZERYNTH. *DOIT Esp32 DevKit v1 - Zerynth Documentation*. 2021. <https://docs.zerynth.com/latest/reference/boards/doit_esp32/docs/>. Acesso em: março de 2021.

ZHOU, X.; CHEN, T.; GUO, D.; TENG, X.; YUAN, B. From one to crowd: A survey on crowdsourcing-based wireless indoor localization. *Frontiers of Computer Science*, Springer, v. 12, n. 3, p. 423–450, 2018.