UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MARCOS FELIPE BARBOZA DE ABREU

Detecção online de dispositivos sem fio intrusos usando o sinal eletromagnético de transmissão

Goiânia
2022

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES

E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico

[X] Dissertação [ ] Tese [ ] Outro*:_____

*No caso de mestrado/doutorado profissional, indique o formato do Trabalho de Conclusão de Curso, permitido no documento de área, correspondente ao programa de pós-graduação, orientado pela legislação vigente da CAPES.

Exemplos: Estudo de caso ou Revisão sistemática ou outros formatos.

2. Nome completo do autor

Marcos Felipe Barboza de Abreu

3. Título do trabalho

Detecção online de dispositivos sem fio intrusos usando o sinal eletromagnético de transmissão

4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:
a) consulta ao(à) autor(a) e ao(à) orientador(a);
b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.
O documento não será disponibilizado durante o período de embargo.
Casos de embargo:
- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Kleber Vieira Cardoso**, **Professor do Magistério Superior**, em 20/09/2022, às 13:34, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **MARCOS FELIPE BARBOZA DE ABREU**, **Discente**, em 20/09/2022, às 14:04, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3203566** e o código CRC **E8E78F29**.

Referência: Processo nº 23070.038600/2022-36 SEI nº 3203566

MARCOS FELIPE BARBOZA DE ABREU

Detecção online de dispositivos sem fio intrusos usando o sinal eletromagnético de transmissão

Dissertação apresentada ao Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação.

Orientador: Prof. Dr. Kleber Vieira Cardoso

Co-Orientador: Prof. Dr. Flávio Henrique Teles Vieira

Goiânia
2022

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Abreu, Marcos Felipe Barboza de
Detecção online de dispositivos sem fio intrusos usando o sinal eletromagnético de transmissão [manuscrito] / Marcos Felipe Barboza de Abreu. - 2022.
LV, 55 f.

Orientador: Prof. Dr. Kleber Vieira Cardoso; co-orientador Dr. Flávio Henrique Teles Vieira.
Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2022.
Bibliografia.
Inclui lista de figuras, lista de tabelas.

1. Identificação radiométrica. 2. Diferenciação de dispositivos. 3. Internet das Coisas. I. Vieira Cardoso, Kleber, orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE DISSERTAÇÃO

Ata nº **18/2022** da sessão de Defesa de Dissertação de **Marcos Felipe Barboza de Abreu**, que confere o título de Mestre em Ciência da Computação, na área de concentração em Ciência da Computação.

Aos vinte e três dias do mês de agosto de dois mil e vinte e dois, a partir das catorze horas, na sala 151 do Instituto de Informática, realizou-se a sessão pública de Defesa de Dissertação intitulada **"Detecção online de dispositivos sem fio intrusos usando o sinal eletromagnético de transmissão"**. Os trabalhos foram instalados pelo Orientador, Professor Doutor Kleber Vieira Cardoso (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professor Doutor Flávio Henrique Teles Vieira (EMC/UFG - coorientador), membro titular interno; Professor Doutor Aldebaro Barreto da Rocha Klautau Júnior (ITEC/UFPA), membro titular externo, cuja participação ocorreu por videoconferência; e Professora Doutora Sand Luz Corrêa (INF/UFG), membra titular interna. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor Kleber Vieira Cardoso, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos vinte e três dias do mês de agosto de dois mil e vinte e dois.

TÍTULO SUGERIDO PELA BANCA

Documento assinado eletronicamente por **Aldebaro Barreto da Rocha Klautau Junior**, **Usuário Externo**, em 23/08/2022, às 16:46, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Sand Luz Corrêa**, **Professor do Magistério Superior**, em 23/08/2022, às 16:46, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **MARCOS FELIPE BARBOZA DE ABREU**, **Discente**, em 23/08/2022, às 17:01, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Kleber Vieira Cardoso**, **Professor do Magistério Superior**, em 23/08/2022, às 18:38, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Flavio Henrique Teles Vieira**, **Professor do Magistério Superior**, em 23/08/2022, às 20:04, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3065514** e o código CRC **87F83137**.

Referência: Processo nº 23070.038600/2022-36 SEI nº 3065514

Marcos Felipe Barboza de Abreu

Graduado em Ciência da Computação pela Universidade Federal de Goiás. Durante o mestrado participou de projetos de pesquisa e desenvolvimento junto à Rede Nacional de Ensino e Pesquisa (RNP) e National Institute for Research in Digital Science and Technology (Inria). Faz pesquisa na área de Redes de Computadores, especificamente Redes Definidas por Software (SDN) e Segurança da Informação.

Dedico este trabalho primeiramente a Deus, por ser essencial em minha vida, autor de meu destino, meu guia. Ao meu pai Antonio, minha mãe Maria Marta, minha irmã Alline e minha esposa Larissa.

Agradecimentos

Agradeço a Deus, por me dar saúde, inteligência e recursos desde o ventre da minha mãe.

Aos meus pais Antonio e Maria Marta, pelo incentivo aos estudos, pelo apoio financeiro durante toda minha vida escolar e pelo amor durante toda minha vida. Também a toda minha família, em especial as minhas tias Geralda e Maria Garcia, que estiveram presentes desde a infância dando suporte aos meus estudos.

A minha esposa, Larissa, que me apoiou nos momentos difíceis, que ouviu minhas reclamações, me deu bons conselhos e compreendeu todos momentos que precisei me ausentar para realizar este trabalho.

Ao Professor Kleber Vieira Cardoso, por sua orientação, seus conselhos e por acompanhar uma grande parte da minha jornada na ciência.

Ao Professor Flávio Henrique Teles Vieira, por sua orientação, contribuição e confiança.

Aos professores que contribuíram de uma maneira especial na minha formação: Antonio Carlos de Oliveira Júnior, Fábio Moreira Costa, Sand Luz Corrêa, Vinicius da Cunha Martins Borges, e aos demais professores do Instituto de Informática.

Ao Professor Abdallah S. Abdallah, por sua orientação e contribuição.

Aos Professores Nadjib e Aline, que me acompanharam durante o estágio no Inria e com os quais aprendi bastante. Também ao meu companheiro Abhishek Mishra, pela troca de conhecimentos durante o estágio de pesquisa no Inria.

Aos meus amigos: Leonardo, Saymon, Phelipe e os demais, por todos esses anos de amizade.

Aos meus companheiros do grupo de pesquisa Labora: Pabllo, João Esper, Ciro, Henrique, Elton, Divino, e os demais, pela amizade, apoio e incentivo.

A toda equipe da secretaria acadêmica, em especial à Mariana, pela atenção, paciência e suporte operacional.

Ao Professor Aldebaro Klautau por aceitar ser membro da banca de mestrado.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), ao National Institute for Research in Digital Science and Technology (INRIA) e a Rede Nacional de Ensino e Pesquisa (RNP) pelo suporte financeiro.

A company can spend hundreds of thousands of dollars on firewalls, intrusion detection systems and encryption and other security technologies, but if an attacker can call one trusted person within the company, and that person complies, and if the attacker gets in, then all that money spent on technology is essentially wasted.

Kevin Mitnick,
A convicted hacker debunks some myths - CNN.com.

Resumo

Abreu, Marcos Felipe Barboza. **Detecção online de dispositivos sem fio intrusos usando o sinal eletromagnético de transmissão**. Goiânia, 2022. 57p. Dissertação de Mestrado. Programa de Pós graduação em ciência da computação, Instituto de Informática, Universidade Federal de Goiás.

A identificação de dispositivos de Internet das Coisas (IoT) através do sinal eletromagnético é um tema bastante investigado na literatura, sendo essa técnica considerada altamente acurada por diversos trabalhos. O uso de técnicas *offline*, ou seja, quando não há presença de dispositivos novos, é amplamente explorado, mas até o momento, não são encontrados sistemas utilizando efetivamente a detecção de dispositivos não conhecidos forma *online*, i.e., um dos maiores potenciais desse tipo de técnica não vem sendo investigado. Este trabalho apresenta um sistema *online* que diferencia dispositivos autênticos de dispositivos intrusos. Para isso, é explorado o uso da matriz de probabilidade de classificadores, visando identificar dispositivos desconhecidos pelos mesmos. Além da técnica, também é apresentado um sistema que tem como características um arquitetura modular, extensível e genérica, que visa interferir minimamente no fluxo normal de uma aplicação de Internet das Coisas. O sistema é implementado utilizando a ferramenta GNU Radio sendo apresentados experimentos, esses que visam mostrar a viabilidade da técnica. Toda a discussão é baseada em dados coletados de ambientes reais, utilizando dispositivos das tecnologias de comunicação sem fio LoRa e ZigBee. Além disso, no trabalho foram analisados dados da tecnologia WiFi, provenientes de coleções encontradas na literatura. Os testes mostram ser possível identificar dispositivos desconhecidos na ordem de milissegundo, com uma baixa taxa de erro.

Palavras–chave

Identificação radiométrica, diferenciação de dispositivos, autenticação de dispositivos, Internet das Coisas.

Abstract

Abreu, Marcos Felipe Barboza. **Classification and identification of wireless devices using the transmitting electromagnetic signal**. Goiânia, 2022. 57p. MSc. Dissertation. Programa de Pós graduação em ciência da computação, Instituto de Informática, Universidade Federal de Goiás.

The identification of Internet of Things (IoT) devices through the electromagnetic signal is a topic widely investigated in the literature, and this technique is considered highly accurate by several works. The use of offline techniques, that is, when there is no presence of new devices, is widely explored, but so far, systems are not found effectively using the detection of unknown devices in the online way, i.e. , one of the greatest potentials of this type of technique has not been investigated. This work presents an online system that differentiates authentic devices from intrusive devices. For this, the use of the probability matrix of classifiers is explored, aiming to identify unknown devices by them. In addition to the technique, it is also presented a system that features a modular, extensible and generic architecture, which aims to minimally interfere with the normal flow of an Internet of Things application. The system is implemented using the GNU Radio tool and experiments are presented, which aim to show the feasibility of the technique. The entire discussion is based on data collected from real environments, using devices from wireless communication technologies LoRa and ZigBee. In addition, the work analyzed data from WiFi technology, from collections found in the literature. Tests show that it is possible to identify unknown devices in the order of milliseconds, with a low error rate.

Keywords

Radiometric identification, privacy analysis, device differentiation, device authentication, Internet of Things.

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

Internet das Coisas, do inglês *Internet of Things* (IoT), é um paradigma que está ganhando espaço na área de comunicações sem fio modernas devido ao seu grande impacto na vida da população. E-saúde, assistentes pessoais, vida assistida são exemplos de aplicações presentes em um ambiente doméstico, enquanto no contexto corporativo temos, como exemplos, automação, logística, gerenciamento de negócios/pessoas e transporte inteligente [Gomez e Paradells 2010].

A segurança de um dispositivo de Internet das Coisas passa por suas várias camadas e protocolos de rede. Um ataque bem conhecido em redes de computadores é o roubo de identidade. Em dispositivos sem fio, sua mitigação pode ser baseada em identificadores únicos, como o endereço MAC (*Media Access Control*) na tecnologia WiFi. No entanto, em algumas tecnologias, esses identificadores podem ser facilmente forjados, permitindo que o roubo de identidade seja bem-sucedido. Outras tecnologias, como no caso da LoRa, não fornecem mecanismos de segurança por padrão. Todos os dispositivos que usam o mesmo fator de propagação (SF) e largura de banda (Bw) podem se comunicar. Neste caso, cabe ao desenvolvedor da aplicação usar um protocolo que implementa uma camada de segurança, como o LoRaWAN, ou definir um protocolo de autenticação e implementá-lo. Caso o desenvolvedor não tenha esse cuidado, um invasor pode transmitir dados falsos, se passando pelo dispositivo, ou até mesmo replicar dados anteriormente coletados.

Estudos propõem aumentar o nível de segurança dos dispositivos adicionando informações coletadas a partir do sinal eletromagnético dos dispositivos ao processo de autenticação [Barbeau, Hall e Kranakis 2006, Brik et al. 2008]. A diferenciação de dispositivos é possível, pois o processo de fabricação do *hardware* permite algumas imperfeições, não intencionais, no circuito gerador do sinal. Estas não interferem na comunicação do dispositivo, mas são suficientes para que um algoritmo de aprendizado de máquina consiga diferenciar sinais originados de diferentes dispositivos.

Aplicar as técnicas na segurança de dispositivos, principalmente em mecanismos de autenticação na Internet das Coisas, também é proposto em alguns trabalhos [Tian et al. 2019]. Entretanto, esses se concentram em relacionar unicamente a viabilidade

da solução com o desempenho no processo de classificação de dispositivos previamente conhecidos. O fato das técnicas *offline* não considerarem cenários em que dispositivos desconhecidos estão presentes e podendo haver variações nestes cenários, implica em poucas aplicações práticas no mundo real.

O trabalho de [Reus-Muns et al. 2020] é o que mais se aproxima do que seria um sistema *online*. Ele apresenta um algoritmo que classifica estações base 5G como intrusas ou não baseado no *score* de classificação de uma rede neural. Embora o trabalho apresente resultados satisfatórios na classificação de Estações Base previdamente conhecidas, ele não apresenta testes envolvendo Estações Base desconhecidas pelo classificador. Um passo importante é ter algoritmos que diferenciem bem os dispositivos conhecidos dos não conhecidos, com uma baixa taxa de erros. Entretanto, existem outras variáveis que podem definir a viabilidade da aplicação dessas técnicas na detecção de intrusos.

Neste trabalho, propomos um sistema *online* de detecção de dispositivos *IoT* intrusos, baseado nas características extraídas dos sinais transmitidos pelos mesmos. Abaixo são listados alguns requisitos essenciais em um sistema de detecção de intruso, considerados no desenvolvimento deste trabalho:

- Não intrusivo: os dados devem ser coletados de forma passiva e as decisões precisam ser tomadas com interferência miníma do ser humano. Assume-se que a assinatura do dispositivo foi previamente coletada e computada.
- Independente de tecnologia: o sistema deve autenticar diferentes dispositivos, independente de sua tecnologia de camada física. A avaliação desta característica consiste em testar a técnica com dispositivos de diferentes tecnologias.
- Robusto às variações do ambiente: a técnica deve responder bem às variações nas condições do ambiente, como o nível de interferência.
- Resposta rápida: o mecanismo de identificação não deve adicionar um atraso de tempo significativo, que possa interferir na comunicação do dispositivo. A verificação rápida da identidade evita que intrusos se passem pelo dispositivo por um período mais longo.

1.1 Objetivos

Tendo em vista as potencialidades do uso da identificação das imperfeições, encontradas no *hardware*, como mecanismo de diferenciação dos dispositivos de comunicação sem fio, este trabalho tem o objetivo de apresentar uma ferramenta de detecção *online* de dispositivos de Internet das Coisas intrusos.

Para atingir o objetivo, a solução proposta deve demonstrar viabilidade na aplicação em sistemas de Internet das Coisas, considerando suas características e limitações.

Para isso, será apresentada uma arquitetura que possuí um arcabouço modular e flexível, podendo ser facilmente extensível.

Também como objetivo específico, para demonstrar a eficácia da técnica, será realizada uma avaliação da solução em base de dados coletadas localmente e também bases encontradas na literatura. Nesta avaliação, a acurácia foi escolhida como parâmetro para comparar os classificadores SVM, LDA e KNN, considerados cenários de variação de temperatura e distância entre receptor e transmissor.

E por último, será apresentado um protótipo, implementado usando um Rádio Definido por *Software* e a ferramenta GNU Radio. O protótipo será utilizado para testar simulações de cenários em que dispositivos reais enviam mensagens para uma aplicação, também cenários em que um dispositivo intruso tenta se passar por outro.

1.2 Organização da dissertação

O restante do texto está organizado em mais 4 capítulos. No Capítulo 2, são apresentados os conceitos fundamentais de processamento de sinais, as ferramentas utilizadas e as técnicas aplicadas na análise dos dados e na classificação, além dos principais trabalhos relacionados a esta dissertação.

O Capítulo 3 descreve a metodologia utilizada nas fases de aquisição, processamento e extração das características a partir do sinal coletado. Também apresenta a proposta do método de classificação *online*.

No Capítulo 4, são apresentadas duas coleções de dados coletadas a partir de dispositivos em um ambiente de laboratório. Também são apresentados os experimentos aplicando algoritmos de classificação nas características extraídas destas coleções. Os mesmos algoritmos são aplicados em coleções encontradas na literatura. Também são apresentados os resultados de simulações com dispositivos reais cenários de ataque, utilizando um protótipo da ferramenta *online*, que também é descrita no Capítulo. Por último, o Capítulo 5 resume as contribuições e apresenta discussões sobre a extensibilidade da técnica e perspectivas de trabalhos futuros.

CAPÍTULO 2

Fundamentação teórica e revisão da literatura

Neste Capítulo, são apresentados os fundamentos necessários para a compreensão do trabalho. A Seção 2.1 descreve a teoria de transmissão e processamento de sinais digitais. A Seção 2.2 apresenta a figura de Traço de Constelação Diferencial, que é gerada a partir do sinal coletado dos dispositivos. Na Seção 2.4 são apresentados os principais características dos Rádio Definidos por Software, também a ferramenta GNU Radio, utilizada no protótipo da solução. Na Seção 2.5 são apresentados os algoritmos de aprendizado de máquina utilizados nos testes. Por último, a Seção 2.6 apresenta os trabalhos relacionados.

2.1 Fundamentos de processamento de sinais

O sinal analógico pode ser digitalizado por um conversor analógico-digital (ADC) de acordo com uma taxa de amostragem [Candès e Wakin 2008]. O teorema de amostragem de Nyquist-Shannon [Shannon 1949] afirma que um sinal pode ser reconstruído exatamente a partir de suas amostras se a frequência de amostragem for maior que o dobro do componente de frequência mais alto do sinal.

Em uma transmissão digital utilizando modulação, os dados (*bits*) são convertidos em um sinal analógico para serem transmitidos no meio. Um sinal analógico complexo x(t) é formado pelas duas componentes $\{xI(t), xQ(t)\}$, onde $xI(t)$ é a parte real e $xQ(t)$ é a parte imaginária, denominadas Em fase (I) e Quadratura (Q) respectivamente [Haykin 2008]. A relação entre essas componentes pode ser representada como:

$$s(t) = xI(t)\cos(2\pi fct) - xQ(t)\sin(2\pi fct) \tag{2-1}$$

Seja a Envoltória complexa dada por:

$$x(t) = xI(t) + jxQ(t) \tag{2-2}$$

onde j é a raiz quadrada de -1. Utilizando a fórmula de Euler:

$$e^{-j2\pi fct} = \cos(2\pi fct) + j\sin(2\pi fct) \tag{2-3}$$

Simplifica-se a equação que representa um sinal em banda base passante em:

$$s(t) = x(t)e^{-j2\pi fct} \tag{2-4}$$

conhecida como equação de transmissão de portadora única.

A Figura 2.1 ilustra a relação entre os valores de I e Q, representados em coordenadas polares. Essa representação é uma forma mais didática de relacionar esses dois valores e será útil para explicar a Figura do traço Diferencial na próxima seção. Entretanto, para desenvolver as equações fundamentais de processamento de sinais, será utilizado o traço analítico com a notação complexa, por ser mais simples matematicamente.

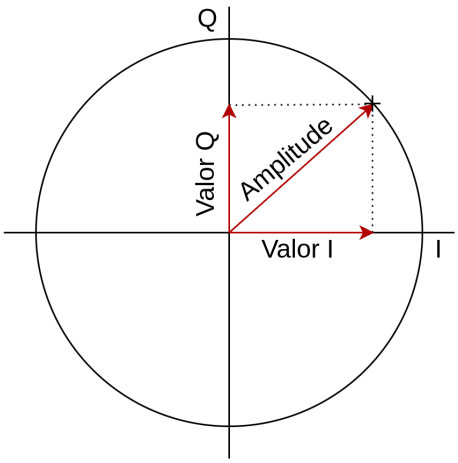


Figura 2.1: Componentes $xI(t)$ e $xQ(t)$ de uma amostra de um sinal.

O sinal s(t) é transmitido em um meio sem fio e o receptor recebe um sinal r(t) em uma frequência *fcr*, que é representado como:

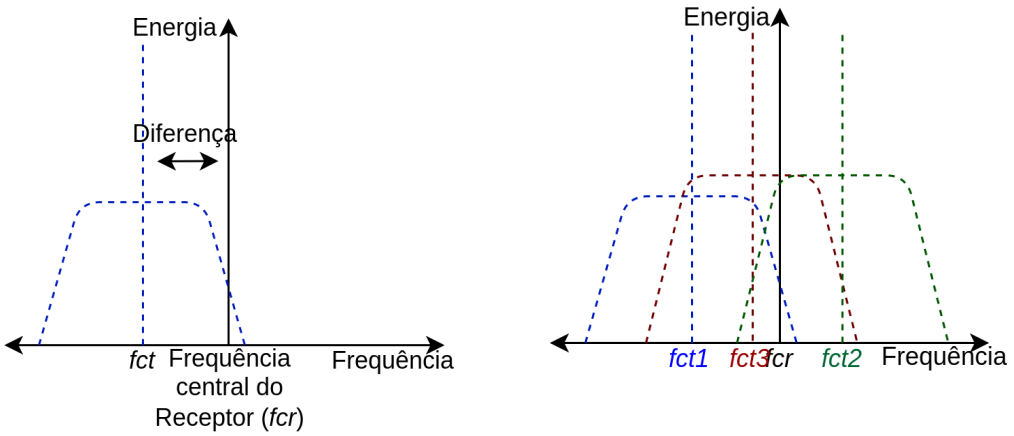
$$r(t) = x(t)e^{-j2\pi fcr} \tag{2-5}$$

Se assumimos um canal de transmissão ideal e um processo de demodulação ideal, a *fcr* é igual à *fct* e x(t) pode ser obtido convertendo o sinal recebido r(t) através da seguinte operação:

$$y(t) = r(t) \cdot e^{j2\pi fct} = x(t)e^{-j2\pi fcr} \cdot e^{j2\pi fct} = x(t) \tag{2-6}$$

Em condições reais, a frequência central obtida no receptor *fcr* pode diferir da frequência central do transmissor. Esse fenômeno é chamado de desvio de frequência da portadora (CFO - *Carrier Frequency Offset*) [Mohammadian e Tellambura 2021] e acontece devido à falta de sincronização entre o transmissor e receptor.

A diferença $\triangle f = fcr - fct$ é relativa aos dois dispositivos envolvidos na comunicação, isso significa que, se o dispositivo receptor for substituído por outro, o $\triangle f$ será diferente. A Figura 2.2 ilustra essa diferença entre receptores e transmissores. Na situação da Figura 2.2(b), em que temos um receptor recebendo sinal de vários dispositivos, podemos observar que cada dispositivo tem uma frequência central relativa ao receptor diferente dos demais.



(a) *Um transmissor* (b) *Múltiplos transmissores*

Figura 2.2: Frequência relativa entre um receptor e transmissores.

A Figura 2.3 ilustra o processo de conversão quando ocorre o efeito CFO. O sinal em banda base y(t) obtido no receptor vai ser diferente do sinal x(t) transmitido por um fator $e^{j2\pi\triangle ft}$ como:

$$y(t) = r(t) \cdot e^{j2\pi fct} = x(t)e^{-j2\pi fcr} \cdot e^{j2\pi fct} = x(t) \cdot e^{j2\pi\triangle ft} \tag{2-7}$$

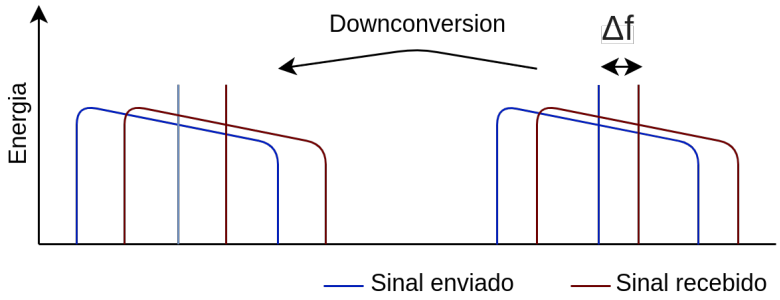


Figura 2.3: Processo de downconversion e o efeito CFO.

2.2 Figura de Traço de Constelação Diferencial

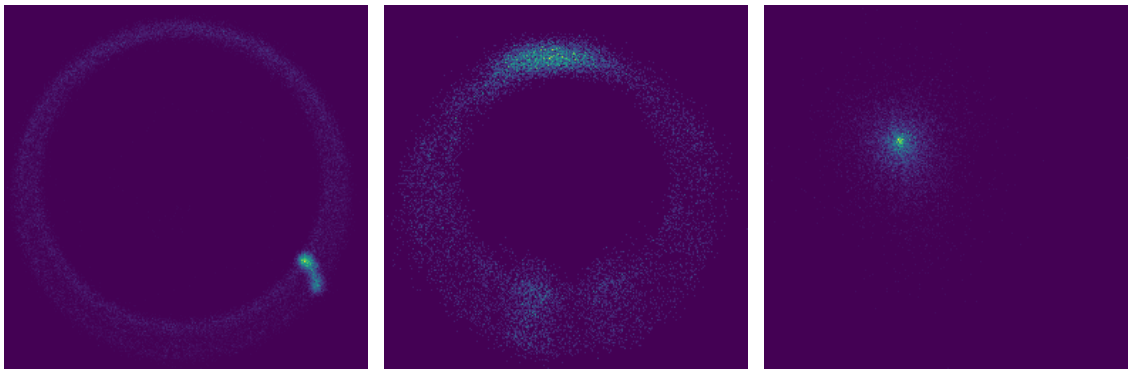
O sinal modulado I/Q contendo $xI(t)$ e $xQ(t)$ pode ser analisado a partir de uma representação em um plano, onde o sinal em fase é plotado no eixo X e o sinal em quadratura no eixo Y. A figura gerada a partir da combinação desses dois sinais em um período de tempo é chamada de figura de rastreamento de constelação (*CTF - Constellation Trace Figure*).

Os autores em [Peng et al. 2016], propõem a técnica denominada figura de Traço de Constelação Diferencial (*DCTF - Differential Constellation Trace Figure*), que é a aplicação de uma operação diferencial na *CTF*. Com isso, é possível observar o efeito *CFO* na *DCTF* gerada a partir dos sinais I/Q de um dispositivo.

Considerando o sinal $y(t)$ recebido, a operação diferencial é calculada multiplicando $y(t)$ pelo conjugado de $y(t+n)$:

$$\begin{aligned} D(t) &= y(t) \cdot y^*(t+n) \\ &= x(t) \cdot e^{j2\pi\triangle ft} \cdot x^*(t+n) \cdot e^{j2\pi\triangle f(t+n)} \\ &= x(t) \cdot x^*(t+n) \cdot e^{-j2\pi\triangle f(t+n)} \end{aligned} \tag{2-8}$$

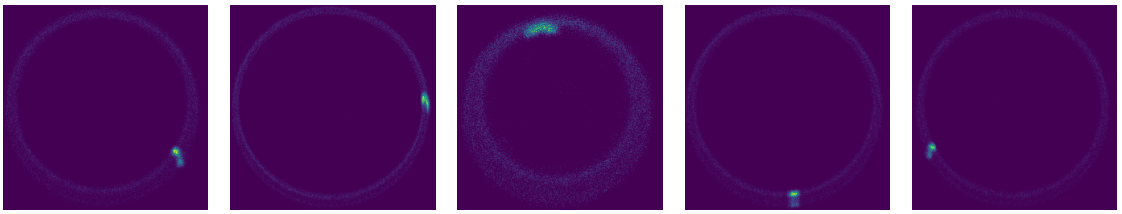
A fim de mostrar as características de diferentes *DCTFs* de forma mais intuitiva, são utilizadas cores para representar a densidade dos símbolos. Uma imagem é construída a partir de uma escala de cores, onde a cor azul representa a menor densidade e a cor amarela a maior. A Figura 2.4 apresenta três imagens, que são resultantes da aplicação da operação diferencial na *CTF* dos dispositivos LoRa 2.4(a), ZigBee 2.4(b) e WiFi 2.4(c). Cada tecnologia, visualmente, possui sua própria característica.



(a) *DCTF do LoRa* (b) *DCTF do ZigBee* (c) *DCTF do WiFi*

Figura 2.4: Exemplos de *DCTF* em diferentes tecnologias.

Também é possível visualizar diferença entre as *DCTFs* de alguns dispositivos da mesma tecnologia. Como visto na Figura 2.5, cada um dos 5 dispositivos possui uma região onde a densidade de pontos é maior.



(a) *Dispositivo 1* (b) *Dispositivo 2* (c) *Dispositivo 3* (d) *Dispositivo 4* (e) *Dispositivo 5*

Figura 2.5: Exemplos de *DCTFs* extraídas de diferentes dispositivos de uma mesma tecnologia (LoRa).

O artigo que propõe a técnica de *DCTF* [Peng et al. 2016] escolhe o valor de n ideal realizando testes de variação deste parâmetro. Nos testes realizados neste trabalho, vimos que um valor ótimo para n é o tamanho de um símbolo na modulação específica. Por exemplo, a duração de um símbolo no LoRa, de acordo com os parâmetros configurados no dispositivo e com a taxa de amostragem, é de 640 amostras.

2.3 Tecnologias de comunicação

Dispositivos LoRa (*Long Range*) [Bor, Vidler e Roedig 2016] têm sido amplamente adotados em aplicações de Internet das Coisas. A tecnologia é caracterizada por reduzir a complexidade do *hardware* e da rede, permitindo a comunicação através de uma infraestrutura mínima e com um baixo consumo de energia. Um dispositivo LoRa possui quatro parâmetros configuráveis: frequência central, fator de espalhamento (SF), largura de banda e taxa de codificação. A camada física da tecnologia LoRa modula sinais em sub-bandas de rádio nas faixas de frequências não licenciadas na ordem dos MHz.

A tecnologia LoRa não fornece mecanismos de segurança por padrão. Quaisquer dispositivos que estejam utilizando o mesmo fator de espalhamento (*Spreading Factor - SF*) e largura de banda poderão se comunicar. Com isso, um atacante pode coletar facilmente dados de dispositivos transmissores, verificar o padrão de transmissão e transmitir dados falsos. Portanto, é necessário utilizar um protocolo de camada superior, como o *LoRaWAN* [Haxhibeqiri et al. 2018], que implementa mecanismos de segurança.

A norma IEEE 802.15.4 é um padrão global de *hardware* e software e é utilizada nas especificações dos padrões ZigBee. Algumas características mandatórias dessa especificação são alta confiabilidade, escalabilidade, baixo custo, baixo consumo energético e baixa taxa de transmissão e recepção de dados [Ergen 2004]. As principais aplicações da tecnologia ZigBee são sensores e controladores, esses que não precisam de uma alta largura de banda mas precisam de baixa latência e baixo consumo energético.

Dispositivos ZigBee utilizam a segurança de camada MAC para transmissão de dados. Em aplicações que utilizam o padrão de múltiplos saltos, é necessário utilizar

protocolos de camadas superiores para adicionar segurança na comunicação. Os ataques direcionados à camada física dos dispositivos ZigBee utilizam *SDRs* para escutar ou adulterar os dados dos quadros.

2.4 Rádio definido por software

Rádio definido por *software* (*SDR - Software Defined Radio*) é um sistema de comunicação onde os componentes que, em geral, fazem parte do *hardware* e.g., amplificadores, filtros, moduladores e demoduladores, são implementados por *software* [Dillinger, Madani e Alonistioti 2005]. Desta forma, consegue-se programar e testar facilmente novas técnicas de processamento de sinais, sem grandes custos. Os *SDRs* são importantes na evolução das redes de comunicação sem fio pois é necessário se adaptar rapidamente às novas necessidades dos usuários e às especificações de *hardware*.

A arquitetura de um *SDR* está ilustrada na Figura 2.6. Em geral, os *SDRs* usam várias antenas para cobrir uma amplitude de frequências. O *RF Front End* é um circuíto implementado no hardware do *SDR* e tem a função de transmitir e receber o sinal em várias frequências. O sinal analógico recebido é convertido em sinal digital pelo ADC e o sinal digital recebido é convertido em analógico pelo DAC. O *Digital Front End* é responsável pelo processamento no sinal digital. Essa parte do processamento é feita em software.

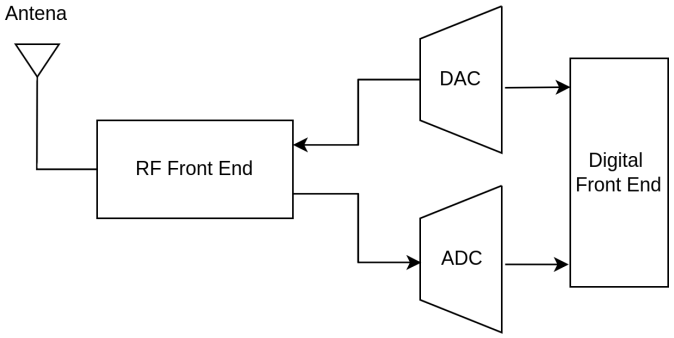


Figura 2.6: Arquitetura de um Rádio Definido por Software.

Existem diferentes modelos de *SDRs* no mercado. Os modelos mais usados nos trabalhos de identificação de dispositivos por assinatura radiométrica são da linha USRP, sendo que, o modelo mais simples custa cerca de US$ 900[1], mas podem chegar a custar US$ 18 000[2]. O modelo USRP mais popular opera nas faixas de 70 MHz a 6 GHz. Os modelos de *SDR* mais baratos apresentam algumas limitações, pois possuem componentes

[1] https://www.ettus.com/all-products/ub200-kit/ - Último acesso em 27/05/2022
[2] https://www.ettus.com/product-categories/usrp-networked-series/ - Último acesso em 27/05/2022

de *hardware* mais simples. Algumas dessas limitações como a baixa taxa de amostragem, faixa de espectro limitada e largura de banda pequena podem inviabilizar o uso do *SDR* na coleta de dados em determinadas tecnologias.

Exemplos de *SDR* baratos é a linha RTL-SDR, que custam cerca de US$ 30[3] e possuem somente a função de recepção de sinal. Este *SDR* opera nas faixas de frequência de 24 MHz a 1766 MHz. Com ele é possível coletar sinais da tecnologia LoRa, entretanto, ele não pode receber sinais de dispositivos ZigBee e WiFi, que operam nas faixas de 2.4 GHz.

GNU Radio [4] é um componente de *software open source* que provê ferramentas de processamento de sinal, sendo utilizada para programação do *SDR*. É possível programá-lo através de uma interface gráfica usando blocos, que são módulos e que podem exercer as mais variadas funções, como filtros, moduladores, conversores de domínio do sinal, etc. Os blocos podem conter *buffers* de entrada, *buffers* de saída ou ambos, sendo denominados *source blocks*, *sink blocks* e *general blocks*, respectivamente.

A flexibilidade da GNU Radio é um importante ponto da solução apresentada neste trabalho, pois além de entregar várias ferramentas de processamento de sinal por padrão, permite a criação de módulos personalizados. Através das linguagens Python e C++ é possível desenvolver um novo módulo ou modificar um pré-existente.

Apesar da flexibilidade e da facilidade de construção de novas soluções, o custo computacional da ferramenta é alto, pois o processamento de sinal é feito em *software* por linguagens de alto nível. Entretanto, a solução implementada no GNU Radio serve como protótipo para uma solução comercial, sendo que as novas funções desenvolvidas podem ser implementadas diretamente no *hardware* do dispositivo receptor.

2.5 Algoritmos de aprendizado de máquina

Algoritmos de aprendizado de máquina permitem diferenciar dispositivos. Isso é possível, pois nos dados dos mesmos é possível encontrar padrões que podem separar classes. No aprendizado supervisionado, os classificadores agrupam elementos de um conjunto com base em exemplos apresentados anteriormente. Alguns exemplos desses classificadores são *Support Vector Machines* (SVM), *K-Nearest Neighbor* (KNN) e *Linear Discriminant Analysis* (LDA). No aprendizado não supervisionado assume se que os elementos de um conjunto não são previamente rotulados, sendo que o objetivo do classificador é agrupar os elementos com base nas similaridades entre eles.

[3] https://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/ - Último acesso em 27/05/2022
[4] https://www.gnuradio.org/ - Último acesso em 25/05/2022

O algoritmo KNN é baseado em uma medida de similaridade entre os dados. A chave para algoritmos baseados em similaridade é encontrar uma métrica que possa ser mensurada pela distância, a fim de separar corretamente os dados em classes. O algoritmo KNN tem o objetivo de agrupar elementos com as mesmas características, decidindo a classe de um elemento com base no vizinho k que está mais próximo a ele [Keller, Gray e Givens 1985]. O valor de k no algoritmo depende do tipo de problema de classificação e do tamanho do conjunto de dados. A equação 2-9 apresenta a fórmula da distância euclidiana usada pelo KNN para dois elementos. Após calcular a distância entre os vizinhos, a maioria dos k vizinhos mais próximos será considerada como a classe resultante.

$$d(x,y) = \sqrt{\sum_{i=1}^{I}(x_i - y_i)^2} \tag{2-9}$$

O SVM é o algoritmo baseado em Kernel mais popular. Esse tipo de algoritmo mapeia os dados de entrada em um espaço vetorial de ordem superior, que facilita a resolução de problemas de classificação ou regressão. O SVM tem como objetivo criar um limite entre as classes que possibilite a previsão de rótulos de um ou mais vetores de características [Hearst et al. 1998]. Este limite é um hiperplano, que é orientado de tal forma que esteja o mais distante possível dos pontos de cada classe. Seja um conjunto de dados de treinamento rotulado:

$$(x_1, y_1), ..., (x_n, y_n), x_i \in R^d \tag{2-10}$$

onde x_i é uma representação de vetor de características e y_i o rótulo de classe binária de um composto de treinamento i. O hiperplano ótimo pode então ser definido como:

$$wx^T + b = 0 \tag{2-11}$$

onde w é o vetor de pesos, x é o vetor de características de entrada e b o viés. Os valores de b e w precisam satisfazer as seguintes desigualdades para todos os elementos do conjunto de treinamento:

$$\begin{aligned} wx_i^T + b \geq +1, sey_i = +1 \\ wx_i^T + b \leqslant -1, sey_i = -1 \end{aligned} \tag{2-12}$$

O objetivo de treinar um modelo SVM é encontrar b e w para que o hiperplano separe os dados e maximize a margem.

O classificador LDA é derivado dos princípios de Bayes, que envolve o cálculo da probabilidade para o evento pertencente à classe i, dada uma observação x. A Probabilidade ($P(i|x)$) é dada por:

$$P(i|x) = \frac{P(x|i)P(i)}{\sum_{\forall j} P(x|j)P(j)} \tag{2-13}$$

onde o numerador é o produto da probabilidade ($P(x|i)$) da observação x pertencente à classe i, multiplicada pela probabilidade anterior ($P(i)$) de qualquer amostra pertencente a classe i. O denominador é a probabilidade da observação ocorrer, independentemente da classe. O Teorema de Bayes é usado para classificação atribuindo o elemento à classe com a maior probabilidade, conforme a seguinte equação:

$$P(i|x)P(i) > P(x|j)P(j), \forall j \neq i \tag{2-14}$$

2.6 Trabalhos relacionados

Identificar dispositivos com base nas suas características eletromagnéticas não é um tema novo. Na Segunda Guerra Mundial controladores de voo analisavam visualmente ondas emitidas por radares para identificar possíveis transmissões que não vinham de seus equipamentos. Entretanto, os primeiros trabalhos na área de computação, que automatizaram esse processo foram publicados em 1995 [Toonstra e Kinsner 1995] e [Choe et al. 1995].

Os autores em [Brik et al. 2008] analisam as características únicas dos dispositivos IEEE 802.11 considerando seus sinais modulados. A avaliação envolveu 130 dispositivos idênticos. O trabalho investigou a quantidade mínima de dados necessária para o treinamento e concluiu que com 20 quadros é possível criar uma boa assinatura para um dispositivo, obtendo uma precisão de 99% de diferenciação. O trabalho também propõe um sistema, denominado PARADIS, que utiliza as características do sinal para autenticação do dispositivo. O sistema é externo aos dispositivos, não requerendo modificações nos transmissores. Além disso, por ser uma técnica simples, é possível conectá-la facilmente a um ponto de acesso.

[Köse et al. 2019] diferencia dispositivos IEEE 802.11 usando características no sinal transiente. O sinal transiente não leva nenhuma informação para a fase de demodulação, mas cada dispositivo tem um padrão e pode ser usado para diferenciar dispositivos. A vantagem de usar a parte transiente do sinal é a baixa latência, pois assim que um dispositivo começa a transmitir, já é possível coletar as características para análise. Entretanto, a técnica proposta requer a coleta de sinais a uma taxa de amostragem de 1G amostras por segundo, o que inviabiliza a aplicação em dispositivos *IoT*, visto que o custo dos dispositivos de captura com essa taxa de amostragem é elevado.

[Riyaz et al. 2018] explora o uso de Redes Neurais Convolucionais na classificação de dispositivos IEEE 801.11ac. Dois Rádios Definidos por *Software* são usados

nos experimentos, um transmitindo e o outro recebendo. As ondas são geradas na ferramenta MATLAB, onde as imperfeições também são adicionadas artificialmente, e transmitidas nos rádios. Os algoritmos de Redes Neurais Convolucionais apresentam desempenho muito superior aos demais algoritmos analisados. O trabalho analisa o impacto da distância na classificação, obtendo uma precisão maior que 95% em distâncias menores que 10 metros. A técnica proposta utiliza características coletadas de qualquer parte do sinal, formando um vetor de 256 amostras, com amplitude instantânea dos componentes do sinal em fase e quadratura.

[Peng et al. 2016] propõe uma técnica baseada na Figura de Traço de Constelação Diferencial (*DCTF*) para identificação de dispositivos. A *DCTF* independe da modulação e pode ser utilizada para identificar diversas tecnologias de IoT e, por isso, esta técnica tem sido utilizada em diversos trabalhos.

LoRaWAN é um protocolo de rede que usa o esquema de modulação *chirp spread spectrum (CSS) scheme* e implementa mecanismos de segurança específicos baseados em chaves criptográficas à tecnologia LoRa. [Tomasin, Zulian e Vangelista 2017] identificou vulnerabilidades no protocolo LoRaWAN, mostrando ser possível executar um ataque *replay* contornando o mecanismo de associação do dispositivo com o Gateway, gerando números aleatórios de pacotes enviados anteriormente pelo dispositivo. Motivado por isso, [Jiang et al. 2019] identifica dispositivos LoRa usando a técnica *DCTF*. As imagens obtidas são classificadas usando um algoritmo de reconhecimento de imagem. Nos experimentos, o trabalho obteve uma precisão de 99,6% utilizando uma taxa de amostragem de 10MS/s. [Jiang et al. 2019] também avalia a influência da distância entre o transmissor e o receptor na precisão da classificação. O método funciona melhor quando é treinado com sinais de locais diferentes.

O trabalho de [Sankhe et al. 2019] tem o objetivo de diferenciar rádios semelhantes usando assinaturas extraídas a partir de imperfeições no *hardware*. Os dispositivos emissores são SDRs modelo USRP X310, que transmitem quadros IEEE 802.11a (*WiFi*). O sinal é capturado por um SDR USRP B210 a uma taxa de amostragem de 5MS/s na frequência central de 2,45 GHz. O conjunto de dados gerado[5] contém 20 milhões de amostras de cada um dos 16 dispositivos. A coleta foi feita em uma área aberta, variando a distancia entre o transmissor e receptor de 0,6 à 19,5 metros.

[Reus-Muns et al. 2020] tem o objetivo de propor uma autenticação de Estações Base 5G (Base Station - *BS*) através da técnica de assinatura radiométrica. Os sinais são coletados de 4 SDRs USRP X310, cada um representando uma BS, na plataforma POWDER[6], mas neste trabalho foi utilizado somente a tecnologia *WiFi*. Os quadros gerados

[5] https://genesys-lab.org/oracle - Último acesso em 27/05/2022
[6] https://powderwireless.net/ - Último acesso em 29/07/2022

por esses dispositivos são das tecnologias IEEE 802.11a (WiFi), LTE e 5G-NR e os autores utilizaram redes neurais profundas no processo de classificação. Além de detectar as Estações Base, o trabalho propõe categorizar os dispositivos em níveis de confiabilidade baseado em um *score*. A classificação em uma rede neural é baseada na função de ativação *softmax score*, essa que representa a distribuição de probabilidade para as classes. Se a probabilidade for baixa, a *BS* é categorizada como não confiável e novos dados do sinal do dispositivo são requisitados para uma nova tentativa de verificação. O artigo apresenta bons resultados com a técnica, mostrando que os dispositivos corretamente classificados possuem *score* próximo de 1 e os incorretamente classificados têm um score médio de 0.118. Embora a técnica de categorização represente uma inovação para a área, os autores não exploram o uso da técnica para classificação de dispositivos não treinados pela rede neural.

O trabalho de [Soltani et al. 2020] usa uma técnica de aumento de dados para simular diferentes condições no canal de transmissão e variações de ruído. Essa técnica é aplicada a uma base de dados de 10 rádios virtuais simulados através da ferramenta MATLAB. Cada rádio transmite quadros WiFi, gerando uma base com 2042 quadros, sendo disponibilizada publicamente [7].

A Tabela 2.1 apresenta uma síntese dos diferentes trabalhos encontrados na literatura, fazendo uma comparação entre eles. Na terceira coluna, são apresentadas as partes do sinal utilizada na classificação, que podem ser de três diferentes tipos. O sinal modulável é a parte que contém os dados a serem extraídas no processo de demodulação. A parte estável além de conter dados também abrange o preâmbulo, que em geral é utilizado para sincronização, excluindo o sinal transiente. Na quarta coluna são apresentadas as técnicas de identificação e a quinta coluna indica se a técnica proposta é aplicável em um sistema online.

Tabela 2.1: Comparação de técnicas de identificação

Trabalho	Tecnologias	Parte do sinal	Técnica de identificação	*Online*
[Brik et al. 2008]	802.11 NICs	Modulável	Erros de modulação	
[Köse et al. 2019]	WiFi	Transiente	Espectro de Energia	
[Riyaz et al. 2018]	802.11ac	Estável	Erros de Modulação	
[Peng et al. 2016]	ZigBee	Estável	*DCTF*	
[Jiang et al. 2019]	LoRa	Estável	*DCTF*	
[Sankhe et al. 2019]	WiFi	Estável	Erros de modulação	
[Reus-Muns et al. 2020]	WiFi	Parte do sinal modulável	Erros de modulação	✓
[Soltani et al. 2020]	WiFi	-	Erros de modulação	
Nosso	LoRa, ZigBee e WiFi	Estável	*DCTF*	✓

[7] https://genesys-lab.org/dataaugmentation - Último acesso em 27/05/2022

Defender dispositivos ou demonstrar mecanismos que comprometem a privacidade dos mesmos são dois temas amplamente estudados na literatura. Apesar de ser uma linha de pesquisa consolidada, onde artigos vêm mostrando altos níveis de identificação de dispositivos, poucos artigos se dedicam a explorar o uso das técnicas de diferenciação de dispositivos em aplicações.

Encontramos na literatura alguns trabalhos que propõem o uso da técnica de identificação de dispositivos em um sistema de autenticação. Por exemplo, [Reus-Muns et al. 2020] apresenta um fluxo de autenticação em uma rede 5G. Entretanto, este e outros trabalhos não analisam o custo computacional da técnica, o tempo de resposta e a heterogeneidade de tecnologias.

CAPÍTULO 3

Proposta e metodologia

Este Capítulo descreve o sistema *online* para detecção de dispositivos intrusos em Internet das Coias. Na Seção 3.1 é apresentada a proposta do método *online*. A arquitetura do sistema é descrita na Seção 3.2. As seções seguintes descrevem a metodologia utilizada no trabalho, sendo que, o conjunto de módulos de processamento de características é descrito na Seção 3.3, e por último, os processos de treinamento e predição são descritos na Seção 3.4.

3.1 Método online

No sistema proposto para identificação de dispositivos, assume-se que, o dispositivo já foi autenticado por outro mecanismo. Assume-se também que um modelo de aprendizado de máquina é treinado de maneira *offline* a partir da assinatura eletromagnética dos dispositivos. Em todo o tempo ou de tempos em tempos, o sinal eletromagnético é amostrado e repassado para o sistema de autenticação. Esse sinal é processado e dele são extraídas as características, que serão utilizadas na identificação do dispositivo. As características extraídas são repassadas ao modelo, que vai retornar se o dispositivo é verídico ou não.

A Figura 3.1 apresenta o fluxo do processo de classificação. Esse fluxo foi baseado no trabalho de [Reus-Muns et al. 2020], que introduz a ideia de *scores* para categorizar bases 5G como confiáveis ou não. Seja D' o identificador informado pelo dispositivo e D um dispositivo previamente conhecido pelo classificador, onde D' é igual a D. Existem duas possibilidades, o dispositivo rotulado D' é realmente D ou ele é um intruso, considerando que o identificador do dispositivo pode ser forjado. O objetivo é determinar se D' é quem ele diz ser, baseado na assinatura gerada a partir do seu sinal eletromagnético. A assinatura de D' é passada para o classificador e ele retorna a classe C que mais se assemelha à D', juntamente com o *score* S da classificação.

Suponhamos que D^i seja um intruso se passando por D, existem duas possibilidades. A primeira delas, ilustrada na Figura 3.2(a), é quando a classe C retornada pelo classificador difere de D^i. Isso acontece, pois, a assinatura de D^i se assemelha mais à as-

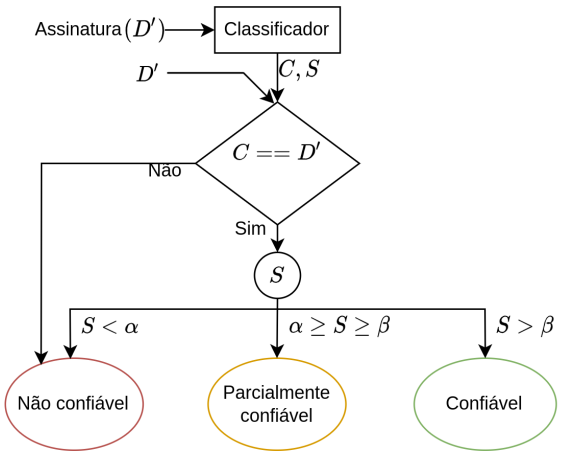
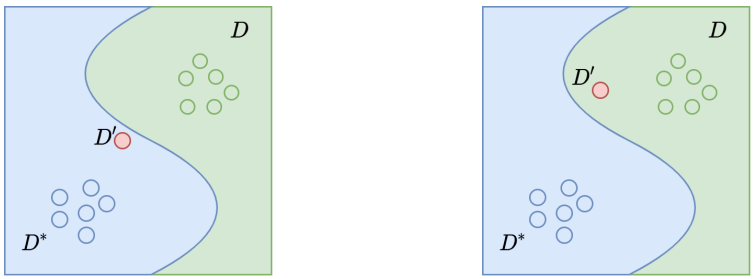


Figura 3.1: Diagrama do processo de classificação.

sinatura de outro dispositivo conhecido. Entretanto, apesar do dispositivo corresponder a uma classe conhecida pelo classificador, ele é categorizado como não confiável. Esse caso também contempla o cenário em que um dispositivo conhecido tenta se passar por outro dispositivo conhecido. O segundo caso, ilustrado na Figura 3.2(b), é quando a assinatura de D^i se assemelha mais à D. Portanto, precisamos de mais uma informação, além de C, para definir se D^i é ou não um intruso.



(a) *Classificado como* D^* (b) *Classificado como* D

Figura 3.2: Dispositivo intruso.

Os algoritmos de classificação normalmente implementam uma função de decisão baseada em um *score*, calculado para cada classe. Este *score* pode representar a probabilidade de uma entrada pertencer a determinada classe. A decisão da saída do algoritmo é realizada com base na maior probabilidade obtida. A partir destas probabilidades, é possível saber o grau de semelhança entre o objeto a ser classificado e seu rótulo. Portanto, esta informação pode ser utilizada para inferir se um dispositivo analisado é ou não quem ele disse ser, com base em um limiar. A ideia é que, dispositivos diferentes possuem um *score* baixo quando comparados na classificação. Se $C == D'$, é preciso analisar o *score* da classificação. A assinatura de D' pode se assemelhar mais à D e o *score* vai dar o grau de semelhança entre as assinaturas.

Conforme mostrado na Figura 3.1, o dispositivo pode ser categorizado como não confiável, parcialmente confiável ou confiável. No primeiro caso, visto que o *score* de classificação é muito baixo, $S < \alpha$, o dispositivo D é considerado não confiável. Caso o *score* fique em uma faixa intermediária, a confiança é parcial. No caso último caso, o *score* de classificação é alto e, portanto o resultado da classificação é um dispositivo confiável. A proposta apresentada para o sistema *online* é que os valores α e β sejam definidos pelo administrador do sistema.

Para fortalecer a segurança geral do processo de autenticação, o paradigma de Autenticação Contínua vem sendo empregado. A Autenticação Contínua foi introduzida para propor novos mecanismos de validação da identidade de usuários, atacando os problemas não resolvidos nas técnicas usuais de autenticação [Traore 2011]. O processo de autenticação consiste em criar uma assinatura baseada em dados gerados pelo usuário e coletar continuamente dados para comparar com esta assinatura. Até o momento, nenhum trabalho na literatura propôs o uso desta técnica para autenticação de dispositivos de Internet das Coisas.

Esta forma de autenticação é direcionada, principalmente, aos cenários em que o dispositivo já realizou a autenticação inicial através de algum outro sistema (e.g, através de chaves criptográficas). Por se tratar de uma coleta de dados, a assinatura gerada pode sofrer alterações devido às condições do ambiente, sendo necessário um novo treinamento de tempos em tempos.

3.2 Arquitetura do sistema

Um sistema de Internet das Coisas é composto de dispositivos e aplicações. Pensando nas tecnologias de comunicação, os dispositivos podem ser de diversas tecnologias e.g WiFi, ZigBee e LoRa e geralmente são escolhidas de acordo com o cenário da aplicação. As aplicações podem receber, enviar e ou processar os dados dos dispositivos. Nessa comunicação é preciso identificar e autenticar o dispositivo que que está enviando enformações.

A Figura 3.3 apresenta uma visão geral do sistema *online* de detecção de intrusos proposto neste trabalho. O sistema foi pensado para ser modular, visando atender à heterogeneidade de dispositivos e aplicações e facilitar a modificação do sistema para obtenção de melhores resultados no processo. Além disso, também possui uma interface de gerenciamento, onde é possível interagir com os módulos, alterando parâmetros sem precisar fazer modificações no código.

O sinal transmitido pelo dispositivo é coletado por um SDR, sendo o módulo Leitura das amostras responsável por fazer a interface entre o SDR e o sistema. Nesse ponto o fluxo é dividido em dois outros fluxos, que são executados paralelamente,

sendo que o objetivo do primeiro fluxo é extrair o identificador do dispositivo e.g., endereço MAC. O objetivo do segundo fluxo é identificar o dispositivo baseado nas características do seu sinal. As saídas destes dois fluxos são unidas no bloco Decisão, esse que é responsável por categorizar o dispositivo como intruso ou não intruso, baseado no identificador do dispositivo, na classe predita e no *score* de classificação.

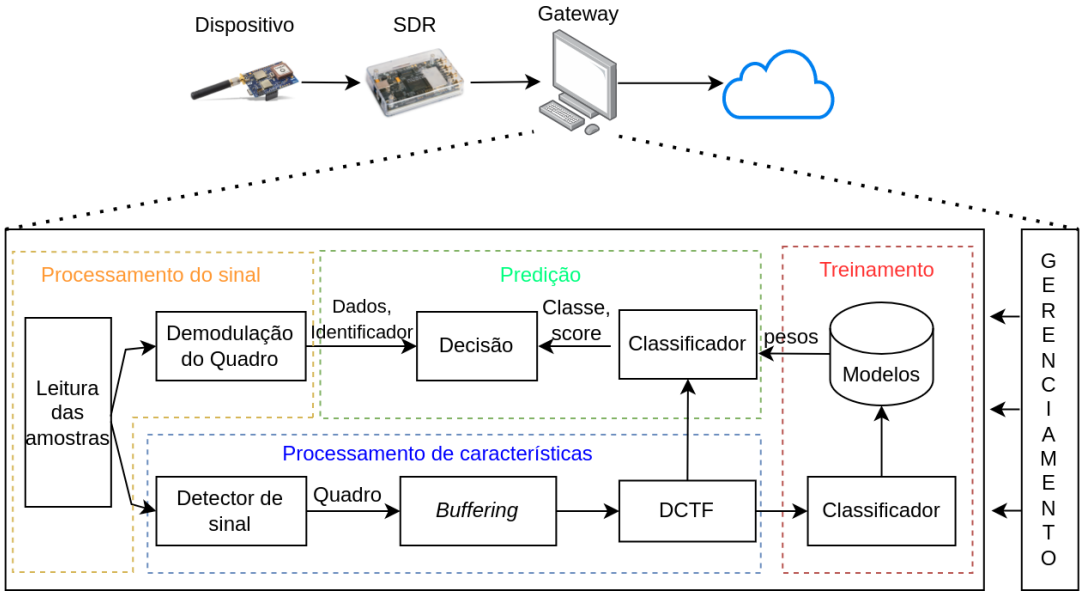


Figura 3.3: Visão geral do sistema de detecção *online*.

A arquitetura aqui proposta tem o objetivo de ser independente de ferramenta. Embora neste trabalho tenha sido utilizado o GNU Radio para implementar a solução, outras ferramentas como o MATLAB[1] também podem ser utilizadas. As próximas seções descrevem a implementação desta arquitetura na ferramenta GNU Radio.

O processo de aquisição do sinal é realizado utilizando um Rádio Definido por Software e a ferramenta GNU Radio é utilizada no processamento do sinal. A Figura 3.4 apresenta o bloco *osmocom Source*[2], que faz interface entre o SDR e o GNU Radio. Este é um bloco que não faz parte da biblioteca padrão do GNU Radio mas pode ser baixado e instalado. O bloco tem como parâmetros a frequência central e a taxa de amostragem, que são configurados de acordo com a tecnologia. A saída do bloco é um fluxo de dados contínuo de vetores complexos de duas posições, cada uma representado os valores I e Q no formato de ponto flutuante.

[1] https://www.mathworks.com/products/matlab.html - Último acesso em 27/05/2022
[2] https://github.com/osmocom/gr-osmosdr - Último acesso em 27/05/2022

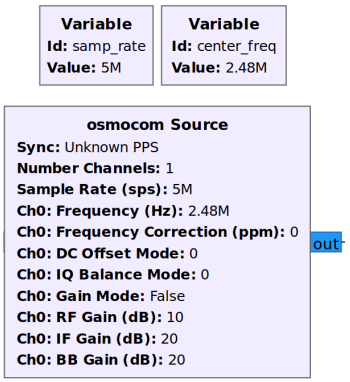


Figura 3.4: Bloco do GNU Radio usado na coleta do sinal.

As amostras do sinal coletadas podem ser salvas em arquivos e carregadas posteriormente, o que possibilita criar uma base de dados com amostras do sinal de todos dispositivos. Esse método foi utilizado para gerar duas bases de dados, que serão descritas no próximo Capítulo. Os arquivos são salvos no formato *Signal Metadata Format* (Sigmf) [Hilburn et al. 2018], onde as sequências de dados são representadas em formato binário em arquivos .bin. Cada arquivo é uma representação da transmissão na forma de valores I e Q intercalados. As amostras estão no formato ponto flutuante de 32 bits e cada valor I ou Q ocupa 4 bytes, conforme ilustrado na Figura 3.5.

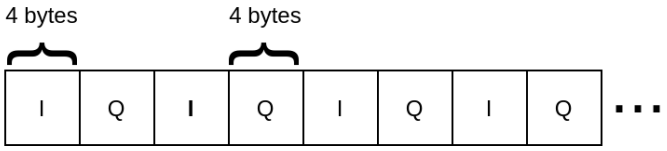


Figura 3.5: Um exemplo de valores de I/Q intercalados em um arquivo .bin.

A ferramenta GNU Radio pode ser utilizada para aquisição e processamento de sinais de maneira *offline*, como descrito anteriormente, ou de maneira *online*. Para isso, é preciso somente adaptar o código, criando novos blocos, ou utilizando os já existentes, e conectá-los à saída do bloco usado na coleta.

3.3 Processamento de características

A leitura das amostras gera um fluxo de dados contínuo, onde o sinal coletado contém partes em que há transmissão e que não há, portanto, é preciso discriminar e extrair os quadros. Existem vários métodos de detecção de transmissão e para simplificar foi escolhido o mais básico, que atende o cenário do trabalho, que é o método de detecção por nível de amplitude [Xuping e Jianguo 2007]. É possível utilizar este método de detecção porque a aquisição do sinal foi feita em um ambiente controlado e não sofre

interferência de outros sinais na mesma frequência. Em um ambiente em que há dois ou mais dispositivos transmitindo nas mesma frequência, a técnica não funcionaria e seria necessária uma técnica mais avançada, como o uso de preâmbulo para detecção do início do quadro no WiFi [Thakur e Khare 2013].

O algoritmo utilizado é baseado em uma janela deslizante, onde o início da transmissão é detetada comparando a saída do detector de energia (Amplitude) com um limite (*threshold*) que depende do nível de ruído. Assim que a Amplitude ultrapassa um certo nível, um *buffer* começa a ser preenchido com as amostras do sinal, até atingir o tamanho de um quadro. A Figura 3.6 ilustra este processo.

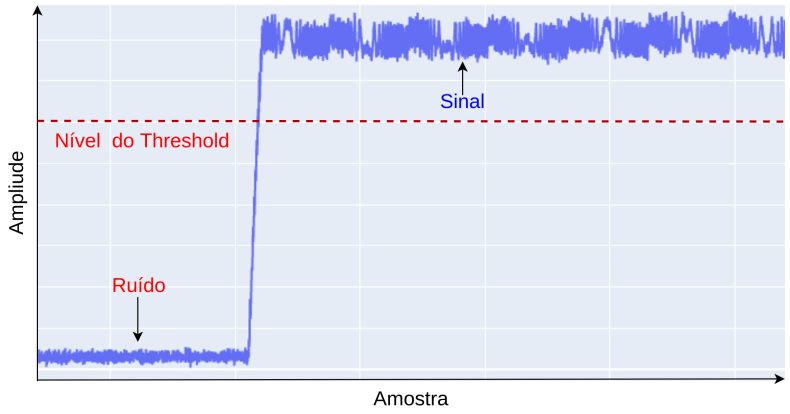


Figura 3.6: Método de detecção de transmissão por nível de amplitude.

O Algoritmo 1 implementa a técnica de janela deslizante para a extração dos quadros a partir do sinal coletado. A entrada do algoritmo é um vetor *Samples* de amostras do sinal coletado pelo *SDR*, o limite *Threshold* usado para detectar o início do sinal e o tamanho do quadro *FrameLength*, específico de cada tecnologia. O vetor *Samples* é percorrido até encontrar uma amostra onde o nível de amplitude é maior que *Threshold*. O *buffer* começa a ser preenchido até atingir o tamanho de um quadro ou o vetor de amostras atingir seu final. Por último, se o tamanho do *buffer* for exatamente o tamanho de um quadro, ele é retornado, caso contrário, é retornado um vetor vazio, indicando que não foi detectado um quadro ou o quadro é inválido.

Assim que o *buffer* é preenchido com as amostras de um quadro, o fluxo continua e os dados são repassados para o próximo módulo. Uma DCTF é gerada para cada quadro obtido do dispositivo conforme a Equação 2-8, descrita do Capítulo anterior. O Algoritmo 2 apresenta a função que gera a DCTF a partir das amostras do quadro.

A última operação realizada, antes da classificação, é a extração da matriz de densidade a partir da DCTF. Cada amostra I/Q é normalizada pela amplitude máxima A e a amplitude miníma B, sendo a dimensão da matriz M x N, conforme a Equação 3-1.

Algoritmo 1: FrameExtraction(*Samples*, *Threshold*, *FrameLength*)

```
Result: List of frames
Buffer ← {};
SampleIndex ← 0;
while SampleIndex < Length(Samples) and Length(Buffer) < FrameLength
 do
    if amplitude(Samples[SampleIndex]) > Threshold then
        if | frame |< FrameLength then
            add Samples[SampleIndex] in Buffer;
            SampleIndex = SampleIndex + 1;
        end
    end
    else
        SampleIndex = SampleIndex + 1;
    end
end
if Length(Buffer)! = FrameLength then
    return 0
end
else
    return Buffer;
end
```

$$m = \frac{d_i(t) - B}{A - B} * M, n = \frac{d_q(t) - B}{A - B} * M \tag{3-1}$$

3.4 Treinamento e predição

O sistema online possui duas fases. Na primeira, o classificador precisa ser treinado com dados coletados dos dispositivos. Esse processo precisa ser realizado em uma fase anterior à predição e pode ser executado outras vezes ao longo do funcionamento do sistema. Algumas motivações para esse re-treino são a adição de novos dispositivos na base de dados, mudança nas condições físicas do ambiente, como temperatura e ruído, e possíveis alterações no hardware dos dispositivos, causadas pelo uso, no tempo de vida do mesmo. O classificador utilizado para treinamento gera um modelo que precisa ser armazenado. Esse modelo é salvo em um arquivo e representa o que foi aprendido pelo algoritmo de aprendizado de máquina.

A entrada do classificador são as DCTFs. O conjunto de dados é dividido em duas partes, uma utilizada na etapa de treinamento, contendo 20% dos dados, e os outros 80% são usados na validação. A etapa de validação ocorre de maneira *offline*, com dados previamente coletados dos dispositivos, e serve para medir a eficácia do modelo gerado.

Algoritmo 2: DCTFGeneration($frame, n$)

Result: DCTF of frame

1 $dctf \leftarrow \{\}$;
2 FrameIndex $\leftarrow 0$;
3 **while** $FrameIndex <$ | $frame$ | - n **do**
4 $dctf[FrameIndex] \leftarrow$
$frame[FrameIndex] * conjugate(frame[FrameIndex * n])$
5 **end**
6 return $dctf$;

A fase de predição acontece *online*. O classificador lê o arquivo que contém o modelo e carrega-o para memória. A partir deste momento, dados novos podem ser injetados no classificador. Como apresentado de mameira teórica na Seção 3.1, a saída do classificador são dois dados, um contendo a classe do dispositivo e outra contém o *score* de classificação. Essas informações são utilizadas no processo de decisão da ferramenta.

CAPÍTULO 4

Avaliação

Neste capítulo, é apresentada a avaliação do sistema, que está dividida em duas partes. Na Seção 4.1 são apresentados os experimentos *offline*, que visam avaliar a técnica aplicada a diversas tecnologias e a robustez às variações do ambiente. Na Seção 4.2 é feita uma avaliação do sistema *online*, apresentando experimentos que mostram a viabilidade da técnica em detectar dispositivos desconhecidos.

4.1 Avaliação offline

Ao longo do processo de revisão da literatura, uma busca por conjuntos de dados contendo sinais eletromagnéticos de dispositivos foi realizada. Grande parte dos trabalhos encontrados não disponibilizam publicamente os conjuntos de dados utilizados em seus experimentos. Portanto, para realizar uma análise mais detalhada e testar a técnica de classificação em tecnologias de comunicação variadas, foram coletados sinais de dispositivos das tecnologias LoRa e ZigBee.

O sinal LoRa foi gerado por um módulo transmissor ESP32 LoRa da empresa Heltec, sendo a frequência central configurada em 915 MHz, o fator de espalhamento igual a 7 e a largura de banda igual a 125kHz. O intervalo de envio de um quadro é de 50ms e a carga útil do pacote contém dados aleatórios.

O sinal ZigBee foi gerado por um módulo transmissor modelo Telos B da empresa Crossbow. A frequência central de transmissão é 2,48GHz, com uma largura de banda de 2MHz. A carga útil dos pacotes, semelhante ao LoRa, é composta de dados aleatórios.

A coleta de sinais foi realizada através de uma Rádio Definido por Software modelo Nuand bladeRF x40, apresentado na Figura 4.1(c), gerando amostras I/Q representadas como vetores de números complexos com 4 bytes cada. A taxa de amostragem usada foi de 2,5MS/s para o LoRa e 5MS/s para o ZigBee. A Tabela 4.1 apresenta os parâmetros usados na aquisição do sinal para cada tenologia e as Figuras 4.1(a) e 4.1(b) apresentam as imagens dos dispositivos LoRa e ZigBee, respectivamente.

Tabela 4.1: Parâmetros usados na aquisição do sinal

Parâmetro	ESP32 Heltec LoRa	ZigBee Telos B
Frequência do transmissor	915MHz	2,48GHz
Taxa de amostragem	2MS/s	10MS/s
Receptor	bladeRF x40	bladeRF x40

(a) *ESP32 Heltec LoRa* (b) *ZigBee Telos B* (c) *Nuand bladeRF x40*

Figura 4.1: Dispositivos envolvidos na coleta.

A Figura 4.2 apresenta um diagrama dos blocos do GNU Radio utilizados na aquisição do sinal. São utilizados dois blocos. O primeiro deles faz a interface com o *SDR* e nele são configurados os parâmetros de captura. O segundo recebe como entrada o sinal coletado e armazena em um arquivo.

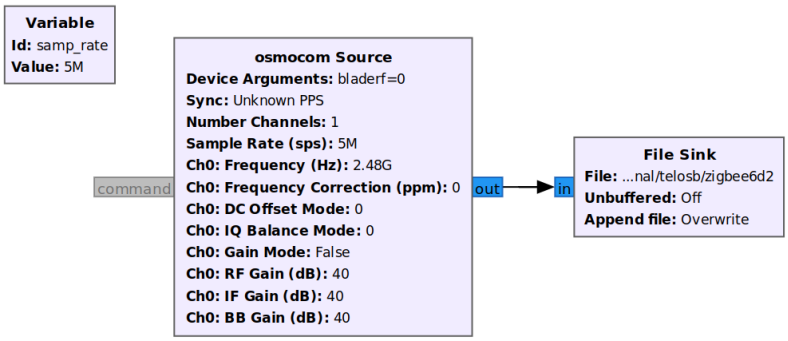


Figura 4.2: Blocos usados para coleta do sinal no GNU Radio.

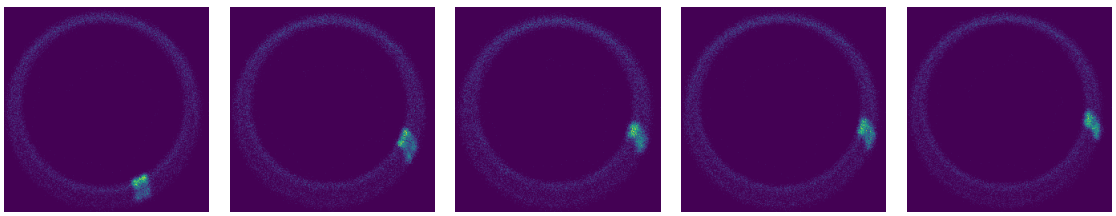
Os sinais dos dispositivos LoRa foram coletados em cinco distâncias diferentes. No primeiro caso, o transmissor e o receptor foram posicionados a uma distância de 1

metro entre eles. A cada nova coleta, essa distância foi dobrada até o limite de 16 metros. Portanto, as distâncias analisadas foram 1,2,4,8 e 16 metros. A duração da coleta em cada distância foi de 5 minutos. O resultado é uma coleção com 270 quadros de cada um dos 5 dispositivos, em cada distância, totalizando 1350 quadros por dispositivo.

Os sinais dos dispositivos ZigBee foram coletados a uma distância fixa de 1 metro entre o transmissor e o receptor. Em testes realizados antes da coleta dos dados, percebeu-se que a potência do sinal recebido diminuía consideravelmente a medida que o transmissor se afastava. Por essa limitação, não foi possível coletar sinais em diferentes distâncias. A coleção contém 450 quadros de cada um dos 8 dispositivos.

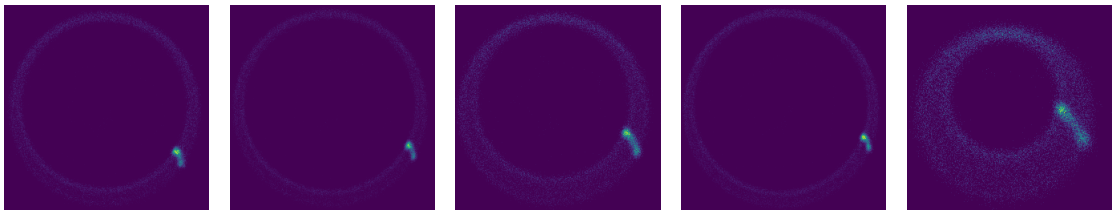
Embora alguns dos trabalhos recentes e.g., [Riyaz et al. 2018] utilizem o estado da arte das Redes Neurais Artificiais Profundas (DNN - *Deep Neural Network*), alguns algoritmos menos complexos foram escolhidos para realizar os experimentos neste trabalho. Os algoritmos de DNN possuem uma alta complexidade de tempo e demandam de um alto poder computacional na etapa de treinamento. O objetivo foi de utilizar classificadores simples nos testes com as coleções encontradas na literatura. A aplicação dos classificadores nos testes envolvendo nossa coleção, se justifica pelo requisito mínimo tempo de resposta na aplicação da técnica na autenticação de dispositivos.

Como é possível observar na Figura 2.5, as regiões de densidade de cada dispositivo formam *clusters* em diferentes posições na DCTF. Como mostrado em [Jiang et al. 2019], um algoritmo de clusterização poderia ser utilizado para agrupar dados de um dispositivo. Entretanto, em nossos testes, percebemos que a temperatura do dispositivo tem influência em sua DCTF, como pode ser visto na figura 4.3. Isso acontece, pois os osciladores, responsáveis por produzirem as ondas eletromagnéticas nos dispositivos, sofrem variações de frequência com a temperatura [Farahvash, Quek e Mak 2008], contribuindo para o efeito *CFO*. A DCTF de um dispositivo também muda conforme a distância entre o transmissor e o receptor, ilustrada na figura 4.4. Logo, são necessários algoritmos que aprendam características além do posicionamento da região de maior densidade.



(a) $t = 0s$ (b) $t = 2s$ (c) $t = 4s$ (d) $t = 6s$ (e) $t = 8s$

Figura 4.3: Efeito da temperatura na DCTF de um dispositivo.



(a) $d = 1m$ (b) $d = 2m$ (c) $d = 4m$ (d) $d = 8m$ (e) $d = 16m$

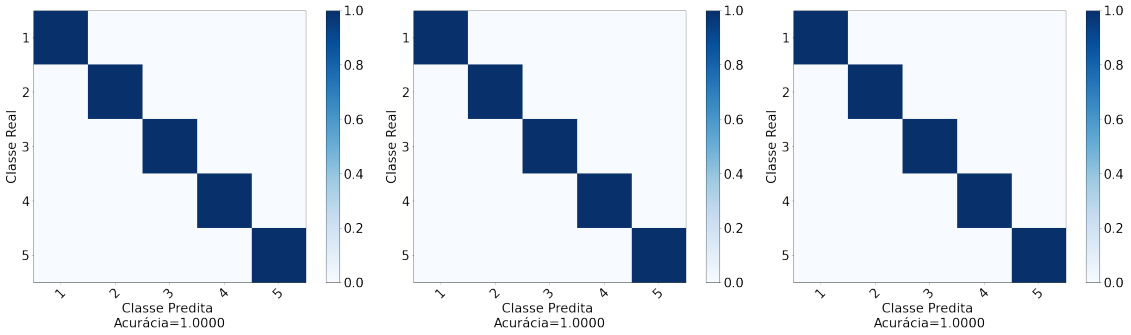
Figura 4.4: Efeito da distância na DCTF de um dispositivo.

Em todos os experimentos apresentados a seguir, a coleção foi dividida em duas partes. A primeira destas partes foi utilizada na etapa de treinamento dos classificadores e corresponde à 80% do total de dados. Os 20% restantes são utilizados na etapa de teste. Essa divisão foi realizada aleatoriamente.

4.1.1 Coleções de dados coletadas localmente

No primeiro experimento, a coleção LoRa foi avaliada através dos algoritmos KNN, SVM e LDA. Estes três algoritmos foram escolhidos, pois possuem características de funcionamento diferentes um dos outros. O SVM é um modelo que aproxima qualquer função não linear contínua e pode ser visto como uma generalização da *Multilayer Perceptron*. Os parâmetros do SVM foram definidos através de testes de *model selection*, onde o *kernel* que apresentou melhores resultados foi o linear. O LDA é uma técnica que tenta encontrar uma combinação linear de características, separando duas ou mais classes. O KNN é uma técnica baseada em análise de vizinhança, que usa aproximação agrupando pontos de dados próximos. O valor do parâmetro K do KNN foi escolhido através de testes e foi usado um $k = 3$.

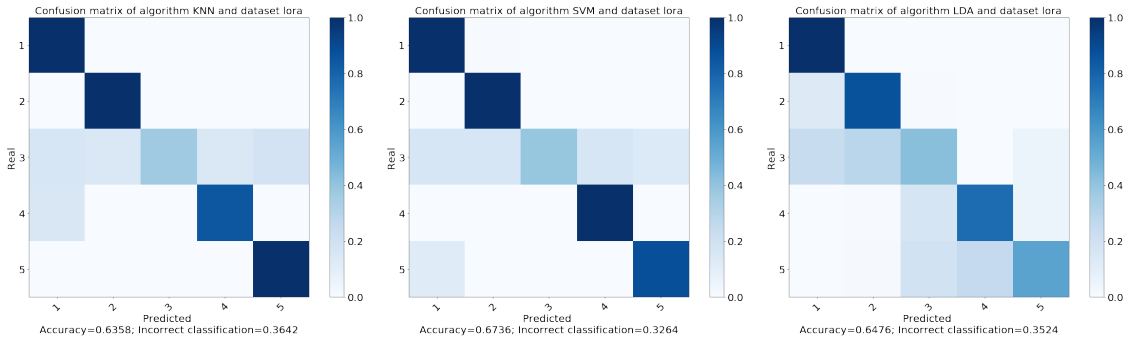
Cada elemento da coleção contém 25 000 amostras, que correspondem ao tamanho do quadro LoRa nas configurações definidas anteriormente. Os algoritmos foram treinados e testados com dados coletados nas 5 distâncias descritas anteriormente. A acurácia dos classificadores foi de 100% em todos testes, semelhante ao obtido em outros trabalhos na literatura [Jiang et al. 2019]. A Figura 4.5 apresenta as matrizes de confusão para os algoritmos aplicados nos dispositivos LoRa.



(a) *KNN* (b) *SVM* (c) *LDA*

Figura 4.5: Resultados LoRa com algoritmos treinados em todas distâncias.

O cenário muda quando os algoritmos são treinados somente com dados de uma distância. A Figura 4.6 apresenta os resultados onde os algoritmos são treinados com dados coletados na distância de 1 metro, enquanto na fase de testes são usados dados coletados em todas as distâncias. Este experimento reflete o que seria um ambiente real, onde os dispositivos podem alterar suas posições. Neste caso, a taxa de acertos dos algoritmos varia entre 63,5% (KNN) à 67,3% (SVM).



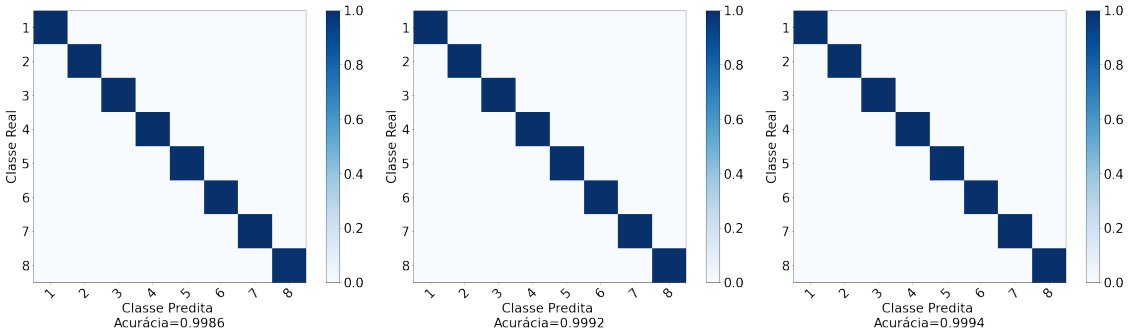
(a) *KNN* (b) *SVM* (c) *LDA*

Figura 4.6: Resultados LoRa variando a distância.

A Figura 4.7 apresenta as matrizes de confusão para os algoritmos KNN, SVM e LDA aplicados nos dispositivos ZigBee. Nos três algoritmos obtivemos uma acurácia de 99,8%, sendo que 6 quadros dentre os 3600 foram classificados incorretamente. Neste teste, todos os dispositivos estavam em uma única posição.

4.1.2 Coleções de dados da literatura

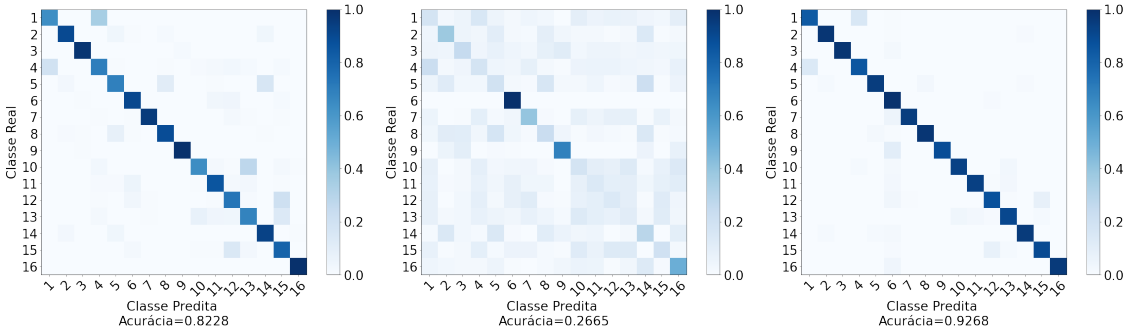
Ao realizar uma busca por coleções de dados com sinais de dispositivos, foram encontradas três trabalhos [Sankhe et al. 2019], [Reus-Muns et al. 2020], [Soltani et al. 2020], apresentados no Capítulo 2.



(a) *KNN* (b) *SVM* (c) *LDA*

Figura 4.7: Resultados Zigbee.

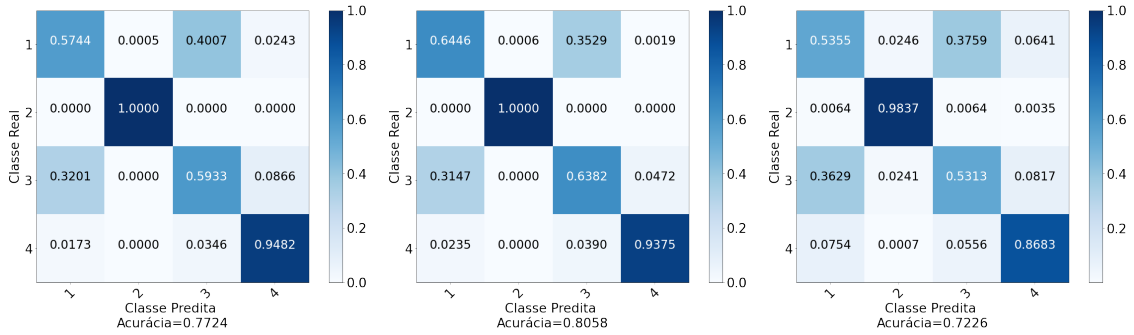
Os dados do dataset ORACLE foram coletados a partir de 16 diferentes SDRs do modelo USRP, simulando dispositivos WiFi. Os quadros transmitidos foram gerados através do MATLAB, sendo que o receptor amostrou o sinal a uma taxa de 5MS/s na frequência central de 2,4GHz. A Figura 4.8 apresenta as matrizes de confusão obtidas no experimento com a coleção ORACLE [Sankhe et al. 2019]. Os classificadores LDA e KNN apresentaram o melhor resultado, com acurácia de 93,7% e 82,7% respectivamente. No teste com o classificador SVM a acurácia foi de 26,7%. O resultado obtido com o LDA é 4,9% inferior ao resultado obtido em [Sankhe et al. 2019], com um classificador CNN.



(a) *KNN* (b) *SVM* (c) *LDA*

Figura 4.8: Resultados para o conjunto de dados ORACLE.

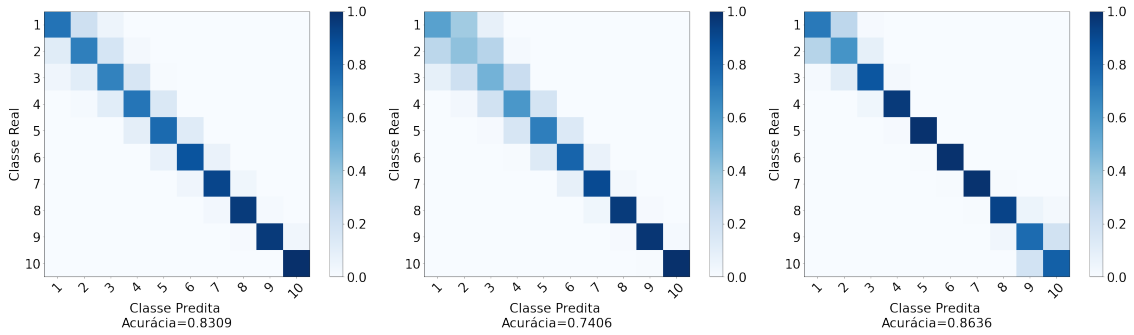
A Figura 4.8 apresenta as matrizes de confusão obtidas no experimento com a coleção POWDER [Reus-Muns et al. 2020]. No teste com o classificador SVM obtivemos o melhor resultado (81,7%), seguido pelo classificador KNN (78,9%). Neste experimento, ao contrário dos demais, o algoritmo LDA teve a pior desempenho.



(a) *KNN* (b) *SVM* (c) *LDA*

Figura 4.9: Resultados conjunto de dados POWDER.

A Figura 4.10 apresenta as matrizes de confusão obtidas no experimento com a coleção Data Augmentation [Soltani et al. 2020]. O classificador LDA obteve o melhor resultado (86,1%), seguido pelo KNN (82,9) e SVM (75,3%).



(a) *KNN* (b) *SVM* (c) *LDA*

Figura 4.10: Resultados conjunto de dados Data Augmentation.

A Tabela 4.2 sintetiza os resultados obtidos nos experimentos com as coleções encontradas na literatura, fazendo uma comparação os resultados obtidos nos trabalhos que propõem estas coleções.

Tabela 4.2: Resultados dos experimentos com coleções encontradas na literatura.

Coleção	WiFi (ORACLE)	WiFi (POWDER)	Genesis Augmentation
Número de dispositivos	16	4	10
Taxa de amostragem	5MS/s	5MS/s	Não informada
Acurácia do trabalho	98,6% (CNN)	99,9% (CNN)	99% (CNN)
Acurácia DCTF	93,7 (LDA), 82% (KNN) e 26% (SVM)	81,7% (SVM), 78,9% (KNN) e 53% (LDA)	86,1 (LDA), 82,9% (KNN) e 75,3% (SVM)

4.2 Avaliação online

Nesta Seção será apresentada a avaliação da ferramenta online. Os testes têm o objetivo de mostrar a flexibilidade da ferramenta, por exemplo, a possibilidade do usuário escolher o algoritmo de classificação e a taxa de amostragem. A Figura 4.11 apresenta os blocos do GNU Radio usados nos testes da solução *online*. Os dados utilizados na avaliação *online* são os mesmos coletados dos dispositivos LoRa e ZigBee, estes que foram descritos na seção anterior. O classificador foi previamente treinado com os dados dos dispositivos, gerando um modelo, que é salvo e carregado no bloco *DeviceClassifier*.

Para executar os experimentos, o bloco *File Source* que lê os arquivos com os quadros dos dispositivos e injeta-os no fluxo. O uso de arquivos foi escolhido para facilitar a replicação dos testes, como mesmos dados, entretanto o bloco de leitura pode ser substituído por um bloco que faz interface com o SDR. Em seguida o fluxo é dividido em dois, sendo o primeiro responsável por identificar e classificar o quadro e o segundo responsável pela demodulação dos quadros. Assim que é terminado o processamento de ambos fluxos, o bloco *Verifier* junta a informação do classificador com o identificador do quadro e faz a comparação.

Os experimentos deste Capítulo foram realizados em um computador com processador Intel Core i7 de 8° geração, 16GB de RAM, e Sistema Operacional Ubuntu 20.04 LTS.

4.2.1 Tempo de resposta do sistema

Algumas aplicações em *IoT* exigem baixa latência no processo de comunicação. A solução proposta neste trabalho incluí um processamento a mais no fluxo de envio de informações pelo dispositivo, e pode adicionar um atraso na entrega da informação à aplicação. Na implementação da solução utilizando a ferramenta GNU Radio, cada bloco executa em uma *thread*, logo a demodulação do quadro é feita em paralelo aos módulos de

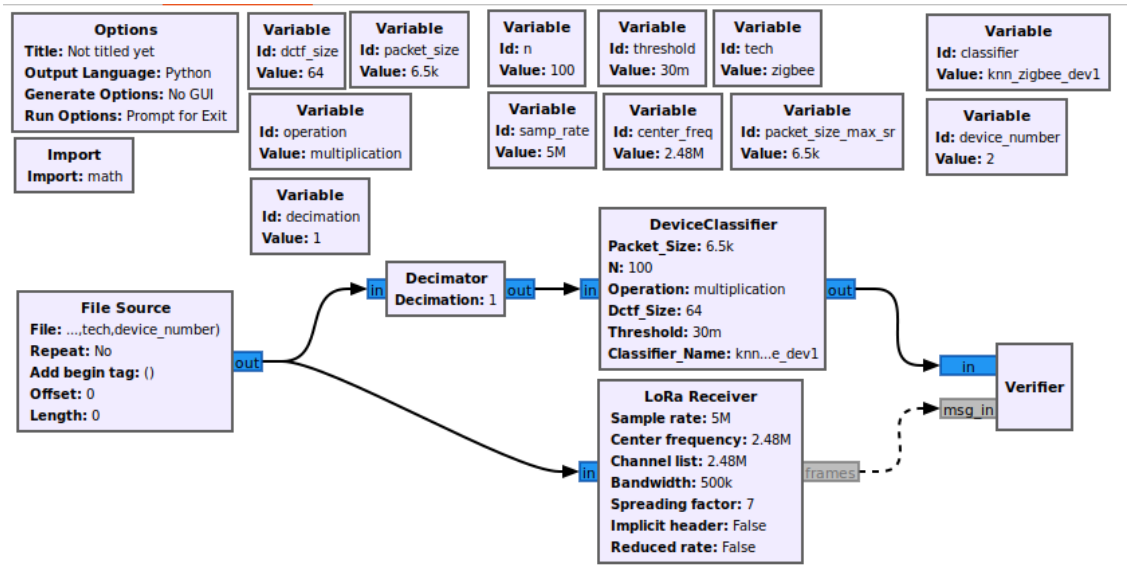


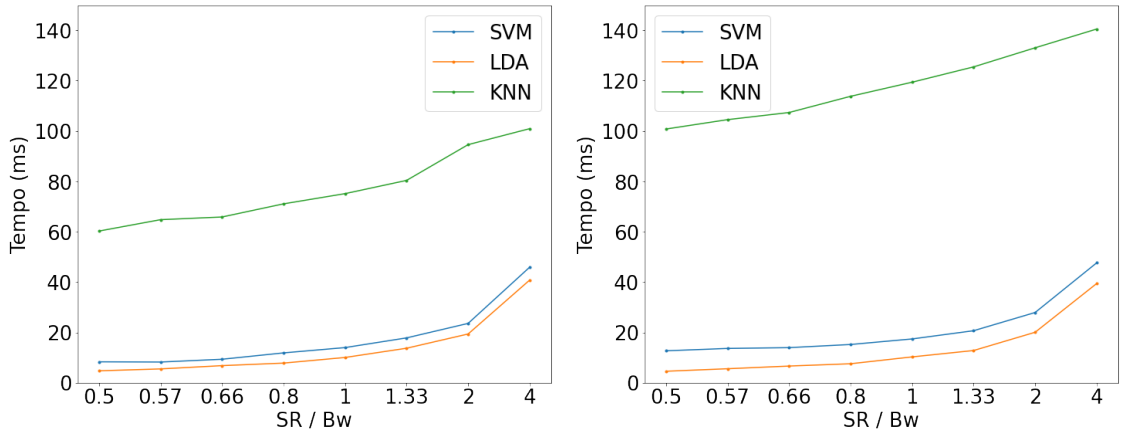
Figura 4.11: Implementação da solução *online* nos blocos do GNU Radio.

classificação, como pode ser visto na Figura 4.11. Portanto, se o tempo gasto no processo de classificação for maior que o tempo gasto na demodulação, o quadro ficará disponível para ser repassado à aplicação, mas ainda sem rótulo.

O tempo total de classificação consiste no tempo do processamento das características somado ao tempo de resposta do classificador. O primeiro experimento se propõe a verificar a relação entre a quantidade de dados e o tempo de classificação. Uma das formas de reduzir a quantidade de dados de entrada do classificador é diminuindo a taxa de amostragem. Para isso, foi criado um bloco denominado *Decimator* no GNU Radio, que tem como entrada as amostras do sinal e realiza o processo de decimação, tendo como saída os dados decimados a uma taxa definida como parâmetro.

Visando padronizar e fazer uma comparação entre as tecnologias LoRa e ZigBee, a referência para a redução da taxa de amostragem foi a largura de banda *Bw* da tecnologia. Por exemplo, os quadros coletados da tecnologia LoRa foram transmitidos em uma largura de banda de 500KHz, portanto uma relação de 2 vezes, que atende o valor mínimo recomendado pelo critério de Nyquist, significa uma taxa de amostragem de 1MHz. As Figuras 4.12(a) e 4.12(b) apresentam o resultado da redução da taxa de amostragem para o LoRa e para o ZigBee, consecutivamente. No geral, é possível perceber nos gráficos que reduzir a quantidade de dados implica em uma redução linear no tempo de classificação.

As Figuras 4.12(a) e 4.12(b) apresentam o tempo gasto na extração de características somado ao tempo de classificação. O tempo de classificação depende do tamanho da *DCTF*, logo, como o tamanho da *DCTF* não é alterado com a variação da taxa de amostragem, esse tempo é constante em todos pontos do gráfico. Já o tempo de extração de características muda conforme a taxa de amostragem, logo a curva vista nos gráficos

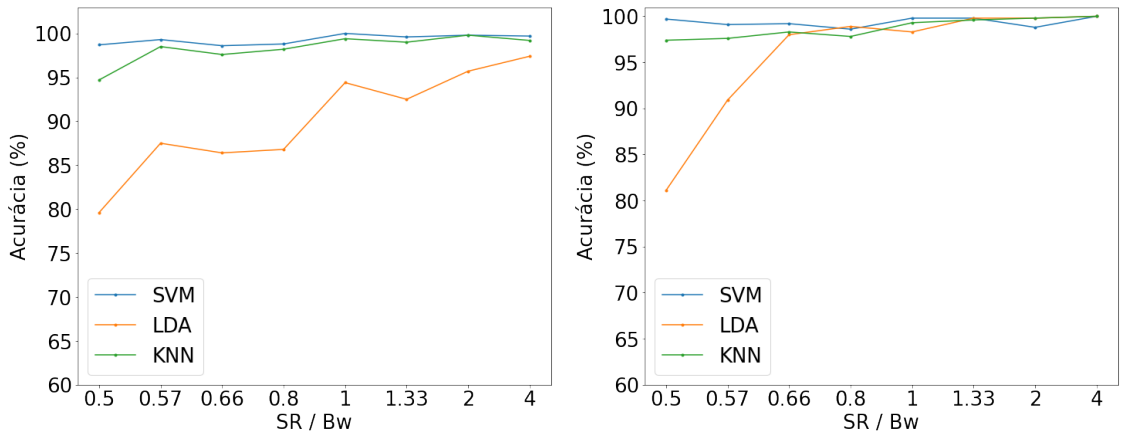


(a) *LoRa* (b) *ZigBee*

Figura 4.12: Tempo médio de extração de características e classificação baseado na quantidade de dados.

está relacionada a essa variação. Em taxas de amostragem pequenas, o tempo de extração de características é desprezível e pode-se ver uma diferença de tempo de até 10 vezes entre os classificadores. Conforme da taxa de amostragem aumenta, proporcionalmente essa diferença passa a ser menor e o tempo gasto pelo classificador fica menos relevante no tempo total.

A consequência de reduzir a quantidade de dados na classificação é a redução da acurácia. Isso acontece, pois com menos dados o classificador aprende menos. Nas Figuras 4.13(a) e 4.13(b) é possível observar um acréscimo na acurácia à medida que a taxa de amostragem aumenta. Os classificadores SVM e KNN permaneceram estáveis, mesmo com taxas de amostragens menores e tiveram uma acurácia miníma de 98,7% e 94,8% respectivamente, com a taxa de amostragem sendo a metade da largura de banda.



(a) *LoRa* (b) *ZigBee*

Figura 4.13: Acurácia baseada na quantidade de dados.

Se o tempo de resposta não for um requisito da aplicação, pode se escolher uma taxa de amostragem maior e selecionar qualquer um dos três classificadores. Em cenários onde o tempo de resposta é essencial, reduzir a taxa de amostragem pode ser uma opção e os algoritmos SVM e KNN têm o desempenho melhor.

4.2.2 Classificação online

A classificação *online*difere da *offline*, pois na etapa de testes são considerador dispositivos novos, estes que não são utilizados na etapa de treinamento. Para que isso seja possível, além do resultado da classificação, o *score* retornado pelo classificador foi utilizado para sinalizar dispositivos desconhecidos.

A Figura 4.14 apresenta os possíveis casos de classificação. No primeiro e no segundo caso, o dispositivo 1 é selecionado como intruso, tentando se passar por um dispositivo conhecido 2. Como o classificador não conhece o dispositivo 1 e só foi treinado com outros dispositivos, o classificador vai associá-lo a uma das quatro classes. No primeiro caso, se o dispositivo 1 for classificado como sendo o dispositivo 2, precisamos verificar o *score* para tomar a decisão de aceitar ou rejeitar o dispositivo. No segundo caso, o dispositivo intruso é associado com outros dispositivos da base e já é rejeitado. No terceiro caso, o dispositivo 2, real e conhecido na nossa base é classificado como sendo ele mesmo, entretanto precisamos analisar o *score* para tomarmos a decisão. E por último, no quarto caso, o dispositivo 2 é classificado como sendo outro dispositivo da base e já é rejeitado.

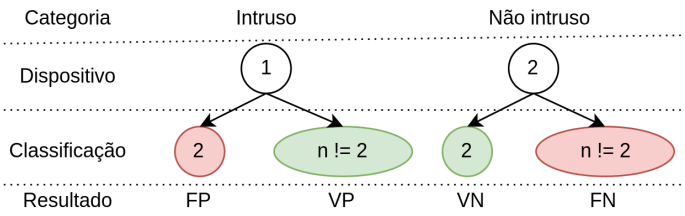


Figura 4.14: Dispositivo 1 tentando se passar por dispositivo 2.

É preciso analisar o *score* somente se a classe retornada pelo classificador for igual ao identificador do dispositivo. A Figura 4.15 apresenta os *scores* retornados pelo classificador para cinco dispositivos LoRa. Neste experimento, a base de treino é composta pelos dispositivos 1, 2, 3 e 5, sendo que o 4 é selecionado como intruso. É possível observar que o *score* de classificação para o dispositivo intruso é menor que os outros dispositivos. Na Figura 4.16 também é possível observar estes *scores* em um gráfico box plot. O primeiro quartil do dispositivo intruso não se encontra com o terceiro quartil dos dispositivos não intrusos, indicando que mais de 75% dos dados não se

sobrepõem. Portanto, é possível escolher um limiar que divide os dados em duas partes. Como descrito no Capítulo anterior, a escolha deste limiar fica a critério do operador do sistema e serão apresentados resultados que o ajudam nessa decisão.

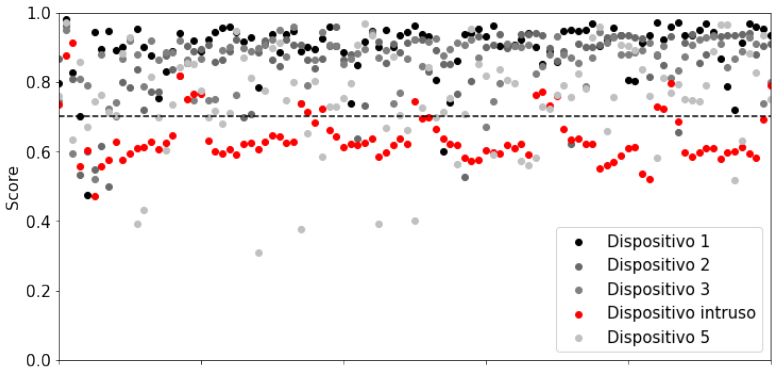


Figura 4.15: Score das amostras dos dispositivos Lora.

Em cada rodada do experimento, um dispositivo foi selecionado como intruso e o outros foram utilizados para treinar o modelo. Para cada limiar foi calculado duas taxas de erro de classificação. O erro do tipo 1 é calculado de acordo com a Equação 4-1 e indica o número de dispositivos não intrusos classificados como intrusos. Investigar falsos positivos custa tempo e recursos e dificulta a concentração em incidentes reais. O erro do tipo 2 é apresentado na Equação 4-2 e diz sobre classificar os dispositivos intrusos como não intrusos. Os falsos negativos são ameaças não detectadas e a consequência disso é o aumento do risco para aplicação, pois se reduz a capacidade de resposta a invasores.

$$Errotipo1 = \frac{FP}{FP + VN} \tag{4-1}$$

$$Errotipo2 = \frac{FN}{FN + VP} \tag{4-2}$$

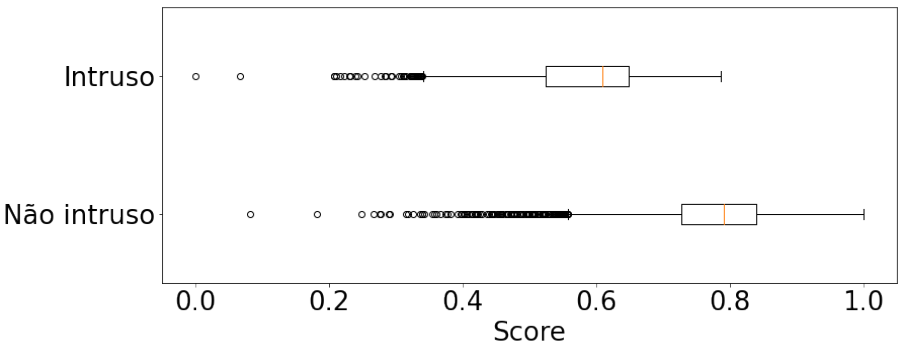


Figura 4.16: Boxplot score Lora.

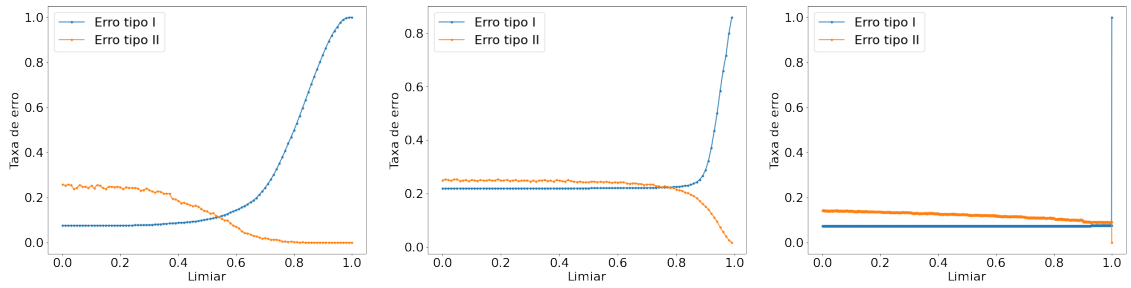
A Figura 4.17 apresenta os erros do tipo 1 e 2, conforme cada limiar, para a tecnologia LoRa. Neste experimento, em cada rodada um dispositivo da base de dados foi escolhido como intruso e não participou da fase de treinamento. A base de dados completa foi divida em treinamento e teste, seguindo a proporção de 80% para treino e 20% para teste. Em seguida, todos os dispositivos foram submetidos ao classificador sendo salvo o resultado da classificação, juntamente com o limiar. Por último, foi realizado um teste variando o limiar de 0 a 1 e medindo a taxa de erro em cada ponto. Conforme podemos ver na Figura 4.17, a medida que uma taxa de erro diminui, a outra aumenta. A ideia deste teste é mostrar que a escolha do limiar depende do cenário da aplicação e o operador pode decidir pelo que mais atende sua demanda.

Para simular os ataques, para cada entrada do dispositivo desconhecido foi selecionado aleatoriamente um dispositivo da base dados para ele imitar. Em seguida os dados dos dispositivos são submetidos ao classificador, sendo verificado o resultado da classificação. Se o resultado da classificação diferir do dispositivo escolhido, o dispositivo desconhecido é rotulado corretamente como intruso. Se o resultado da classificação for igual ao dispositivo escolhido é aleatoriamente feita uma verificação do *score*. Caso o *score* seja maior que o limiar, o dispositivo é rotulado erroneamente como não intruso, e se o *score* for menor, rotulado corretamente como intruso.

Neste experimento foi simulado um cenário em que 25% das entradas são de dispositivos intrusos, ou seja, dispositivos não conhecidos pelo classificador. Suponha que foi escolhido o limiar 0,55 da Figura 4.17(b). Conforme podemos observar, neste limiar o erro do tipo 1 é igual ao erro do tipo 2, ou seja, cerca de 12%. Como dito anteriormente, essa escolha fica a critério do administrador do sistema. Quando se deseja detectar o maior número possível de ataques, é preciso escolher um limiar maior, o que vai causar um aumento no número de alerta falsos.

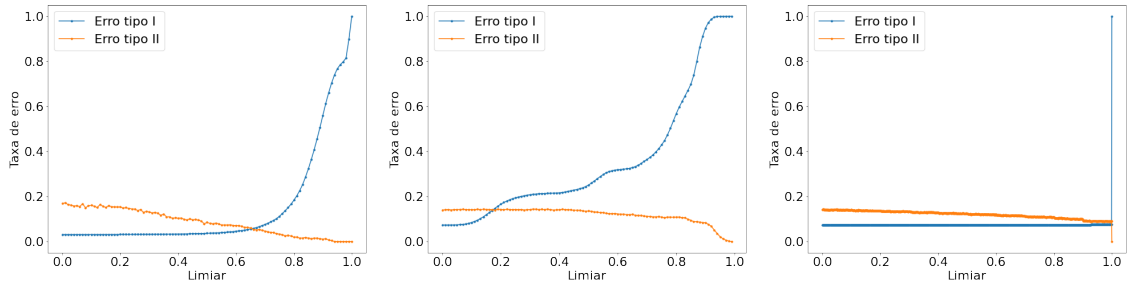
Olhando em conjunto os dois erros, os LDA não tem um bom desempenho em ambos os casos. No experimento com os dispositivos LoRa, os dois erros ficam acima de 20%, sendo que quando o erro do tipo 1 abaixa, o erro do tipo 2 cresce exponencialmente. No ZigBee, com um limiar baixo é possível ter uma taxa de erros do tipo 1 pequena, entretanto o erro do tipo 2 somente abaixa em limiares altos. Embora o LDA classifique bem dispositivos conhecidos, o uso da informação de score não ajuda muito a identificar dispositivos desconhecidos. O comportamento do KNN é um pouco diferente dos outros dois classificadores. A maioria dos dispositivos pertencentes a base de dados, ou seja, os dispositivos treinados, possuem *score* igual a 1.

No próximo experimento, o sistema foi analisado do ponto de vista do administrador. Além de detectar dispositivos intrusos com precisão, o número de alerta do sistema é uma métrica importante da perspectiva de um administrador. Como dito anteriormente, os alertas requerem tempo para serrem analisadas e aletas falsos são uma perda de tempo.



(a) *SVM* (b) *LDA* (c) *KNN*

Figura 4.17: Simulação selecionando um dispositivo LoRa como intruso.

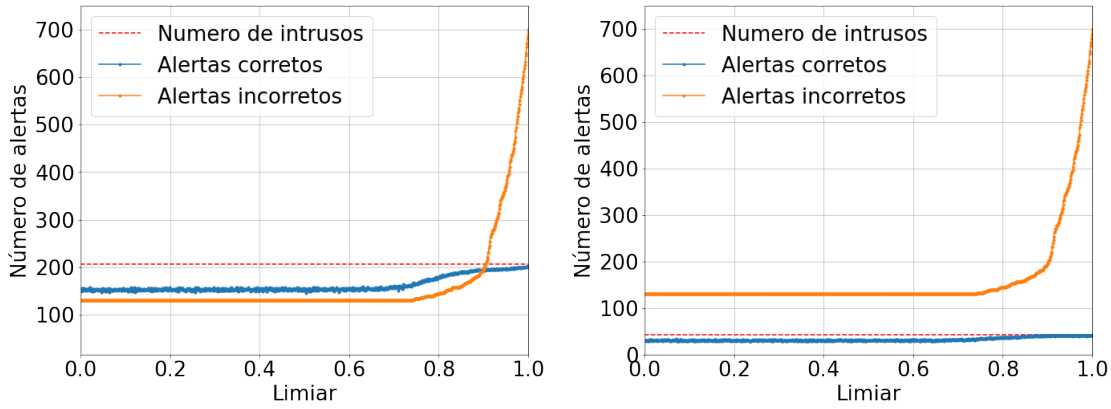


(a) *SVM LoRa* (b) *LDA* (c) *KNN*

Figura 4.18: Simulação selecionando um dispositivo ZigBee como intruso.

As Figuras 4.19(a) e 4.19(b) apresentam os resultados da escolha do limiar na geração de alertas em dois cenários diferentes. No primeiro cenário, os quadros do dispositivo intruso representam 5% do número de quadros dos dispositivos intrusos. No segundo cenário, os quadros do dispositivo intruso representam 1% do total de quadros. O classificador foi treinado com 4 dispositivos LoRa e um dispositivo foi escolhido como intruso, não participando da fase de treinamento. Após o treinamento, o sistema recebeu quadros dos 5 dispositivos. As linhas pontilhadas vermelhas nas figuras representam o número de quadros do dispositivo intruso, que no primeiro cenário é 206 e no segundo 43.

Para reduzir o número de combinações de parâmetros possíveis, foi escolhido o classificador SVM e o dispositivo LoRa. Os eventos anormais são raros de acontecer e na maioria do tempo o sistema não está sendo atacado [Elshoush e Osman 2010]. É possível observar que quanto maior o número de ataques, menor é o número de alertas falsos em relação aos alertas verdadeiros. Isso significa que se deseja ser alertado quanto a ataques quase 100% das vezes, é preciso assumir cerca de 5% de alertas falsos.



(a) *Cenário 1: 5% dos quadros são do dispositivo intruso.* (b) *Cenário 2: 1% dos quadros são do dispositivo intruso.*

Figura 4.19: Número de alertas gerados com 4000 quadros transmitidos.

4.3 Conclusão

Neste Capítulo, foi apresentada a avaliação do sistema de detecção de dispositivos intrusos. A avaliação *offline* mostrou que a técnica DCTF permite classificar dispositivos de diferentes tecnologias. Usando uma mesma implementação e variando alguns parâmetros, é possível estender a técnica para outras tecnologias de comunicação. Para isso, foram apresentados testes de classificação com dispositivos LoRa, ZigBee e WiFi. Os testes demonstraram que com o uso de classificadores simples, é possível classificar dispositivos com 100% de acurácia, entretanto com variações no ambiente como temperatura e posição dos dispositivos diminui essa acurácia. Também foi mostrado que a temperatura do dispositivo e a distância entre o transmissor e o receptor influenciam na DCTF.

Em aplicações em que os fatores tempo e custo computacional não são restritos, os algoritmos de redes neurais podem ser utilizados, por historicamente apresentarem taxas de aceitos superiores. Os classificadores mais simples podem ser utilizados em aplicações onde há essas restrições, com o custo de uma menor taxa de acertos.

Na avaliação online, foram apresentados testes que mensuraram o impacto do uso da técnica no QoS das aplicações IoT. A diminuição da quantidade de dados através da redução da taxa de amostragem possibilita classificar os dispositivos em menos de 20ms. Foram realizados experimentos utilizando o *score* para classificação de dispositivos desconhecidos. Embora os experimentos tenham apresentado uma boa taxa de acertos pelos classificadores, quando o *score* é utilizado a quantidade de alertas falsos ainda é grande.

CAPÍTULO 5

Considerações finais e trabalhos futuros

Este trabalho propôs um sistema capaz de identificar dispositivos intrusos de modo *online*. O sistema foi baseado na técnica de DCTF, que possibilita classificar dispositivos de diferentes tecnologias. A arquitetura modular proposta, além de permitir a experimentação de novas técnicas, permite integração com novas tecnologias e aplicações.

Através dos experimentos, foi possível notar que não existe um algoritmo de aprendizado de máquina que tenha um desempenho melhor em todos casos. Dependendo da tecnologia escolhida, um algoritmo apresentou um resultado melhor que em outra tecnologia. A escolha do algoritmo também pode ser baseada no tempo de resposta do sistema, que em alguns casos o algoritmo com desempenho melhor levou mais tempo para dar resposta.

Classificar dispositivos conhecidos já é um desafio superado. Este e outros trabalhos da literatura demonstram ser possível classificar dispositivos usando o sinal eletromagnético com uma alta acurácia. Entretanto, o cenário muda quando dispositivos desconhecidos entram em cena. Os experimentos deste trabalho mostraram que embora se identifique bem dispositivos intrusos, o número de alertas falsos ainda é grande, o que pode gerar trabalho extra aos administradores do sistema ao tentar identificar um alerta verdadeiro. O trabalho apresentou diversos parâmetros e técnicas que podem ser utilizados em conjunto e têm desempenho melhor em um cenário específico. Logo o conhecimento do administrador é essencial para análise dos cenários e escolha dos parâmetros.

Durante o desenvolvimento desta pesquisa, foram identificados alguns tópicos que não foram abordados neste trabalho e que possuem potencial para serem investigados. Portanto, como trabalho futuro, pretende-se fazer uma avaliação com uma quantidade maior de dispositivos. Também pretende-se estudar a fundo as variações do ambiente, como temperatura e ruído, e sua influência na geração da DCTF, com o intuito de propor uma técnica que se adapte a essas condições. Também tem-se o interesse de integrar o sistema com outras aplicações de detecção de intrusão e.g., Snort e Suricata [Albin 2011].

O código fonte dos blocos do GNU Radio e o código utilizado nos experimentos *offline* estão disponíveis na seguinte URL: https://github.com/marcosfelipp/iot-device-detection.

Referências Bibliográficas

[Albin 2011]ALBIN, E. A comparative analysis of the snort and suricata intrusion-detection systems. In: . [S.l.: s.n.], 2011.

[Barbeau, Hall e Kranakis 2006]BARBEAU, M.; HALL, J.; KRANAKIS, E. Detection of rogue devices in bluetooth networks using radio frequency fingerprinting. In: *proceedings of the 3rd IASTED International Conference on Communications and Computer Networks, CCN*. [S.l.: s.n.], 2006.

[Bor, Vidler e Roedig 2016]BOR, M. C.; VIDLER, J.; ROEDIG, U. Lora for the internet of things. In: *EWSN*. [S.l.: s.n.], 2016.

[Brik et al. 2008]BRIK, V. et al. Wireless device identification with radiometric signatures. In: *Proceedings of the 14th ACM international conference on Mobile computing and networking*. [S.l.: s.n.], 2008.

[Candès e Wakin 2008]CANDÈS, E. J.; WAKIN, M. B. An introduction to compressive sampling. *IEEE signal processing magazine*, 2008.

[Choe et al. 1995]CHOE, H. C. et al. Novel identification of intercepted signals from unknown radio transmitters. In: *Wavelet Applications II*. [S.l.: s.n.], 1995.

[Dillinger, Madani e Alonistioti 2005]DILLINGER, M.; MADANI, K.; ALONISTIOTI, N. *Software defined radio: Architectures, systems and functions*. [S.l.]: John Wiley & Sons, 2005.

[Elshoush e Osman 2010]ELSHOUSH, H. T.; OSMAN, I. M. Reducing false positives through fuzzy alert correlation in collaborative intelligent intrusion detection systems—a review. In: *International Conference on Fuzzy Systems*. [S.l.: s.n.], 2010.

[Ergen 2004]ERGEN, S. C. Zigbee/ieee 802.15. 4 summary. *UC Berkeley, September*, 2004.

[Farahvash, Quek e Mak 2008]FARAHVASH, S.; QUEK, C.; MAK, M. A temperature-compensated digitally-controlled crystal pierce oscillator for wireless applications. In: *2008 IEEE International Solid-State Circuits Conference-Digest of Technical Papers*. [S.l.: s.n.], 2008.

[Gomez e Paradells 2010]GOMEZ, C.; PARADELLS, J. Wireless home automation networks: A survey of architectures and technologies. In: . [S.l.: s.n.], 2010.

[Haxhibeqiri et al. 2018]HAXHIBEQIRI, J. et al. A survey of lorawan for iot: From technology to application. *Sensors*, 2018.

[Haykin 2008]HAYKIN, S. *Communication systems*. [S.l.]: John Wiley & Sons, 2008.

[Hearst et al. 1998]HEARST, M. A. et al. Support vector machines. *IEEE Intelligent Systems and their applications*, 1998.

[Hilburn et al. 2018]HILBURN, B. et al. Sigmf: the signal metadata format. In: *Proceedings of the GNU Radio Conference*. [S.l.: s.n.], 2018.

[Jiang et al. 2019]JIANG, Y. et al. Physical layer identification of lora devices using constellation trace figure. *EURASIP Journal on Wireless Communications and Networking*, 2019.

[Keller, Gray e Givens 1985]KELLER, J. M.; GRAY, M. R.; GIVENS, J. A. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, 1985.

[Köse et al. 2019]KÖSE, M. et al. Rf fingerprinting of iot devices based on transient energy spectrum. *IEEE Access*, 2019.

[Mohammadian e Tellambura 2021]MOHAMMADIAN, A.; TELLAMBURA, C. Rf impairments in wireless transceivers: Phase noise, cfo, and iq imbalance–a survey. *IEEE Access*, 2021.

[Peng et al. 2016]PENG, L. et al. A differential constellation trace figure based device identification method for zigbee nodes. In: *2016 8th International Conference on Wireless Communications & Signal Processing (WCSP)*. [S.l.: s.n.], 2016.

[Reus-Muns et al. 2020]REUS-MUNS, G. et al. Trust in 5g open rans through machine learning: Rf fingerprinting on the powder pawr platform. In: *2020 IEEE Global Communications Conference (GLOBECOM), IEEE*. [S.l.: s.n.], 2020.

[Riyaz et al. 2018]RIYAZ, S. et al. Deep learning convolutional neural networks for radio identification. *IEEE Communications Magazine*, 2018.

[Sankhe et al. 2019]SANKHE, K. et al. No radio left behind: Radio fingerprinting through deep learning of physical-layer hardware impairments. *IEEE Transactions on Cognitive Communications and Networking*, 2019.

[Shannon 1949]SHANNON, C. E. Communication in the presence of noise. *Proceedings of the IRE*, 1949.

[Soltani et al. 2020]SOLTANI, N. et al. More is better: Data augmentation for channel-resilient rf fingerprinting. *IEEE Communications Magazine*, 2020.

[Thakur e Khare 2013]THAKUR, R.; KHARE, K. Synchronization and preamble concept for frame detection in ofdm. *International Journal of Modeling and Optimization*, 2013.

[Tian et al. 2019]TIAN, Q. et al. New security mechanisms of high-reliability iot communication based on radio frequency fingerprint. *IEEE Internet of Things Journal*, 2019.

[Tomasin, Zulian e Vangelista 2017]TOMASIN, S.; ZULIAN, S.; VANGELISTA, L. Security analysis of lorawan join procedure for internet of things networks. In: *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. [S.l.: s.n.], 2017.

[Toonstra e Kinsner 1995]TOONSTRA, J.; KINSNER, W. Transient analysis and genetic algorithms for classification. In: *WESCANEX 95. Communications, Power, and Computing. Conference Proceedings., IEEE*. [S.l.: s.n.], 1995.

[Traore 2011]TRAORE, I. *Continuous Authentication Using Biometrics: Data, Models, and Metrics: Data, Models, and Metrics*. [S.l.]: Igi Global, 2011.

[Xuping e Jianguo 2007]XUPING, Z.; JIANGUO, P. Energy-detection based spectrum sensing for cognitive radio. IET, 2007.