Alberto Egon Schaeffer-Filho
Luigi Carro
Weverton Cordeiro

Escola de Computação PPGC/UFRGS 50 Anos:

Transformando Desafios em Oportunidades para o Futuro

Outubro 2023

ALBERTO EGON SCHAEFFER-FILHO
LUIGI CARRO
WEVERTON LUIS DA COSTA CORDEIRO

ESCOLA DE COMPUTAÇÃO PPGC/UFRGS 50 ANOS:

TRANSFORMANDO DESAFIOS EM OPORTUNIDADES PARA O FUTURO

Porto Alegre
Sociedade Brasileira de Computação – SBC
2023

Dados Internacionais de Catalogação na Publicação (CIP)

E74 Escola de Computação PPGC/UFRGS 50 anos: transformando desafios em oportunidades para o futuro (Alberto Egon Schaeffer-Filho, Luigi Carro, Weverton Luis da Costa Cordeiro). Dados eletrônicos. – Porto Alegre: Sociedade Brasileira de Computação, 2023.

212 p. : il. : PDF ; 52 MB

Modo de acesso: World Wide Web.
Inclui bibliografia
ISBN 978-85-7669-558-5 (e-book)

1. Computação. 2. Universidade Federal do Rio Grande do Sul. 4. Ensino de computação. I. Schaeffer-Filho, Alberto Egon. II. Carro, Luigi. III. Cordeiro, Weverton Luis da Costa. IV. Sociedade Brasileira de Computação. VI. Título.

CDU 004(037)

Ficha catalográfica elaborada por Annie Casali – CRB-10/2339

Biblioteca Digital da SBC – SBC OpenLib

APRESENTAÇÃO

A evolução da computação nos últimos 50 anos passou pelo foco no algoritmo e sua execução eficiente, com dados locais, para foco na informação global, com máquinas capazes de entender o problema e resolvê-lo autonomamente. A TV levou 26 anos para chegar a 100 milhões de pessoas, enquanto que o computador pessoal, de 1975, levou 16 anos. O acesso à web por 100 milhões de pessoas levou 7 anos depois de seu lançamento em 1991, e o smartphone (de 2007) levou apenas 3 anos para atingir a mesma marca. O Whatsapp levou apenas 3,5 anos, e este tempo, antes medido em anos, agora se mede em meses, pois o ChatGPT levou apenas 2 meses para atingir 100 milhões de usuários.

Este mundo em que a velocidade adaptação a novas tecnologias é tão alta reflete não somente a evolução da Computação nos últimos 50 anos, mas também o surgimento de muitas áreas específicas da Computação, para que se pudesse estudar mais profundamente cada novo problema. Aliando-se especificidade com necessidade de velocidade, fica evidente que a capacidade de aprender e saber utilizar novos conceitos é um diferencial importante na formação de profissionais capacitados.

Neste contexto, o Programa de Pós-Graduação em Computação (PPGC) da Universidade Federal do Rio Grande do Sul (UFRGS) realizou nos dias 05 e 06 de outubro a “Escola de Computação – PPGC/UFRGS 50 Anos”, que se somou a uma série de ações, que ocorreram ao longo de 2023, para comemorar os 50 anos do PPGC. O evento, que aconteceu presencialmente na uMov.me Arena, na cidade de Porto Alegre, foi uma oportunidade única de discutir temas relevantes no campo da computação e refletir sobre as transformações das últimas cinco décadas.

O evento teve como objetivo abordar a evolução da computação ao longo dos últimos 50 anos, destacando como a área passou de um foco em algoritmos e eficiência de execução para uma ênfase na informação global e máquinas capazes de resolver problemas de forma autônoma. A escola contou com oito apresentações que abordaram tópicos de vanguarda em diversas áreas de pesquisa no PPGC. Cada apresentação foi acompanhada por um capítulo que faz parte do presente livro, contribuindo para a disseminação do conhecimento gerado no evento:

- Capítulo 1: **“Desenvolvendo o Código da Internet do Futuro”,** *Alberto Schaeffer Filho, Jéferson C. Nobre, Juliano Wickboldt, Lisandro Granville, Luciano Gaspary, Weverton Cordeiro*
- Capítulo 2: **“Processamento de Imagens Omnidirecionais e Aplicações”**, *Cláudio R. Jung, Thiago L. T. da Silveira*
- Capítulo 3: **“Pensamento Computacional Paralelo: Desafios do Presente e do Futuro”**, *Arthur Francisco Lorenzon, Lucas Mello Schnorr*
- Capítulo 4: **“Desafios para a Computação Energeticamente Eficiente”**, *Luigi Carro, Gabriel Luca Nazar*
- Capítulo 5: **“Realidade Virtual: Potencialidades de uma Nova Plataforma Interativa”**, *Luciana Nedel, Carla Maria Dal Sasso Freitas*
- Capítulo 6: **“Computação Visual e Detecção Precoce de Doenças em Escala Global: Oportunidades e Desafios”**, *Manuel Menezes de Oliveira Neto (PPGC), Giovani André Meneguel (PPGC), Pantelis Varvaki Rados (PPG Odontologia)*
- Capítulo 7: **“Segurança Cibernética 2030: Experiências, Desafios e Oportunidades”**, Alberto Schaeffer Filho, Jéferson C. Nobre, Juliano Wickboldt, Lisandro Granville, Luciano Gaspary, Weverton Cordeiro
- Capítulo 8: **“A Nova Eletricidade: Aplicações, Choques e Tendências da IA Moderna”**, Ana L. C. Bazzan, Anderson R. Tavares, André G. Pereira, Cláudio R. Jung, Jacob Scharcanski, Joel Carbonera, Luis C. Lamb, Mariana Recamonde-Mendoza, Thiago L. T. da Silveira, Viviane Moreira

A Escola foi agraciada com o patrocínio da Nelogica e SDC, além do apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), da Sociedade Brasileira de Computação (SBC) e da uMov.me. A “Escola de Computação – PPGC/UFRGS 50 Anos” foi destinada a alunos de graduação, pós-graduação e à sociedade em geral, proporcionando um ambiente propício para a discussão e reflexão sobre os avanços e desafios da computação.

Capítulo 1

Desenvolvendo o Código da Internet do Futuro

Alberto Egon Schaeffer-Filho, Jéferson Campos Nobre, Juliano Wickboldt, Lisandro Zambenedetti Granville, Luciano Paschoal Gaspary, Weverton Luis da Costa Cordeiro

Programa de Pós-Graduação em Computação (PPGC) - UFRGS

Abstract

Research in computer networks in the 1970s and 1980s led to the emergence of the Internet, a significant technological achievement. It evolved from an academic network for simple text files exchanges into a multimedia platform that transformed global interaction and disrupted various industries. The Internet's success is attributed to robust technologies, but its inflexibility led to its "ossification." Network programmability, allowing software development on network devices, is revitalizing the Internet. This chapter explores this evolution, focusing on virtualization, network programming languages, monitoring, current challenges, and opportunities.

Resumo

As pesquisas em redes de computadores nas décadas de 1970 e 1980 culminaram com o surgimento da Internet, uma conquista tecnológica significativa. Ela evoluiu de uma rede acadêmica simples para uma plataforma multimídia que transformou a interação global e perturbou várias indústrias. O sucesso da Internet se deve a tecnologias robustas, mas sua rigidez levou à "ossificação". A programabilidade de redes, permitindo o desenvolvimento de software nos dispositivos de rede, está revitalizando a Internet. Este capítulo explora essa evolução, com foco em virtualização, linguagens de programação para redes, monitoramento, desafios atuais e oportunidades.

1.1. Introdução

As pesquisas em redes de computadores nas décadas de 1970 e 1980 culminaram com o surgimento da Internet, uma das conquistas tecnológicas recentes mais significativas da

Vídeo com a apresentação do capítulo: `https://youtu.be/ma8UYCJC_kc`

humanidade. A Internet passou de uma rede acadêmica voltada para a troca de arquivos de texto simples para uma plataforma multimídia que revolucionou a forma como as pessoas de todo o mundo interagem. Isso levou a disrupções em diversas áreas, abalando indústrias como a de telefonia, fonográfica, audiovisual, jornalística, entre outras. A Internet provocou a queda vertiginosa de várias indústrias tradicionais e criou as condições para o surgimento de impérios tecnológicos inovadores, como Google, Facebook e Amazon.

O sucesso da Internet se deve, entre outros fatores, a um conjunto de tecnologias de redes de computadores absolutamente robustas e estáveis, representadas principalmente pela suíte de protocolos TCP/IP [Comer 2017]. No entanto, ao longo dos anos, a Internet, que foi tão inovadora e possibilitou tantas criações, tornou-se um ambiente cujo núcleo se mostrou praticamente imutável, dificultando a absorção de novas soluções. Essa incapacidade de absorver inovações em seu núcleo ficou conhecida como a ossificação da Internet [Clark 2003], o que motivou a comunidade científica a buscar formas de tornar a Internet mais flexível, assim como era nos primórdios.

Um dos movimentos de maior sucesso, que tem transformado a Internet em um ambiente novamente propício às inovações, é aquele que ficou conhecido como programabilidade de redes [Liu et al. 2021]. Com a programabilidade de redes, adquire-se a capacidade de desenvolver e implantar software diretamente nos dispositivos de rede, como switches e roteadores. Esse software de rede pode ser desenvolvido por qualquer pessoa com conhecimento técnico, não se limitando mais ao software ou firmware fornecido pelos fabricantes de dispositivos. Dessa forma, desenvolvedores criativos podem codificar soluções com software executando não apenas na periferia da Internet (*e.g.*, em servidores, estações de trabalho, dispositivos móveis, IoT), mas também e principalmente em seu núcleo (*e.g.*, switches e roteadores). Assim, a programabilidade de redes "reabre"a Internet para acomodar inovações.

Este capítulo aborda a evolução das tecnologias da Internet e, principalmente, foca nos avanços científicos recentes que têm permitido uma Internet novamente disruptiva. Serão discutidos assuntos como virtualização em redes de computadores (Seção 1.2), linguagens de programação para redes (Seção 1.3), monitoramento e desempenho (Seção 1.4), in-network computing (Seção 1.5), redes móveis programáveis (Seção 1.6) e oportunidades para novos desenvolvimentos via redes programáveis (Seção 1.7).

1.2. Virtualização em Redes de Computadores

Comparando a indústria de redes de computadores com a indústria de desktops e servidores no que diz respeito às tecnologias de virtualização, a primeira apresenta um histórico de ciclos defasados em relação à segunda. A virtualização de desktops e servidores permite que sistemas operacionais distintos sejam executados no mesmo hardware, desacoplando assim o software do hardware. Isso possibilita que os consumidores adquiram hardware de um fornecedor e software de outro. Esse fato se tornou tão comum que agora parece óbvio. No entanto, até recentemente, o mesmo não era aplicável à indústria de redes de computadores. Os operadores de redes costumavam adquirir equipamentos e funções (em software ou firmware) sempre do mesmo fornecedor. Não havia a possibilidade de instalar software de um fornecedor em hardware de outro fornecedor. Isso resultava em custos mais elevados, limitava a liberdade de escolha e dificultava a inovação

nas redes.

A virtualização em redes de computadores é uma abordagem que ataca o problema acima ao permitir a criação de redes virtuais independentes dentro de uma infraestrutura de rede física compartilhada. Essa tecnologia é baseada em software de virtualização que divide e compartilha recursos de rede, como largura de banda, endereços IP e switchings, entre várias instâncias virtuais. Isso se traduz em benefícios significativos para a gerência e o desempenho da rede.

Uma das principais tecnologias empregadas na criação de redes com suporte a virtualização é o conceito de SDN (*Software-Defined Networking*) [Feamster et al. 2014]. O SDN separa o plano de controle do plano de dados em uma rede, desacoplando a inteligência de gerenciamento de rede (controle) dos dispositivos de rede físicos (dados) (Figura 1.1). O elemento central do SDN é o controlador, um software responsável por tomar decisões sobre como o tráfego na rede deve ser encaminhado com base em políticas definidas por software. Os dispositivos de rede, como switches e roteadores, são simplificados e se tornam dispositivos rudimentares que simplesmente encaminham o tráfego conforme as instruções do controlador.

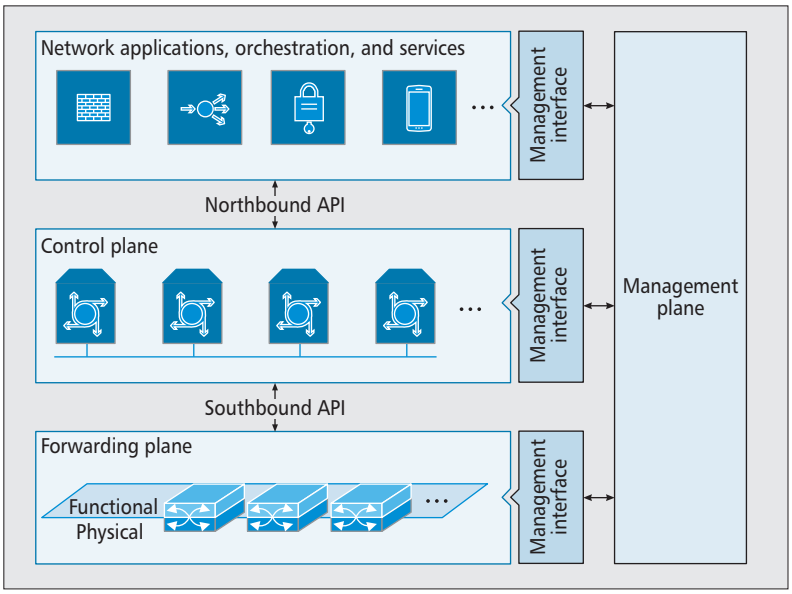


Figura 1.1. Arquitetura SDN

No contexto da virtualização de redes, o SDN desempenha um papel fundamental. Ele permite a criação de várias redes virtuais independentes em cima da infraestrutura física compartilhada. Isso é especialmente valioso em ambientes como data centers e nuvem, onde a flexibilidade e a escalabilidade são essenciais. Os benefícios do SDN incluem maior agilidade na implantação de serviços, redução de custos operacionais, melhor utilização de recursos de rede, automação avançada e capacidade de adaptação dinâmica às mudanças no tráfego.

Em termos de aplicações, o SDN é amplamente utilizado em data centers, provedores de serviços de nuvem, redes empresariais e até mesmo em ambientes de campus universitários. Em resumo, o SDN transforma a maneira como as redes são configuradas

e gerenciadas, tornando-as mais flexíveis, eficientes e adaptáveis, o que é fundamental em ambientes onde a virtualização de redes desempenha um papel crucial [Wickboldt et al. 2015].

Além de SDN, NFV (Virtualização de Funções de Rede) [Guo and McKeown 2018] é uma abordagem fundamental no contexto da virtualização de redes. NFV visa transformar funções de rede tradicionalmente executadas em hardware dedicado, como roteadores e firewalls, em software que pode ser executado em servidores de propósito geral e dispositivos de rede comuns. Isso é alcançado por meio de Funções de Rede Virtualizadas (VNFs), que são pacotes de software que representam essas funções de rede.

O coração do NFV é a virtualização das funções de rede. Essas VNFs podem ser implantadas, escalonadas e gerenciadas de forma flexível em ambientes virtualizados, proporcionando flexibilidade às operadoras de rede para adaptar suas infraestruturas às demandas em constante mudança.

O NFV também envolve um sistema de orquestração, que é responsável por gerenciar a implantação e o escalonamento das VNFs em servidores virtuais. Além disso, a infraestrutura virtualizada fornece os recursos de computação, armazenamento e rede necessários para hospedar as VNFs.

Os benefícios do NFV incluem flexibilidade, redução de custos, escalabilidade, maior inovação e gerenciamento simplificado. Ele é utilizado para aprimorar a agilidade das redes, permitir a rápida adaptação às mudanças nas demandas de tráfego e facilitar a introdução de novos serviços e funções de rede. Em resumo, o NFV desempenha um papel crucial na transformação das redes tradicionais em infraestruturas de rede virtualizadas, tornando-as mais eficientes, flexíveis e adaptáveis às necessidades dinâmicas das redes modernas.

1.2.1. Ossificação da Internet

A "ossificação da Internet"é um termo que descreve um fenômeno no qual a evolução e a inovação na arquitetura e nos protocolos da Internet se tornam mais lentas ou até mesmo estagnam devido à resistência à mudança e à dependência de sistemas existentes. Esse fenômeno ocorre devido a várias razões.

Uma das principais razões para a ossificação da Internet é a ampla implantação de tecnologias e padrões estabelecidos. Quando as redes e os sistemas já estão funcionando de acordo com um conjunto específico de protocolos, é difícil convencer os operadores a adotar novos padrões, mesmo que sejam mais eficientes ou seguros. A interoperabilidade e a compatibilidade retroativa com equipamentos mais antigos podem se tornar desafios significativos.

Outro fator que contribui para a ossificação é a influência de interesses comerciais e econômicos. Grandes empresas e organizações que investiram pesadamente em tecnologias existentes podem resistir a mudanças que possam afetar seus modelos de negócios ou investimentos. Isso pode resultar na manutenção de sistemas obsoletos por mais tempo do que o ideal.

Além disso, a complexidade da Internet atual também desempenha um papel na

ossificação. À medida que a Internet cresceu e evoluiu, tornou-se uma rede global altamente interconectada, e qualquer mudança na infraestrutura ou protocolos deve ser cuidadosamente coordenada em todo o mundo. Isso pode levar a processos de tomada de decisão lentos e conservadores.

A ossificação da Internet é um desafio significativo, pois limita a capacidade da Internet de se adaptar às novas demandas e ameaças emergentes, como segurança cibernética e escalabilidade. Para combater esse fenômeno, é importante incentivar a pesquisa contínua, a padronização flexível e a adoção de novas tecnologias, enquanto também considera cuidadosamente as implicações para a interoperabilidade e a estabilidade da rede.

1.2.2. Mitigando a Ossificação da Internet com Virtualização

A virtualização de redes desempenha um papel importante no comabate contra a ossificação da Internet. A ossificação ocorre quando a evolução e a inovação na arquitetura e nos protocolos da Internet diminuem devido à resistência à mudança e à dependência de sistemas e tecnologias existentes. A virtualização de redes oferece uma solução para este problema, introduzindo flexibilidade e adaptabilidade na infraestrutura de rede.

Uma das maneiras pelas quais a virtualização de redes combate a ossificação é permitindo a introdução rápida de novos serviços e funções de rede. Isso é feito por meio da criação de VNFs (como parte do arcabouço NFV), que podem ser implantadas como software sem a necessidade de modificar o hardware físico ou os protocolos existentes. Isso promove a inovação contínua sem interromper os serviços existentes.

A virtualização de redes também oferece escalabilidade sob demanda, permitindo que os recursos de rede sejam dimensionados de acordo com as necessidades do tráfego, eliminando a necessidade de adquirir hardware dedicado. Além disso, sistemas de gerenciamento centralizados e orquestração simplificam a administração e a introdução de mudanças na infraestrutura de rede.

Ao aderir a padrões abertos e desacoplar o software do hardware, a virtualização de redes promove a interoperabilidade e a colaboração entre fornecedores e desenvolvedores, facilitando a introdução de soluções inovadoras. Além disso, ela permite uma migração gradual de serviços legados para ambientes virtualizados, garantindo a compatibilidade com sistemas existentes.

Em última análise, a virtualização de redes oferece a flexibilidade necessária para que a Internet continue evoluindo e atendendo às crescentes demandas dos usuários e das aplicações, ao mesmo tempo em que mantém a estabilidade e a interoperabilidade com sistemas legados, combatendo assim a ossificação da Internet.

A virtualização de redes emerge como uma solução para combater a ossificação da Internet. Ela traz flexibilidade, agilidade e inovação para a infraestrutura de rede, permitindo a evolução contínua sem a necessidade de alterar hardware ou protocolos existentes. Isso é essencial em um cenário em que a resistência à mudança e a dependência de tecnologias estabelecidas podem impedir a adaptação da Internet às crescentes demandas e desafios. A virtualização de redes também promove a introdução rápida de novos serviços e funções de rede, escalabilidade eficiente e gerenciamento simplificado, tornando-a uma abordagem valiosa para operadoras de telecomunicações, provedores de serviços de rede

e empresas que buscam inovação e eficiência.

Relacionado a esse contexto de evolução das redes, surgem linguagens de programação específicas, como o P4 (*Programming Protocol-independent Packet Processors*), que desempenham um papel fundamental. O P4 permite a programação de dispositivos de rede para personalizar o processamento de pacotes, abrindo caminho para a criação de redes mais inteligentes e adaptáveis. Essa linguagem fornece as ferramentas necessárias para definir como os pacotes de dados são manipulados na rede, oferecendo um controle granular sobre o comportamento da rede. Em síntese, a combinação da virtualização de redes com linguagens de programação como o P4 representa uma abordagem importante para modernizar e revitalizar a Internet.

1.3. Linguagens para Programação para Redes

Redes programáveis correspondem a um paradigma que permite que a funcionalidade da rede seja definida e modificada dinamicamente por meio do software [Feamster et al. 2014]. As redes programáveis podem incluir vários componentes, como switches e roteadores programáveis, FPGAs, SmartNICs e servidores virtualizados. Esses componentes podem ser programados para executar tarefas específicas ou implementar protocolos personalizados, tornando a rede mais flexível e adaptável às mudanças de requisitos.

1.3.1. Perspectiva Histórica

Embora as redes programáveis estejam se tornando mais comuns, a ideia remonta à década de 90, quando as *active networks* foram introduzidas. O objetivo de *active networks* era tornar os roteadores mais flexíveis, permitindo que os desenvolvedores executassem programas personalizados neles. As *active networks* foram motivadas pela necessidade de alguns pesquisadores implementarem seus próprios protocolos e funcionalidades em suas redes, sem que fosse necessário esperar pelos processos de padronização da IETF para que pudessem implantar uma nova ideia [Wetherall and Tennenhouse 2019].

Conceitualmente, os pesquisadores idealizaram duas formas de processamento de código em roteadores:

- ***Cápsulas:*** a primeira é a substituição dos pacotes por *cápsulas* que carregam código que a infraestrutura poderia hospedar e processar;
- ***Switch/roteador programável***: a outra abordagem foi o *switch programável*, permitindo instalar programas no switch que seriam responsáveis por controlar a infraestrutura [Tennenhouse and Wetherall 1996].

Porém, a ideia de *active networks* não teve a "transferência de resultados" esperada para a indústria, e acabou não sendo amplamente adotada por setores fora do meio acadêmico [Wetherall and Tennenhouse 2019]. Apesar disso, os pesquisadores pela primeira vez imaginaram a noção de uma rede programável, que era um design radicalmente diferente se comparado às arquitetura tradicionais de redes existentes na época. As *active networks* pressupunham que dispositivos de rede seriam capazes de expor seu estado a uma interface programável, um conceito posteriormente revisitado pelo OpenFlow e pelos planos de dados programáveis [Feamster et al. 2014].

Enquanto isso, os operadores de rede careciam de mecanismos adequados para gerenciar e configurar suas redes. Tipicamente, essas redes eram compostas por dispositivos de diferentes fornecedores, incluindo protocolos diferentes, e muitas vezes exigiam muito esforço manual para configurá-las. Normalmente, o plano de dados e o plano de controle estavam totalmente integrados aos dispositivos, dificultando a execução de tarefas de engenharia de tráfego ou depuração. Para simplificar o gerenciamento, os pesquisadores começaram a defender a separação do plano de controle e do plano de dados [Feamster et al. 2014]. Vários pesquisadores deram contribuições significativas mostrando que era possível separar os planos, mas esta separação só foi concretamente alcançada com o surgimento do OpenFlow [McKeown et al. 2008].

1.3.2. Surgimento de SDN e OpenFlow

O conceito de *Software-Defined Networking* (SDN) fornece uma abstração de controlador logicamente centralizado, sendo capaz de interagir com o plano de dados usando APIs específicas. A ascensão do SDN motivou a noção de sistemas operacionais de rede, como ONOS [Berde et al. 2014] e NOX [Gude et al. 2008]. Além disso, devido às preocupações relacionadas aos riscos de ter um controlador centralizado, a comunidade SDN adotou a ideia de ter um plano de controle fisicamente distribuído (porém mantendo a consistência do estado distribuído, e fornecendo suporte para aplicações de rede em geral).

Dentre as APIs destinadas à interação entre o plano de controle e o plano de dados destaca-se o protocolo OpenFlow [McKeown et al. 2008]. O OpenFlow aproveitou uma importante abstração comum entre vários switches e roteadores: a abstração *match+action*. Usando essa abstração, uma API OpenFlow permitiu que programas externos alterassem entradas de tabelas de fluxo do switch, normalmente implementadas em TCAM. A abstração match+action era simples, porém poderosa, e rapidamente se tornou difundida porque o hardware da tabela de fluxo era comum entre os chips de switch existentes. Isso era vantajoso para os fornecedores de equipamentos pois eles não precisavam trocar completamente seus equipamentos.

O surgimento de SDN e OpenFlow permitiu que pesquisadores e operadores pudessem usar SDN para estudar e começar a implantar suas próprias soluções em suas redes. Por exemplo, um operador de rede poderia escrever uma aplicação de plano de controle para balancear efetivamente a carga na rede. Esta aplicação poderia, por exemplo, executar um algoritmo personalizado e definir novas entradas nas tabelas de encaminhamento para pacotes subsequentes de um fluxo. A estratégia de alterar as entradas da tabela se opõe à forma como o balanceamento de carga era implementado nas redes tradicionais: uma rede tradicional estaria limitada a alterar os pesos dos links do protocolo do plano de controle (por exemplo, OSPF) implementado pelo fornecedor do equipamento, não permitindo que os operadores construíssem sua própria solução.

Ao motivar o uso de uma interface aberta entre os planos de dados e de controle, as redes OpenFlow permitiram aos pesquisadores experimentar novas ideias e simplificar o gerenciamento. Embora o OpenFlow tenha trazido esses benefícios para os operadores de rede, o grau de programabilidade na rede usando SDN ainda era pequeno. Um switch OpenFlow possui uma tabela de fluxos que mapeia fluxos para uma ação específica. A única programabilidade no switch é feita alterando as entradas da tabela de fluxo por meio

da API OpenFlow para executar ações específicas. Além disso, as entradas nas tabelas de match+action do OpenFlow correspondiam apenas a um conjunto fixo de campos de cabeçalhos do pacote. Por fim, o conjunto de ações disponíveis era fixo, incluindo apenas ações de descartar, encaminhar e enviar um pacote para o controlador.

1.3.3. Programabilidade do Plano de Dados

A programabilidade do plano de dados permite que o operador da rede defina a funcionalidade do plano de dados usando artefatos de software. A funcionalidade a ser implementada nos switches é frequentemente expressa usando linguagens de domínio específico como P4 [Bosshart et al. 2014] ou POF [Song 2013] e então é adaptada em um modelo abstrato de plano de dados [Hauser et al. 2021]. O código resultante é então compilado em uma arquitetura de processamento de pacotes que suporta o modelo de plano de dados. Um pipeline de processamento de pacotes inclui uma sequência de operações que um dispositivo do plano de dados da rede executa para processar um pacote [Gunturi et al. 2005]. Os componentes de um pipeline de processamento de pacotes podem variar de acordo com as diferentes arquiteturas de pipeline.

Um exemplo concreto de arquitetura de processamento de pacotes é a Protocol Independent Switch Architecture (PISA), que generaliza o modelo Reconfigurable Match-Table (RMT) [Bosshart et al. 2013]. O modelo RMT é uma tecnologia que torna as tabelas match+action dinamicamente programáveis sem modificar o hardware. Isso distingue os switches que usam RMT dos switches OpenFlow de várias maneiras. Na arquitetura de processamento de pacotes do OpenFlow, o número de tabelas, seu pipeline, seus tipos de correspondência e tamanhos são determinados durante a fabricação. Consequentemente, estes parâmetros permanecem sempre os mesmos, limitando a flexibilidade. Em contraste com os switches OpenFlow, o RMT permite que os administradores de rede definam os campos na tabela match+action para suas necessidades específicas. Além disso, o modelo RMT permite alterar a largura e a profundidade de uma tabela match+action, suportando diferentes tamanhos de entradas. A ação disparada e o número de ações disponíveis por uma correspondência também são programáveis, podendo executar funcionalidades customizadas caso um pacote corresponda a uma entrada em uma tabela. Finalmente, o programador pode definir a topologia entre tabelas match+action no pipeline em vez de ser determinada durante a fabricação.

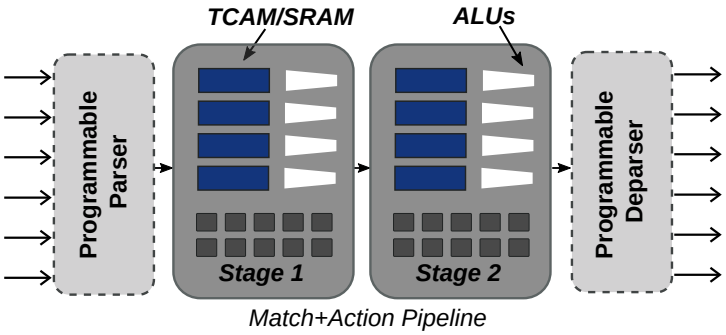


Figura 1.2. Arquitetura PISA

Arquitetura PISA. A arquitetura PISA aproveita a tecnologia RMT para fornecer

processamento de pacotes de taxa de linha. A Figura 1.2 ilustra a arquitetura PISA. Nela, os pacotes passam por um analisador de pacotes, que extrai os campos de cabeçalho necessários para o programa, como TCP, destinos IP e *metadados* importantes. *Metadados* são variáveis por pacote que armazenam valores temporários para auxiliar no processamento de pacotes, como portas de entrada e saída. Depois que o analisador processa um pacote, ele segue para um pipeline de processamento de pacotes que pode ser composto de diversas construções, como tabelas e registros *match+action*. As correspondências podem ter vários formatos, como correspondência de prefixo mais longo, correspondência ternária ou exata. Cada correspondência corresponde à execução de uma ação que executa computação e realiza operações de armazenamento. As tabelas *match+action* podem ser executadas em TCAM ou SRAM, e as ações são implementadas usando ALUs e devem ser capazes de executar na taxa de linha [Sivaraman et al. 2015]. Cada pipeline possui um conjunto fixo de estágios e cada estágio possui registradores disponíveis, que armazenam estado persistente. Cada estágio também possui um conjunto restrito de operações ALU, divididas entre operações ALU sem estado e com estado. Depois que os fluxos de controle processam um pacote, os cabeçalhos dos pacotes são emitidos por um analisador. O analisador reconstrói os cabeçalhos do pacote e o envia para uma porta de saída.

P4 - Programming Protocol-Independent Packet Processors. Linguagens de programação, como P4 [Bosshart et al. 2014] e POF [Song 2013], foram propostas para especificar a lógica de processamento de pacotes de dispositivos no plano de dados programáveis por meio de uma arquitetura de alto nível independente de abstrações. Tal linguagem pode ser usada para que operadores de rede possam rapidamente implementar novos protocolos em dispositivos de encaminhamento, personalizar suas funcionalidades e desenvolver serviços inovadores. A linguagem P4 [Bosshart et al. 2014] é uma linguagem de especificação de alto nível. A linguagem foi projetada para facilitar aos desenvolvedores a descrição do processamento de pacotes. P4 fornece uma camada de abstração acima da arquitetura PISA, fazendo com que o trabalho do compilador P4 seja mapear a funcionalidade especificada para os estágios de hardware [Hogan et al. 2022].

A linguagem P4 é frequentemente referenciada como uma linguagem declarativa [Shahbaz and Feamster 2015, Eichholz et al. 2022], com o objetivo de fornecer uma abstração de alto nível e liberar os desenvolvedores da necessidade de se preocupar como as funcionalidades são implementada no hardware do plano de dados. Os cabeçalhos dos pacotes podem ser declarados de forma semelhante a `structs` em C. O analisador (*parser*) é especificado através de uma abordagem de máquina de estado, onde os estados geralmente correspondem a uma parte do cabeçalho do pacote e as transições entre estados são transições entre cabeçalhos. Durante o tempo de execução, o analisador processa o pacote extraindo os valores dos bits do pacote para variáveis internas do programa, que podem então ser acessadas e modificadas nos demais elementos de processamento.

O código P4 é organizado logicamente da seguinte forma (veja Figura 1.3):

1. *Declaração de dados*: é uma seção que define o formato do cabeçalho do pacote e as informações de metadados que podem ser usadas para sua análise. Esta seção é mapeada em um cabeçalho e um barramento de metadados que transporta essas informações por todos os estágios de processamento. Os tipos de cabeçalho são declarados de forma semelhante às estruturas em C, ou seja, os campos são definidos

em uma ordem específica e com um tamanho pré-determinado;

2. *Parser logic*: é uma seção que especifica como, quando e em que ordem cada um dos cabeçalhos deve ser analisado. Esta seção de um programa P4 é mapeada para os elementos *parser* e *deparser* do modelo de encaminhamento. Esses elementos são então responsáveis por extrair o campo de cabeçalho dos pacotes em seu ingresso (*parser*) e no processo de tradução de estruturas de dados em um formato que possa ser armazenado e reconstruído na sequência no mesmo ou em um ambiente computacional diferente; e

3. *Match-action tables and control flows*: é uma seção que especifica tabelas capazes de fazer match em campos de cabeçalho arbitrários e realizar modificação de cabeçalhos de pacotes (e metadados) e outras ações personalizadas. Também expressa a ordem e condições nas quais cada tabela deve ser executada.

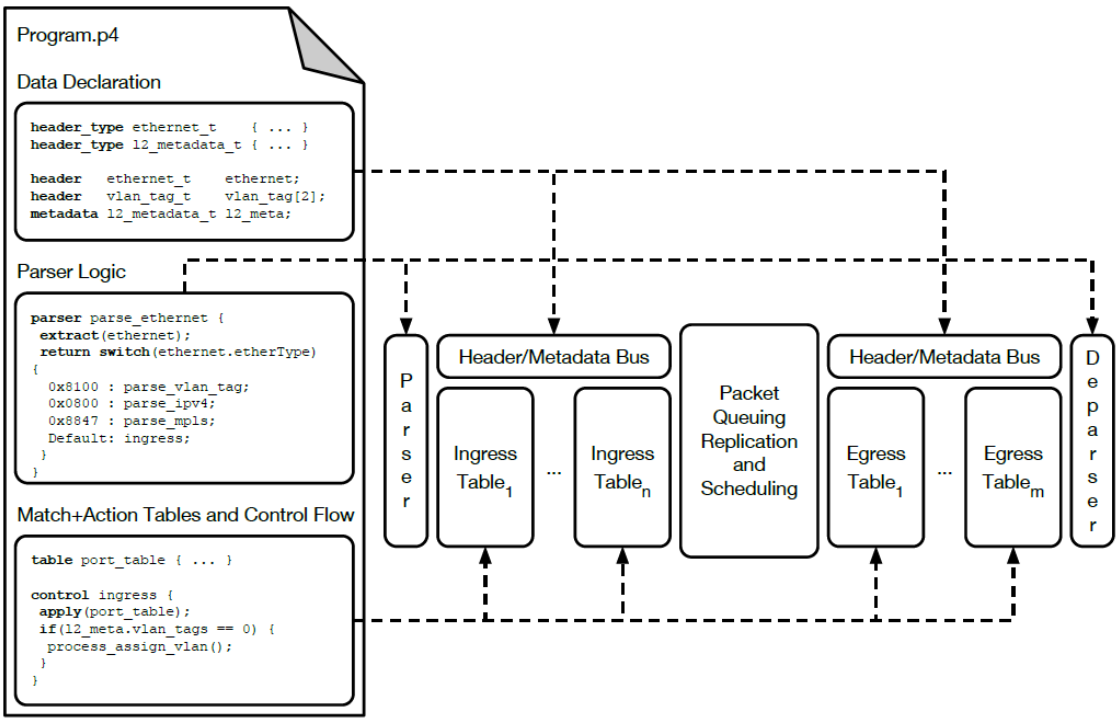


Figura 1.3. Seções de código P4 e mapeamento para o modelo de encaminhamento abstrato (Adaptado de [Kim et al. 2006])

1.4. Monitoramento e Desempenho

As redes de comunicação atuais operam com expectativas de alto desempenho em latência, largura de banda e *jitter*, especialmente com o surgimento e a proliferação de novos serviços e aplicações (por exemplo, negociação algorítmica, telecirurgia e *streaming* de vídeos de realidade virtual). Os usuários demandam garantias estritas que devem ser cumpridas, exigindo a definição de metas claras para o desempenho da rede, os chamados objetivos de nível de serviço (SLOs – *Service-Level Objectives*). Um SLO pode prescrever que o atraso fim-a-fim deve ser menor que 5 milissegundos em 95% dos pacotes de

tráfego de telecirurgia, ou que a largura de banda fornecida deve ser maior que 1 Gbps pelo menos 95% do tempo para um tráfego agregado de *streaming* de vídeo.

O monitoramento de conformidade com SLOs e o diagnóstico imediato de violações são essenciais para a operação das redes atuais e do futuro. No entanto, o monitoramento de rede é uma tarefa inerentemente difícil, às vezes comparada à busca por uma agulha no palheiro[1]. Infelizmente, as arquiteturas existentes não são projetadas para monitorar SLOs com o nível de detalhe e precisão exigidos. As ferramentas tradicionais de monitoramento passivo (por exemplo, SNMP [Fedor et al. 1990, Gaspary et al. 2005] e Net-Flow/IPFIX [Cisco Networks 2023, Aitken et al. 2013]) operam em escalas de tempo longas (dezenas de segundos ou mais) e, portanto, carecem de granularidade adequada para detectar eventos como rajadas de tráfego de curta duração (por exemplo, *micro-bursts*), que podem ser críticas para toda uma nova geração de aplicações. Técnicas de medições ativas (por exemplo, ping, traceroute, OWAMP [Zekauskas et al. 2006] e TWAMP [Babiarz et al. 2008]) também não fornecem "resolução" de tempo suficiente; além disso, não há garantia de que a rede roteará e priorizará as sondas (*probes*) da mesma maneira que os pacotes de produção. O movimento consistente em direção à heterogeneidade no tratamento de tráfego [Jeyakumar et al. 2014, Hong et al. 2013], roteamento por caminhos múltiplos [Jain et al. 2013, Kumar et al. 2015] e balanceamento de carga de fluxos [Alizadeh et al. 2014, He et al. 2015] exacerba essa limitação.

1.4.1. Telemetria de Rede em Banda

A programabilidade do plano de dados torna viável um novo método de monitoramento de rede, denominado *In-band Network Telemetry* (INT). Segundo um painel recente realizado no âmbito da iniciativa The NetworkingChannel [Foster et al. 2023], INT vem sendo considerada uma das aplicações "matadoras" (de *killer application*) por fornecer uma maneira de monitorar redes e serviços com níveis de precisão e detalhes sem precedentes, ao mesmo tempo que é capaz de operar em velocidade de linha. Em essência, o método consiste em registrar informações de operação, administração e manutenção dentro de um pacote de dados à medida que ele atravessa uma rede. Várias estruturas e técnicas foram propostas para realizar telemetria de rede em banda, por exemplo, TPP [Jeyakumar et al. 2014], INT [Kim et al. 2015] e Cisco iOAM [Cisco Sa]. Elas compartilham a ideia de permitir que pacotes de dados consultem indicadores instantâneos do estado interno de cada switch por onde passam – como tamanho e latência de filas, e utilização de enlaces – e armazenem essas informações em cabeçalhos de telemetria.

A Figura 1.4 descreve o fluxo de execução do INT [Kim et al. 2015], a estrutura de telemetria em banda atualmente mais proeminente e que pode ser implementada usando P4. A abstração da arquitetura INT é composta por um controlador de monitoramento remoto e por nós de origem, trânsito e destino, cada um dos quais representando um papel em sua instanciação. Cada switch programável no caminho de um pacote (conforme é transmitido pela rede) pode assumir uma ou mais funções. A figura ilustra um cenário em que o sistema final (*end-host*) 1 envia um pacote de dados tradicional para o sistema

[1]Everflow [Zhu et al. 2015]: "[Problem diagnosis] is not only akin to searching in the proverbial haystack for needles, but for specific needles of arbitrary size, shape and color."
Sonata [Gupta et al. 2018]: "[...] telemetry queries often require finding 'needles in a haystack' where the fraction of total traffic or flows that satisfies these queries is tiny."

final 2 por meio de uma rede de switches compatíveis com INT.

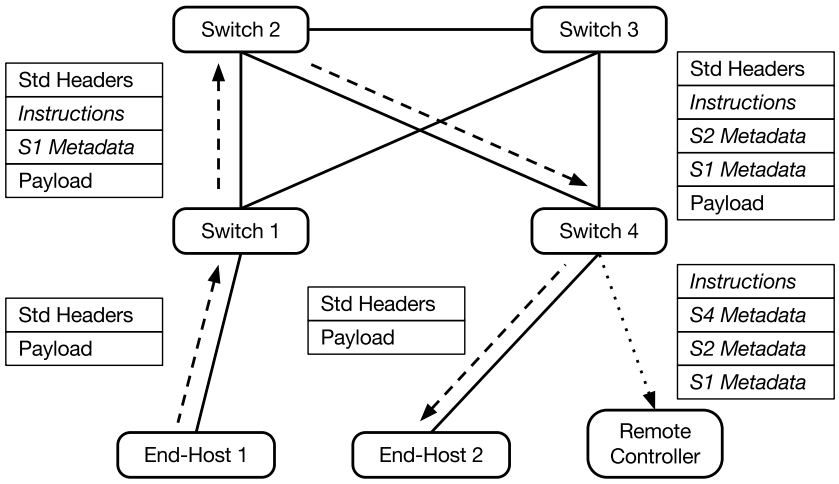


Figura 1.4. Arquitetura da estrutura INT para telemetria de rede em banda (extraída de [Marques 2022] *apud* [Laupkhov and Thomas 2016])

Nós de origem (no exemplo, *Switch 1*) são responsáveis por incorporar instruções de medição (normalmente na forma de valores de cabeçalho) em pacotes regulares ou de sondagem. *Nós de trânsito* executam as instruções e acrescentam valores medidos aos pacotes. No exemplo, *Switches 1*, *2* e *4* assumem o papel de nós de trânsito, já que o caminho do pacote é *End-Host 1* $\rightarrow$ *Switch 1* $\rightarrow$ *Switch 2* $\rightarrow$ *Switch 4* $\rightarrow$ *End-Host 2*. Por último, *sink nodes* (no exemplo, *Switch 4*) recuperam os resultados das instruções e reportam (subconjuntos apropriados delas) a um controlador. Exemplos de metadados que podem ser coletados em cada switch são o ID do switch, o ID da porta de entrada/saída, o carimbo de data/hora, a contagem de bytes, a contagem de descartes e a utilização do enlace, bem como um ID de fila, ocupação e estado de congestionamento.

Diversos desafios vêm sendo abordados pelo Grupo de Pesquisa em Redes de Computadores no âmbito de INT [Marques et al. 2019, Marques et al. 2020, Vassoler et al. 2023]. A seguir, a título de exemplo motivador, descreve-se um desses trabalhos.

1.4.2. Sistema IntSight

Capitalizando sobre as oportunidades habilitadas por INT, projetou-se e desenvolveu-se IntSight [Marques et al. 2020], um sistema para detectar e diagnosticar violações de SLO, com foco em atraso fim-a-fim e largura de banda. Em contraste com INT clássico, IntSight usa cabeçalhos de telemetria como espaços de trabalho, onde os dispositivos de encaminhamento calculam metadados *path-wise* (por exemplo, IDs de caminho, pontos de contenção e atrasos fim-a-fim) progressivamente. A detecção no plano de dados, então, torna-se uma questão de comparar os valores calculados com os definidos pelo SLO. O sistema proposto permite que dispositivos de encaminhamento resumam dados de monitoramento e *seletivamente* relatem eventos de interesse para o plano de controle, onde ocorre o *diagnóstico* das violações.

A Figura 1.5 ilustra a abordagem do IntSight para monitoramento de rede. Ela exemplifica a trajetória de um pacote de dados em uma rede monitorada pelo sistema.

Para orientar a visão geral, considere, como exemplo, um fluxo de pacotes sujeito a SLOs relacionados a atraso fim-a-fim. Os pacotes desse fluxo ingressam na rede via o sistema final A, são roteados através do caminho $[N_x, N_y, N_z]$ e deixam a rede para chegar ao sistema final B.

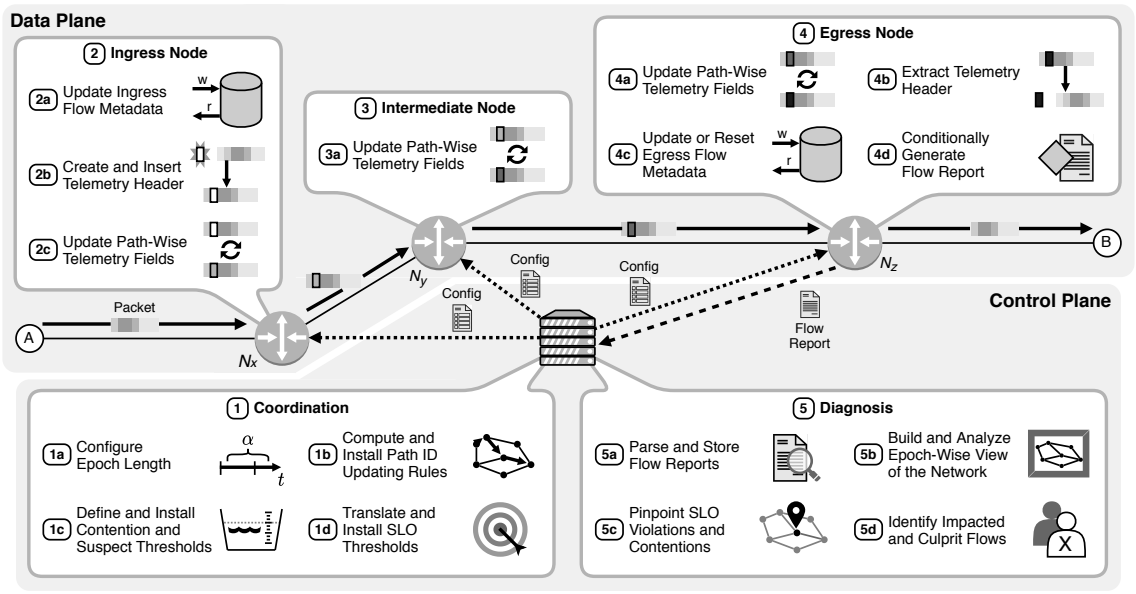


Figura 1.5. Visão geral do IntSight (extraída de [Marques et al. 2020])

No exemplo, o dispositivo N_x representa o *ingress node*, ou seja, o primeiro dispositivo na rede a receber pacotes do fluxo. Os nós de entrada têm duas tarefas principais: (*i*) armazenar metadados persistentes sobre o tráfego que ingressa na rede por meio deles e (*ii*) inicializar o processo de telemetria para os pacotes. Quando o dispositivo N_x recebe um pacote de dados vindo do sistema final A, ele segue as três etapas descritas a seguir. Inicialmente, ele atualiza os valores dos campos de metadados (que são armazenados em *arrays* de registradores) associados ao fluxo ao qual o pacote pertence (Etapa 2a na Figura 1.5). Em seguida, N_x cria e insere um cabeçalho de telemetria no pacote contendo campos como EEDelay (Etapa 2b). A lista completa de campos presentes nos cabeçalhos de telemetria é apresentada em [Marques et al. 2020].

Seguindo as duas etapas anteriores (ou seja, 2a e 2b), N_x atualiza os campos do cabeçalho de telemetria *path-wise*, de acordo com o que o pacote observou neste nó (Etapa 2c). Por exemplo, o campo CPs (*Contention Points*) é marcado caso tenha sido observada contenção no dispositivo, caracterizada pela alta ocupação da fila. O campo SPs (*Suspition Points*) é marcado quando a vazão do fluxo é responsabilizada pela alta utilização da porta ou enlace de saída, considerando sua capacidade. Para ambos os campos CPs e SPs, a contenção e a suspeita são determinadas pela comparação dos valores medidos com limites pré-configurados em tabelas de *lookup*. Por fim, o atraso ao qual o pacote foi submetido no dispositivo é adicionado ao campo EEDelay (*End-to-End Delay*).

Nós intermediários (N_y no exemplo) executam apenas uma etapa: eles atualizam os campos *path-wise* no cabeçalho de telemetria (Etapa 3a). Isso é equivalente à Etapa 2c nos nós de ingresso. Em comparação com as demandas de processamento nos nós de

ingresso e saída, a nos nós intermediários é a mais simples e que consome menos recursos. Por exemplo, nós intermediários não armazenam persistentemente quaisquer metadados relativos ao tráfego que encaminham. Isso isenta os dispositivos centrais da rede de terem a mesma capacidade de processamento e memória que os dispositivos de borda.

Um *nó de saída*, ou seja, o último dispositivo a processar um pacote antes de ele deixar a rede (N_z na Figura 1.5), é encarregado das etapas mais importantes no procedimento de monitoramento. Depois de atualizar os campos *path-wise* (Etapa 4a, igual às Etapas 2c e 3), ele extrai o cabeçalho de telemetria do pacote (Etapa 4b). Isso faz com que o pacote retorne ao seu formato original antes de ser encaminhado ao sistema final (B no exemplo). Em seguida, o nó de saída atualiza os campos de metadados persistentes em relação ao fluxo, considerando os valores dos campos de telemetria do cabeçalho extraído (Etapa 4c). Semelhante aos nós de entrada, os nós de saída armazenam campos de metadados em *arrays* de registradores.

A última etapa em um nó de saída é gerar, *condicionalmente*, relatórios de fluxo (Etapa 4d), que são pacotes de controle com todos os metadados armazenados para o respectivo fluxo durante uma época de monitoramento. Um relatório é gerado se, e somente se, uma época acabou de terminar *e* um evento de interesse foi observado para o fluxo. Um evento de interesse pode ser uma violação de SLO, uma contenção experimentada pelo fluxo em um dispositivo ou uma suspeita em relação ao fluxo (de ser agressor). Ao final dessa etapa, o relatório gerado é enviado ao plano de controle para ser analisado e informar o diagnóstico do problema, conforme descrito a seguir.

A aplicação do IntSight que executa no plano de controle possui duas tarefas principais: a coordenação do processo de monitoramento e o diagnóstico de problemas. As etapas relacionadas a ambas as tarefas estão ilustradas na seção inferior da Figura 1.5. Focando no diagnóstico de problemas, enquanto os dispositivos do plano de dados processam e encaminham pacotes, o plano de controle "escuta" continuamente os relatórios de fluxo provenientes desses dispositivos e executa as etapas de diagnóstico a seguir. Primeiro, ele analisa os relatórios recebidos e armazena suas informações em um banco de dados de metadados (Etapa 5a). Essa etapa salva informações de forma persistente, permitindo análises históricas e em tempo real. Os relatórios são armazenados no banco de dados considerando características temporais, ou seja, sua época. IntSight constrói uma visão global da rede para cada época e analisa seu estado, comportamento e desempenho em busca de eventos de interesse (Etapa 5b).

Quando um evento de interesse (ou seja, violação, contenção ou suspeição) é detectado, o IntSight usa as informações relatadas para identificar onde (ou seja, em quais dispositivos) tal evento aconteceu (Etapa 5c). Por fim, o IntSight diagnostica o evento de interesse, analisando o comportamento de todos os fluxos presentes nos dispositivos identificados e indicando as vítimas e os culpados (Etapa 5d).

1.5. In-network Computing

In-network computing (INC) ou computação em rede é um paradigma emergente na área de redes programáveis onde uma parte significativa do processamento de um sistema distribuído é descarregada dos servidores para o plano de dados dos dispositivos de rede [Michel et al. 2021, Benson 2019, Sapio et al. 2017]. A ideia por trás do INC é realizar

tarefas de computação próximas à fonte de dados, reduzindo a sobrecarga de transmissão de dados e melhorando assim a eficiência geral do sistema.

Além disso, ao invés de adicionar novos equipamentos à infraestrutura, a computação em rede concentra-se em dispositivos que já existem na infraestrutura para encaminhar o tráfego. Isso se traduz na redução e necessidade de equipamentos especializados adicionais, como aceleradores ou middleboxes.

1.5.1. Princípios da In-network Computing

A adoção de uma estratégia de descarregamento ou *offloading* para a rede é motivada principalmente pelas vantagens de desempenho que ela pode proporcionar. Com INC, é possível **reduzir a latência** interceptando e processando pacotes no plano de dados dos dispositivos de rede em vez de enviar pacotes para serem processados pelos servidores finais. Dessa forma, em vez de exigir uma solicitação para concluir um RTT inteiro enviando o pacote a um servidor, alguns pacotes podem ser rapidamente processados no hardware da rede e encaminhados ao seu destino final. Além disso, ao processar pacotes nos dispositivos de rede, o INC também **economiza largura de banda** entre o switch que executa o INC e os servidores que executam a funcionalidade do host final. Ao economizar largura de banda, é possível evitar congestionamentos e obstruções que muitas vezes causam ocupação de buffer, perdas de pacotes e degradação de desempenho.

Por fim, a **redução do consumo de energia** é uma vantagem surpreendente do paradigma INC. Switches programáveis que hospedam INCs consomem recursos iguais ou menores que os tradicionais. Ao descarregar funcionalidades INC para um dispositivo programável, é possível observar a mesma quantidade de consumo de recursos observada em um switch ocioso [Tokusashi et al. 2019]. Além disso, aumentar a taxa de tráfego aumenta o consumo de energia de acordo com uma taxa constante. Portanto, quando comparados aos servidores, que podem dobrar o consumo à medida que a taxa aumenta, os switches podem escalar para taxas de tráfego maiores. No entanto, embora o consumo de um servidor seja geralmente menor do que um switch, se assumirmos que os switches já estão disponíveis na infraestrutura, é possível economizar o consumo do servidor movendo a funcionalidade usando INC [Tokusashi et al. 2019].

1.5.2. Exemplos de Sistemas INC: NetLock, NetGVT e SwitchML

Esforços recentes mostraram muitas aplicações que podem ser usadas na rede. Os casos de uso mais simples são funções de rede, como detecção de DDoS, balanceamento de carga e NAT. Casos de uso mais sofisticados incluem sistemas para treinamento e inferência de Machine Learning [Sanvito et al. 2018], que foi estudado em diversas aplicações, como classificação de tráfego, controle de braços robóticos e até visão computacional [Glebke et al. 2019]. O armazenamento em cache de pares chave-valor [Jin et al. 2017, Jin et al. 2018] e o gerenciamento de locks [Yu et al. 2020] também são possíveis, permitindo transações rápidas de bancos de dados em data centers.

Para melhor ilustrar o uso e as aplicações de INC, apresentamos a seguir um conjunto de sistemas do estado-da-arte:

Controle de concorrência. Realizar o controle de concorrência dentro da rede em switches permite o processamento de transações com RTT reduzido em comparação

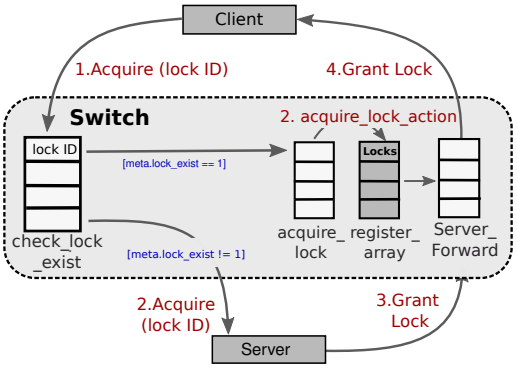


Figura 1.6. NetLock

com abordagens baseadas em servidores. No sistema Netlock [Yu et al. 2020], quando um cliente deseja adquirir um *lock* para um objeto, ele primeiro tenta obtê-lo do switch. O switch verifica em uma tabela de encaminhamento se ele é responsável por gerenciar aquele objeto. Se o *lock* existir no switch, o pacote é processado por outra tabela responsável por chamar uma ação que grava o *lock* em um array de registradores persistentes. Por outro lado, caso o switch não seja o responsável, a solicitação do *lock* será encaminhada normalmente para um servidor. Esse funcionamento é ilustrado na Figura 1.6.

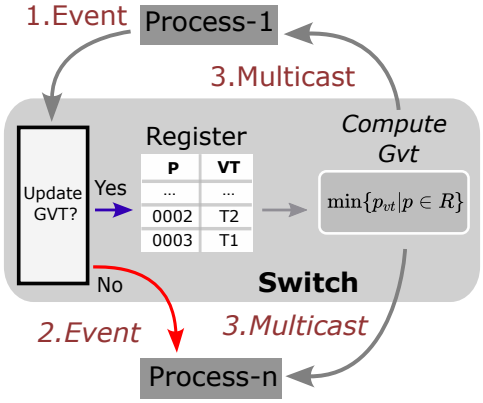


Figura 1.7. NetGVT

Sincronização de eventos. NetGVT [Parizotto et al. 2022] é um sistema que emprega in-network computing para realizar a sincronização eficiente de eventos em sistemas distribuídos. Ele foi projetado para descarregar o cálculo de um tempo virtual global (GVT) em switches de rede, permitindo-lhes sincronizar eventos com um RTT reduzido em comparação com uma solução tradicional baseada em servidores. O sistema opera interceptando pacotes de eventos enviados entre processos em execução em servidores e armazenando a *clock* virtual local do processo remetente em um registrador dentro do switch. O sistema então realiza uma comparação entre o tempo virtual armazenado nos registradores e o tempo virtual mínimo de todos os processos existentes no sistema. Se o tempo virtual do processo for o novo mínimo, o switch executa uma ação de registro que grava o novo mínimo em um registrador persistente e transmite o novo valor para todos os processos. Por outro lado, se o tempo virtual do processo não for o novo mínimo, o pacote

de eventos é encaminhado normalmente ao seu destino. Este funcionamento é ilustrado na Figura 1.7.

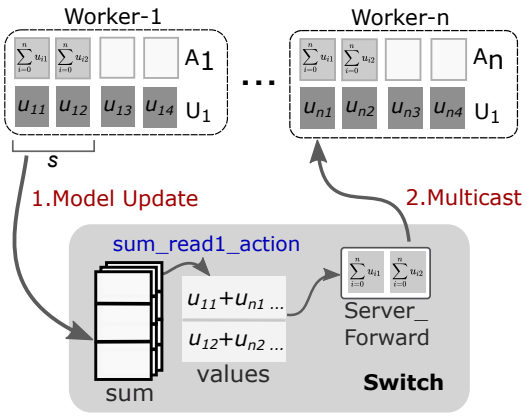


Figura 1.8. SwitchML

Agregação na rede. SwitchML [Sapio et al. 2021] é um sistema para acelerar o treinamento de aprendizado de máquina distribuído que descarrega a agregação de parâmetros de treinamento (gradientes) para o hardware do switch. Ao realizar a agregação de parâmetros nos switches, essa computação ocorre mais perto dos trabalhadores e evita a sobrecarga de comunicação imposta por soluções tradicionais baseadas em servidores. Uma tabela mapeia um pacote para uma ação que agrega os parâmetros de treinamento de cada trabalhador e os armazena em um conjunto de registradores. O gradiente resultante é encapsulado em um pacote e enviado de volta aos trabalhadores para atualizar seu modelo local. Este processo se repete para todas as partes do modelo de forma síncrona. O SwitchML oferece diversas vantagens em relação aos sistemas de agregação tradicionais, incluindo melhor escalabilidade, latência reduzida e maior eficiência devido à proximidade da agregação com os trabalhadores e às capacidades de taxa de linha dos switches. Este funcionamento é ilustrado na Figura 1.8.

1.6. Redes Móveis Programáveis

A evolução das redes móveis tem sido marcada por avanços significativos, sendo as transições para as redes de quarta e quinta geração (4G e 5G) dois dos marcos recentes mais importantes. Uma mudança notável nesse processo tem sido a transição de implementações baseadas em hardware de propósito específico, desenvolvidas por um grupo restrito de grandes fabricantes, para implementações baseadas em software, inclusive de código aberto, e em tecnologias como o Rádio Definido por Software (*Software-Defined Radio – SDR*) [Bonati et al. 2020]. Essa mudança é relevante do ponto de vista científico e tecnológico, pois permite maior flexibilidade e participação de diversos atores no desenvolvimento de componentes de rede. Além disso, torna cada vez mais economicamente viável a implantação de redes móveis (5G ou 4G) privadas, sem vínculo com operadoras utilizando software aberto e hardware de propósito geral [Aijaz 2020]. Atualmente, essa mudança de paradigma permite uma inovação mais distribuída tanto nos componentes de Rede de Acesso via Rádio (*Radio Access Network – RAN*) quanto na Rede de Núcleo (*Core Network – CN*).

Projetos de software de código aberto, juntamente com a tecnologia SDR, têm desempenhado um papel crucial na promoção da inovação na academia e na indústria de telecomunicações. Entre os projetos mais notáveis estão o Open Air Interface[2] e o srsRAN[3], que oferecem implementações completas e funcionais dos componentes da RAN compatíveis com os padrões do 3GPP (3rd Generation Partnership Project), tanto para 4G quanto para 5G, e podem ser executados sobre dispositivos SDR programáveis via FPGA, como Ettus USRP, LimeSDR e Nuand bladeRF [Mihai et al. 2022]. Isso permite que pesquisadores conduzam experimentos realistas e contribuam para o desenvolvimento de futuras gerações de redes móveis, uma vez que podem modificar e personalizar essas implementações de acordo com suas necessidades de pesquisa.

Além disso, projetos como o Open5GS[4] e o Free5GC[5] fornecem implementações funcionais dos componentes de núcleo da rede, que são responsáveis pelo registro e autenticação de usuários, gestão de mobilidade, fatiamento de rede, entre outras funções. Isso amplia ainda mais as oportunidades para a pesquisa e desenvolvimento em redes móveis de próxima geração. Essas implementações abertas associadas a tecnologias nativas da nuvem (*cloud-native*), como o Docker e Kubernetes, vem sendo gradualmente adotadas pela indústria de telecomunicações [Sekigawa et al. 2022]. Elas desempenham um papel primordial na transformação das redes móveis, permitindo a virtualização e orquestração de recursos de rede de forma eficiente e escalável, o que é essencial para a revolução baseada em software que está ocorrendo nas redes móveis.

Outra iniciativa importante para a democratização das redes de telecomunicações de futura geração é a OpenRAN (O-RAN) Alliance [Garcia-Saavedra and Costa-Pérez 2021]. No Brasil, encabeçada pela Rede Nacional de Ensino e Pesquisa (RNP), temos a OpenRAN@Brasil[6]. Esses projetos visam abrir o mercado de RAN para mais fornecedores, promovendo a concorrência e a inovação. Eles também apresentam uma arquitetura aberta que permite a construção de redes mais flexíveis e personalizáveis, fortemente baseadas em tecnologias de Inteligência Artificial, adaptando as redes às necessidades específicas de diferentes operadoras, usuários, aplicações e serviços.

As iniciativas e projetos de software de código aberto e mencionados trazem consigo uma mudança fundamental no cenário das telecomunicações e uma série de desafios e oportunidades de pesquisa. Anteriormente, apenas um pequeno grupo de engenheiros experientes das grandes empresas fornecedoras de equipamentos de redes tinham acesso ao código para atuar no desenvolvimento de protocolos padronizados. No entanto, a democratização proporcionada pelos projetos de código aberto faz com que qualquer pessoa possa contribuir efetivamente para esses projetos e influenciar seu desenvolvimento. Nesse contexto, o teste de software de redes se torna uma peça crucial para garantir que as implementações sejam corretas, conformes e robustas [Lando et al. 2023].

Os testes de conformidade desempenham um papel fundamental na garantia de que as implementações de software aberto para redes 5G e de gerações futuras estejam

[2]Site do projeto Open Air Interface: https://openairinterface.org/
[3]Site do projeto srsRAN: https://www.srsran.com/
[4]Site do projeto Open5GS: https://open5gs.org/
[5]Site do projeto Free5GC: https://free5gc.org/
[6]Site oficial da iniciativa: https://openranbrasil.org.br/

em conformidade com os padrões estabelecidos por organismos internacionais, como o 3GPP, responsável pelas especificações de redes móveis atuais (e.g., LTE, 5G-NR, etc.). Esses testes verificam se os protocolos e funcionalidades estão sendo implementados de acordo com as especificações técnicas, garantindo a interoperabilidade entre diferentes componentes de rede e dispositivos de usuários. Isso é essencial, uma vez que a heterogeneidade de implementações em um ambiente de código aberto pode levar a problemas de compatibilidade.

Além dos testes de conformidade, os testes de robustez são igualmente cruciais. Eles visam identificar vulnerabilidades e pontos fracos nas implementações de software, tornando-as mais resistentes a falhas, ataques e condições adversas, maliciosas ou não. Com a natureza aberta das implementações de software, é importante que a comunidade de desenvolvedores trabalhe em conjunto para identificar e corrigir vulnerabilidades, e os testes de robustez auxiliam nesse processo.

Outro aspecto essencial é a realização de testes de desempenho padronizados, também conhecidos como benchmarks. Esses testes avaliam o desempenho das implementações de software em cenários realistas de uso. Para redes privadas 5G, por exemplo, os benchmarks podem fornecer medições de latência, taxa de bits, qualidade de serviço e escalabilidade, considerando uma variedade de aplicações de interesse (e.g., vídeo, voz, jogos, IoT). Essas métricas são vitais para garantir que não apenas as implementações de software atendam às expectativas de desempenho de operadoras e usuários finais, mas também para auxiliar na identificação de gargalos das cada implementações e sua relação com as plataformas de hardware sobre as quais são implantadas.

Em termos de oportunidades de pesquisa e inovação, o campo de desenvolvimento e testes de software para redes abertas 5G e gerações futuras é vasto. Os membros do Grupo de Pesquisa em Redes de Computadores vêm se concentrando no desenvolvimento de metodologias avançadas de testes para software de redes móveis, na criação de ferramentas automatizadas de teste e na exploração de técnicas de análise de desempenho atuando em projetos como o PORVIR-5G (Programmability, ORchestration and VIRtualization of 5G Networks)[7]. Além disso, o grupo vem instalando e aprimorando um ambiente de testes (testbed) realístico com equipamentos e sistemas para realizar pesquisas nessa área utilizando as plataformas abertas mencionadas além de tecnologias de computação em nuvem e borda.

1.7. Desafios e Oportunidades em Redes Programáveis

As infraestruturas de redes e serviços, e a própria Internet, estão passando por mudanças dramáticas, com avanços rápidos sendo feitos em tópicos como IoT, 5G e, mais recentemente, 6G. Esses sistemas prometem uma super alta taxa de transferência e baixas latências fim-a-fim, e será interessante observar sua evolução. Igualmente importante é o surgimento de aplicações como realidade virtual e aumentada, aplicações holográficas, veículos autônomos e IoT industrial, que levantam à questão de se está pronto para suportá-las e se dispõe das ferramentas necessárias para gerenciá-las. Redes programáveis têm o potencial de fornecer a flexibilidade necessária para projetar soluções de rede que atendam aos requisitos operacionais cada vez mais rigorosos desses serviços emer-

[7] Site do projeto PORVIR-5G: https://porvir-5g-project.github.io/

gentes. Existem oportunidades enormes para pesquisas ainda mais impactantes, tanto de natureza mais aplicada quanto fundamental. Algumas delas estão resumidas a seguir.

Avanços na computação por pacote. Realizar cálculos por pacote usando planos de dados programáveis requer um esforço significativo. As primitivas atuais são inadequadas até mesmo para tarefas de gerenciamento básicas. Superar limitações de linguagem ou hardware requer considerável criatividade. A pergunta é: como se pode progredir nessa área? Os desafios residem em compreender as demandas e formular construções que equilibrem os *trade-offs* entre flexibilidade e segurança, flexibilidade e desempenho, e outros fatores.

Equilibrar os ciclos de controle curtos (plano de dados) e longos (plano de controle). Ao projetar uma solução de gerenciamento definida por software, o desenvolvedor muitas vezes se depara com o dilema de determinar o que implementar nos planos de dados e de controle. O debate sobre o gerenciamento por delegação, que ocorreu na comunidade de pesquisa na década de 1990, permanece relevante até hoje. Pode valer a pena revisitar os fundamentos desse debate e alinhá-los com a realidade tecnológica atual. Ao fazê-lo, poder-se-ia aproveitar essas valiosas contribuições para orientar essa questão.

Abstrações de linguagem adequadas para PDPs. Em nosso grupo de pesquisa, provou-se complicada a utilização das abstrações oferecidas por linguagens para planos de dados programáveis. Defende-se a disponibilização de abstrações de linguagem apropriadas, pois elas poderiam simplificar a programação do plano de dados e reduzir a probabilidade de erros de programação. Apesar dos desafios envolvidos, há espaço para revisitar a pesquisa pioneira em políticas de rede (ou *intents*). Essas políticas ditariam o comportamento esperado da rede e poderiam ser refinadas para produzir código P4.

Sistemas convenientes de compilação e implantação. Uma possível área de pesquisa reside no desenvolvimento de sistemas de compilação e implantação simplificados. As ferramentas disponíveis atualmente são bastante rudimentares, tornando o processo de compilar programas para alvos de rede específicos uma tarefa complexa, devido à falta de mecanismos de depuração e solução de problemas. Da mesma forma, a implantação de programas, atualmente, é um processo relativamente *ad hoc*, dependente de interfaces pouco adequadas. Dadas essas limitações, seria interessante investigar o potencial de uma abordagem DevOps para redes programáveis, apoiada por técnicas robustas de gerenciamento de ciclo de vida de software.

Gerenciamento orientado por IA e ML na rede. A integração de técnicas de IA e ML tem recebido muita atenção para auxiliar em diferentes tarefas de gerenciamento de rede e serviços. Uma vez que os programas do plano de dados operam na taxa de linha e os dispositivos possuem portas com dezenas de gigabits por segundo, vale a pena explorar a possibilidade de transferir mecanismos baseados em ML para os dispositivos.

Quais técnicas de ML são adequadas para planos de dados? É viável utilizar métodos simplificados? Essa é uma área de pesquisa promissora.

Gerenciamento de redes híbridas. Redes híbridas, compostas por dispositivos tradicionais e programáveis, estão se tornando cada vez mais comuns. Em tais configurações, a implementação de uma solução de monitoramento baseada em Telemetria de Rede em Banda requer levar em consideração os dispositivos na infraestrutura que não podem ser programados. Para obter medições detalhadas, é crucial identificar as localizações ideais para implantar dispositivos programáveis, minimizando quaisquer possíveis "sacrifícios" na precisão. O desenvolvimento de abordagens de gerenciamento para essas redes híbridas apresenta uma área de pesquisa promissora que pode produzir resultados significativos.

De construções *ad hoc* para bibliotecas de gerenciamento reutilizáveis. O processo de implementação de procedimentos de gerenciamento normalmente envolve a criação de construções "do zero". No entanto, há espaço para novos desenvolvimentos que permitam evoluir de construções *ad hoc* para bibliotecas de gerenciamento reutilizáveis (para planos de dados programáveis). Tal evolução facilitaria o desenvolvimento de procedimentos corretos e reutilizáveis.

Planos de dados confiáveis. A pesquisa em planos de dados confiáveis tem recebido muita atenção nos últimos anos devido à sua relevância na programação de redes. O problema de garantir a correção dos programas de rede continua é chave. A literatura apresenta inúmeras propostas de métodos e sistemas de verificação que se concentram na detecção de erros ou na garantia de conformidade com propriedades desejadas. A maioria dessas soluções depende de abordagens demoradas, como análise estática ou execução simbólica do código. No entanto, dada a natureza altamente dinâmica das redes e fluxos, ainda há espaço considerável para melhorar a escalabilidade. Além disso, a questão da segurança continua sendo uma preocupação premente. Em caso de injeção de código malicioso em um dispositivo de plano de dados, o comportamento da rede pode ser comprometido. Portanto, mais pesquisas são necessárias para aprimorar a segurança dos planos de dados programáveis.

Aplicativos conscientes de gerenciamento. Operar em um plano de dados pode proporcionar vantagens se tal funcionalidade vier acompanhada da possibilidade de acesso às informações mantidas pelas aplicações. Investigar como isso pode ser realizado de forma sistemática e modular apresenta um problema interessante para exploração adicional.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

[Aijaz 2020] Aijaz, A. (2020). Private 5g: The future of industrial wireless. *IEEE Industrial Electronics Magazine*, 14(4):136–145.

[Aitken et al. 2013] Aitken, P., Claise, B., and Trammell, B. (2013). Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information. RFC 7011.

[Alizadeh et al. 2014] Alizadeh, M., Edsall, T., Dharmapurikar, S., Vaidyanathan, R., Chu, K., Fingerhut, A., Lam, V. T., Matus, F., Pan, R., Yadav, N., and Varghese, G. (2014). Conga: Distributed congestion-aware load balancing for datacenters. In *Proceedings of the 2014 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '14, pages 503–514, New York, NY, USA. ACM.

[Babiarz et al. 2008] Babiarz, J., Krzanowski, R. M., Hedayat, K., Yum, K., and Morton, A. (2008). A Two-Way Active Measurement Protocol (TWAMP). RFC 5357.

[Benson 2019] Benson, T. A. (2019). In-Network Compute: Considered Armed and Dangerous. In *Proceedings of the Workshop on Hot Topics in Operating Systems*, HotOS '19, pages 216–224, New York, NY, USA. ACM.

[Berde et al. 2014] Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., Lantz, B., O'Connor, B., Radoslavov, P., Snow, W., et al. (2014). Onos: towards an open, distributed sdn os. In *Proceedings of the third workshop on Hot topics in software defined networking*, pages 1–6.

[Bonati et al. 2020] Bonati, L., Polese, M., D'Oro, S., Basagni, S., and Melodia, T. (2020). Open, programmable, and virtualized 5g networks: State-of-the-art and the road ahead. *Computer Networks*, 182:107516.

[Bosshart et al. 2014] Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., and Walker, D. (2014). P4: Programming Protocol-independent Packet Processors. *SIGCOMM Comput. Commun. Rev.*

[Bosshart et al. 2013] Bosshart, P., Gibb, G., Kim, H.-S., Varghese, G., McKeown, N., Izzard, M., Mujica, F., and Horowitz, M. (2013). Forwarding metamorphosis: Fast programmable match-action processing in hardware for sdn. *ACM SIGCOMM Computer Communication Review*, 43(4):99–110.

[Cisco Sa] Cisco (S.a.). In-band OAM (iOAM). https://github.com/CiscoDevNet/iOAM.

[Cisco Networks 2023] Cisco Networks (2023). Cisco IOS NetFlow. http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html.

[Clark 2003] Clark, D. D. (2003). The tussle in cyberspace: defining tomorrow's Internet. *ACM SIGCOMM Computer Communication Review*, 33(1):347–356.

[Comer 2017] Comer, D. E. (2017). *Internetworking with TCP/IP, Vol. 1: Principles, Protocols, and Architecture*. Pearson.

[Eichholz et al. 2022] Eichholz, M., Campbell, E. H., Krebs, M., Foster, N., and Mezini, M. (2022). Dependently-typed data plane programming. *Proceedings of the ACM on Programming Languages*, 6(POPL):1–28.

[Feamster et al. 2014] Feamster, N., Rexford, J., and Zegura, E. (2014). The Road to SDN: An Intellectual History of Programmable Networks. *SIGCOMM Comput. Commun. Rev.*, 44(2):87–98.

[Fedor et al. 1990] Fedor, M., Schoffstall, M. L., Davin, J. R., and Case, D. J. D. (1990). Simple Network Management Protocol (SNMP). RFC 1157.

[Foster et al. 2023] Foster, N., McKeown, N., and Rexford, J. (2023). Network programmability: The road ahead. In *The Networking Channel*. https://networkingchannel.eu/network-programmability-the-road-ahead/.

[Garcia-Saavedra and Costa-Pérez 2021] Garcia-Saavedra, A. and Costa-Pérez, X. (2021). O-ran: Disrupting the virtualized ran ecosystem. *IEEE Communications Standards Magazine*, 5(4):96–103.

[Gaspary et al. 2005] Gaspary, L., Sanchez, R., Antunes, D., and Meneghetti, E. (2005). A snmp-based platform for distributed stateful intrusion detection in enterprise networks. *IEEE Journal on Selected Areas in Communications*, 23(10):1973–1982.

[Glebke et al. 2019] Glebke, R., Krude, J., Kunze, I., Rüth, J., Senger, F., and Wehrle, K. (2019). Towards executing computer vision functionality on programmable network devices. In *Proceedings of the 1st ACM CoNEXT Workshop on Emerging in-Network Computing Paradigms*, pages 15–20.

[Gude et al. 2008] Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., and Shenker, S. (2008). Nox: towards an operating system for networks. *ACM SIGCOMM computer communication review*, 38(3):105–110.

[Gunturi et al. 2005] Gunturi, R., Johnson, E., and Seow, C. (2005). Packet processing pipeline. US Patent App. 10/766,282.

[Guo and McKeown 2018] Guo, Y. and McKeown, N. (2018). *Network Function Virtualization*. Morgan & Claypool.

[Gupta et al. 2018] Gupta, A., Harrison, R., Canini, M., Feamster, N., Rexford, J., and Willinger, W. (2018). Sonata: Query-driven streaming network telemetry. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '18, pages 357–371, New York, NY, USA. ACM.

[Hauser et al. 2021] Hauser, F., Häberle, M., Merling, D., Lindner, S., Gurevich, V., Zeiger, F., Frank, R., and Menth, M. (2021). A survey on data plane programming with p4: Fundamentals, advances, and applied research. *arXiv preprint arXiv:2101.10632*.

[He et al. 2015] He, K., Rozner, E., Agarwal, K., Felter, W., Carter, J., and Akella, A. (2015). Presto: Edge-based load balancing for fast datacenter networks. In *Proceedings of the 2015 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '15, pages 465–478, New York, NY, USA. ACM.

[Hogan et al. 2022] Hogan, M., Landau-Feibish, S., Arashloo, M. T., Rexford, J., and Walker, D. (2022). Modular switch programming under resource constraints. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pages 193–207.

[Hong et al. 2013] Hong, C.-Y., Kandula, S., Mahajan, R., Zhang, M., Gill, V., Nanduri, M., and Wattenhofer, R. (2013). Achieving high utilization with software-driven wan. In *Proceedings of the 2013 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '13, pages 15–26, New York, NY, USA. ACM.

[Jain et al. 2013] Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., Zolla, J., Hölzle, U., Stuart, S., and Vahdat, A. (2013). B4: Experience with a globally-deployed software defined wan. In *Proceedings of the 2013 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '13, pages 3–14, New York, NY, USA. ACM.

[Jeyakumar et al. 2014] Jeyakumar, V., Alizadeh, M., Geng, Y., Kim, C., and Mazières, D. (2014). Millions of little minions: Using packets for low latency network programming and visibility. In *Proceedings of the 2014 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '14, pages 3–14, New York, NY, USA. ACM.

[Jin et al. 2018] Jin, X., Li, X., Zhang, H., Foster, N., Lee, J., Soulé, R., Kim, C., and Stoica, I. (2018). Netchain: Scale-free sub-rtt coordination. In *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)*, pages 35–49.

[Jin et al. 2017] Jin, X., Li, X., Zhang, H., Soulé, R., Lee, J., Foster, N., Kim, C., and Stoica, I. (2017). Netcache: Balancing key-value stores with fast in-network caching. In *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP '17, page 121–136, New York, NY, USA. Association for Computing Machinery.

[Kim et al. 2015] Kim, C., Sivaraman, A., Katta, N., Bas, A., Dixit, A., and Wobker, L. J. (2015). In-band network telemetry via programmable dataplanes. In *Proceedings of the 2015 ACM Symposium on SDN Research*, SOSR'15, New York, NY, USA. ACM.

[Kim et al. 2006] Kim, Y., Lau, W. C., Chuah, M. C., and Chao, H. J. (2006). PacketScore: a statistics-based packet filtering scheme against distributed denial-of-service attacks. *IEEE transactions on dependable and secure computing*, 3(2):141–155.

[Kumar et al. 2015] Kumar, A., Jain, S., Naik, U., Raghuraman, A., Kasinadhuni, N., Zermeno, E. C., Gunn, C. S., Ai, J., Carlin, B., Amarandei-Stavila, M., Robin, M., Siganporia, A., Stuart, S., and Vahdat, A. (2015). Bwe: Flexible, hierarchical bandwidth allocation for wan distributed computing. In *Proceedings of the 2015 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '15, pages 1–14, New York, NY, USA. ACM.

[Lando et al. 2023] Lando, G., Schierholt, L. A. F., Milesi, M. P., and Wickboldt, J. A. (2023). Evaluating the performance of open source software implementations of the 5g network core. In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pages 1–7.

[Laupkhov and Thomas 2016] Laupkhov, P. and Thomas, J. (2016). Using int to build a real-time network monitoring system @ scale. 3rd P4 Workshop. https://2016p4workshop.sched.com/event/6otq/using-int-to-build-a-real-time-network-monitoring-system-scale.

[Liu et al. 2021] Liu, P., Kim, H., Li, Y., Xu, X., and Wang, X. (2021). Network Programming and Automation: A Survey. *IEEE Communications Surveys & Tutorials*, 23(4):3029–3058.

[Marques et al. 2020] Marques, J., Levchenko, K., and Gaspary, L. (2020). Intsight: Diagnosing slo violations with in-band network telemetry. In *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '20, page 421–434, New York, NY, USA. Association for Computing Machinery.

[Marques 2022] Marques, J. A. (2022). Advancing network monitoring and operation with in-band network telemetry and data plane programmability.

[Marques et al. 2019] Marques, J. A., Luizelli, M. C., da Costa Filho, R. I. T., and Gaspary, L. P. (2019). An optimization-based approach for efficient network monitoring using in-band network telemetry. *J. Internet Serv. Appl.*, 10(1):12:1–12:20.

[McKeown et al. 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review*, 38(2):69–74.

[Michel et al. 2021] Michel, O., Bifulco, R., Rétvári, G., and Schmid, S. (2021). The programmable data plane: Abstractions, architectures, algorithms, and applications. *ACM Computing Surveys (CSUR)*, 54(4):1–36.

[Mihai et al. 2022] Mihai, R., Craciunescu, R., Martian, A., Li, F. Y., Patachia, C., and Vochin, M.-C. (2022). Open-source enabled beyond 5g private mobile networks: From concept to prototype. In *2022 25th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 181–186.

[Parizotto et al. 2022] Parizotto, R., Mello, B., Haque, I., and Schaeffer-Filho, A. (2022). Netgvt: Offloading global virtual time computation to programmable switches. In *Proceedings of the Symposium on SDN Research*, SOSR '22, page 16–24, New York, NY, USA. Association for Computing Machinery.

[Sanvito et al. 2018] Sanvito, D., Siracusano, G., and Bifulco, R. (2018). Can the network be the AI accelerator? In *Proceedings of the 2018 Morning Workshop on In-Network Computing*, pages 20–25.

[Sapio et al. 2017] Sapio, A., Abdelaziz, I., Aldilaijan, A., Canini, M., and Kalnis, P. (2017). In-Network Computation is a Dumb Idea Whose Time Has Come. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, HotNets-XVI.

[Sapio et al. 2021] Sapio, A., Canini, M., Ho, C.-Y., Nelson, J., Kalnis, P., Kim, C., Krishnamurthy, A., Moshref, M., Ports, D., and Richtarik, P. (2021). Scaling distributed machine learning with in-network aggregation. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 785–808. USENIX Association.

[Sekigawa et al. 2022] Sekigawa, S., Sasaki, C., and Tagami, A. (2022). Toward a cloud-native telecom infrastructure: Analysis and evaluations of kubernetes networking. In *2022 IEEE Globecom Workshops (GC Wkshps)*, pages 838–843.

[Shahbaz and Feamster 2015] Shahbaz, M. and Feamster, N. (2015). The case for an intermediate representation for programmable data planes. In *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research*, pages 1–6.

[Sivaraman et al. 2015] Sivaraman, A., Budiu, M., Cheung, A., Kim, C., Licking, S., Varghese, G., Balakrishnan, H., Alizadeh, M., and McKeown, N. (2015). Packet transactions: A programming model for data-plane algorithms at hardware speed. *CoRR, vol. abs/1512.05023*.

[Song 2013] Song, H. (2013). Protocol-Oblivious Forwarding: Unleash the Power of SDN through a Future-Proof Forwarding Plane. In *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, HotSDN '13, page 127–132, New York, NY, USA. Association for Computing Machinery.

[Tennenhouse and Wetherall 1996] Tennenhouse, D. L. and Wetherall, D. J. (1996). Towards an active network architecture. *ACM SIGCOMM Computer Communication Review*, 26(2):5–17.

[Tokusashi et al. 2019] Tokusashi, Y., Dang, H. T., Pedone, F., Soulé, R., and Zilberman, N. (2019). The case for in-network computing on demand. In *Proceedings of the Fourteenth EuroSys Conference 2019*, pages 1–16.

[Vassoler et al. 2023] Vassoler, G., Marques, J. A., and Gaspary, L. P. (2023). Vermont: Towards an in-band telemetry-based approach for live network property verification. In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pages 1–7.

[Wetherall and Tennenhouse 2019] Wetherall, D. and Tennenhouse, D. (2019). Retrospective on"towards an active network architecture". *ACM SIGCOMM Computer Communication Review*, 49(5):86–89.

[Wickboldt et al. 2015] Wickboldt, J., Jesus, W. D., Isolani, P., Both, C., Rochol, J., and Granville, L. Z. (2015). Software-Defined Networking: Management Requirements and Challenges. *IEEE Communications Magazine*, 53(1):278–285.

[Yu et al. 2020] Yu, Z., Zhang, Y., Braverman, V., Chowdhury, M., and Jin, X. (2020). NetLock: Fast, Centralized Lock Management Using Programmable Switches. In *SIGCOMM*.

[Zekauskas et al. 2006] Zekauskas, M. J., Karp, A., Shalunov, S., Boote, J. W., and Teitelbaum, B. R. (2006). A One-way Active Measurement Protocol (OWAMP). RFC 4656.

[Zhu et al. 2015] Zhu, Y., Kang, N., Cao, J., Greenberg, A., Lu, G., Mahajan, R., Maltz, D., Yuan, L., Zhang, M., Zhao, B. Y., and Zheng, H. (2015). Packet-level telemetry in large datacenter networks. *ACM SIGCOMM Computer Communication Review (CCR)*, 45(4):479–491.

Capítulo 2

Processamento de Imagens Omnidirecionais e Aplicações

Thiago L. T. da Silveira e Cláudio R. Jung

Programa de Pós-Graduação em Computação (PPGC) - UFRGS

Abstract

Omnidirectional images and videos have been widely disseminated due to the popularization of devices for capture and visualization. Unlike images captured with perspective projection, omnidirectional media are defined on the surface of a sphere, having a field of view of $360^\circ \times 180^\circ$. *Thus, they store the light intensities in the entire region around the capture point, with high potential for use in applications involving immersive augmented, mixed and virtual reality experiences. Although defined in the spherical domain, omnidirectional images are often mapped to a (multi)planar representation, which results in distorted images and degrades the performance of most traditional visual computing algorithms designed to work in the plane. This chapter reviews the spherical camera model, the most common capture devices, and popular (multi)planar representations of omnidirectional media. It also lists the main challenges of omnidirectional visual computing, focusing on the deep learning paradigm, and discusses potential applications.*

Resumo

Imagens e vídeos omnidirecionais têm sido amplamente difundidos devido à popularização de dispositivos para captura e visualização. Ao contrário das imagens capturadas com projeção em perspectiva, as mídias omnidirecionais são definidas sobre a superfície de uma esfera, tendo um campo de visão de $360^\circ \times 180^\circ$. *Assim, elas armazenam as intensidades de luz em toda região em torno do ponto de captura, com alto potencial de uso em aplicações que envolvem experiências imersivas de realidade aumentada, mista e virtual. Embora definidas no domínio esférico, as imagens omnidirecionais muitas vezes são mapeadas para uma representação (multi) planar, o que resulta em imagens distorcidas*

Vídeo com a apresentação do capítulo: `https://youtu.be/rqLWrSRm-Y0`

e degrada o desempenho da maioria dos algoritmos tradicionais de computação visual projetados para funcionar no plano. Este capítulo revisa o modelo de câmera esférica, os dispositivos de captura mais comuns e as representações (multi) planares populares de mídias omnidirecionais. Ele também elenca os principais desafios da computação visual omnidirecional, com foco no paradigma de aprendizado profundo, e aborda potenciais aplicações.

2.1. Introdução

Imagens omnidirecionais – também conhecidas como imagens esféricas, panorâmicas ou em 360° – são populares nos dias de hoje graças à acessibilidade e portabilidade dos dispositivos de captura lançados nos últimos anos [J. Huang et al. 2017, da Silveira and Jung 2019b]. Imagens e vídeos em 360° aproximam-se do modelo de imagem ideal chamado modelo de imagem plenóptica, onde toda a informação visual da cena é capturada a partir de todos os pontos de vista possíveis ao longo do tempo [Ebrahimi et al. 2016]. Além das aplicações clássicas, as mídias esféricas ajudam a proporcionar experiências imersivas ao usuário em novas aplicações de realidade aumentada, mista e virtual (AR/MR/VR) quando visualizadas em dispositivos de visualização montados na cabeça (HMDs) [Serrano et al. 2019]. Em particular, a edição de panoramas permite a manipulação de imagens e vídeos, o que pode melhorar a experiência do usuário [Zhang et al. 2022b, Zhang et al. 2021]. Ao contrário das imagens regulares baseadas no modelo de projeção em perspectiva, definidas em um plano, as imagens omnidirecionais são definidas na superfície da esfera unitária [Li 2008, Fujiki et al. 2007]. As imagens esféricas têm um campo de visão (FoV, ou *Field-of-View*) de $360° \times 180°$ [da Silveira and Jung 2019b] que captura as intensidades de luz de toda a cena.

Para exemplificar as diferenças visuais entre imagens em perspectiva e panorâmicas, considere a Figura 2.1. Mais precisamente, a Figura 2.1(a) ilustra uma captura usando uma câmera em perspectiva com FoV limitado, enquanto a Figura 2.1(a) mostra uma imagem esférica capturada do mesmo ponto de vista[1]. Percebe-se claramente que a topologia de ambas imagens são bastante distintas.

Embora as imagens panorâmicas sejam definidas no domínio esférico, elas são comumente representadas em formato planar ou multi-planar [Yang et al. 2018, Zelnik-Manor et al. 2005, da Silveira et al. 2022]. Muitas funções de mapeamento da esfera para um ou mais planos podem ser usadas para gerar a representação planar, mas todas elas introduzem distorções [Su and Grauman 2017, Azevedo et al. 2020]. Um panorama pode ser representado no plano (como em formato de "mapa-múndi"), mas o algoritmo de computação visual que o utiliza como entrada ainda precisa considerar as deformações introduzidas para ser preciso em sua tarefa [Cruz-Mota et al. 2012, da Silveira et al. 2021].

Comparado com a evolução dos algoritmos projetados para imagens em perspectiva, a computação visual *omnidirecional* ainda está em estágio embrionário, e apenas alguns problemas clássicos são abordados sob esta ótica renovada. Este artigo lança luz sobre como se pode esperar que a computação visual omnidirecional difira da tradicional

[1]As imagens foram geradas artificalmente, a partir do modelo *Classroom* 3D, disponível sob licença CC0 license em `https://www.blender.org`.

(a) (b)

Figura 2.1: Duas capturas da mesma cena 3D com poses idênticas, mas com câmeras diferentes. A primeira captura foi feita por uma (a) câmera em perspectiva de FoV estreito, e a segunda (b) veio de uma câmera 360°, ilustrada no plano por projeção ortográfica.

e quais esforços podem ser empregados para lidar com essas discrepâncias. Ele traz uma discussão aprofundada sobre o modelo de câmera esférica, *pipelines* padrão de aquisição de imagens em 360° e representações planares ou multi-planares. Além disso, o artigo ataca deficiências do uso de redes convolucionais em panoramas, e apresenta alternativas recentes.

2.2. Uma visão geral sobre imagens 360°

Esta seção abrange aspectos técnicos envolvendo a base matemática do imageamento omnidirecional, sistemas comuns para aquisição de imagem 360° e representações planares padrão de panoramas. Mais detalhes sobre criação de conteúdo omnidirecional podem ser encontrados em pesquisas como [Wang et al. 2020b].

2.2.1. O modelo de câmera esférica

Uma câmera *pinhole* é modelada por projeções centrais e em perspectiva, onde um raio vem de um ponto tridimensional (3D) em coordenadas de mundo, passa por seu centro de projeção e toca o plano da imagem [Hartley and Zisserman 2003]. As particularidades do mapeamento 3D–2D subjacente – como a cobertura da cena na imagem resultante – dependem da matriz da câmera, que combina parâmetros intrínsecos e extrínsecos [Hartley and Zisserman 2003].

Por outro lado, o modelo de câmera esférica deriva das projeções central e esférica [S. Li and Fukumori 2005]. Abstrai-se a câmera como uma *esfera unitária* [da Silveira and Jung 2019b] localizada e orientada no espaço. Uma vez que uma câmera omnidirecional cobre todo o FoV, todos os pontos do mundo 3D ao redor da câmera são capturados em uma única projeção esférica [Akihiko et al. 2005], com exceção de regiões com oclusões. O modelo de câmera esférica não considera parâmetros intrínsecos e assume que a câmera é totalmente representada por seus seis graus de liberdade (6 DoF, ou *Degrees of Freedom*) extrínsecos [Guan and Smith 2017, Krolla et al. 2014].

Primeiro precisamos definir um sistema de coordenadas de mundo (3D) para entender o modelo de câmera esférica. Dado esse sistema de coordenadas, podemos centralizar a câmera em uma dada posição $\mathbf{C} \in \mathbb{R}^3$ e orientá-la usando uma dada matriz de

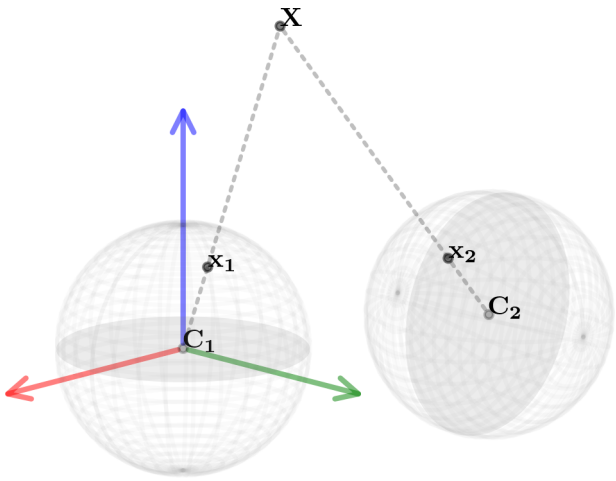


Figura 2.2: Projeção de um ponto 3D $\mathbf{X}$ em duas câmeras esféricas com diferentes parâmetros. As câmeras são descritas pelos extrínsecos$[\mathbf{R}_1 = \mathbf{I}|\mathbf{t}_1 = -\mathbf{R}_1\mathbf{C}_1 = \mathbf{0}]$ e $[\mathbf{R}_2 \neq \mathbf{I}|\mathbf{t}_2 = -\mathbf{R}_2\mathbf{C}_2 \neq \mathbf{0}]$.

rotação $\mathbf{R} \in SO(3)$. Assim, podemos caracterizar a câmera através de seus parâmetros extrínsecos $[\mathbf{R}|\mathbf{t}]$, onde $\mathbf{t} = -\mathbf{RC} \in \mathbb{R}^3$ é chamado de "vetor de translação" [da Silveira et al. 2022].

Um ponto 3D $\mathbf{X} \in \mathbb{R}^3$ em coordenadas de mundo, parametrizado de acordo com o sistema de coordenadas definido, é então projetado na câmera definida por $[\mathbf{R}|\mathbf{t}]$ usando [Akihiko et al. 2005]

$$\mathbf{x} = \frac{\mathbf{RX} + \mathbf{t}}{\|\mathbf{RX} + \mathbf{t}\|_2}, \tag{1}$$

onde $\|\cdot\|_2$ é a norma ℓ_2. Observa-se que o ponto $\mathbf{x}$ resultante da imagem está na superfície de uma esfera unitária, *i.e.,* $\mathbf{x} \in S^2 \subset \mathbb{R}^3$ [S. Li and Fukumori 2005].

A Figura 2.2 ilustra um ponto tridimensional $\mathbf{X}$ do mundo projetado em duas câmeras esféricas distintas. Uma das câmeras está posicionada na origem e alinhada ao sistema de coordenadas pré-definido, tendo extrínsecos $[\mathbf{R}_1 = \mathbf{I}|\mathbf{t}_1 = -\mathbf{R}_1\mathbf{C}_1 = \mathbf{0}]$. A outra câmera não está na origem e tem uma orientação diferente, com extrínsecos $[\mathbf{R}_2 \neq \mathbf{I}|\mathbf{t}_2 = -\mathbf{R}_2\mathbf{C}_2 \neq \mathbf{0}]$. Note que os pontos de imagem $\mathbf{x}_1$ e $\mathbf{x}_2$ são descritos em coordenadas locais da imagem, ou seja, em relação a cada câmera. Sozinhos, eles não codificam informações explícitas sobre as posições originais das câmeras no sistema de coordenadas pré-definido.

2.2.2. Aquisição de imagens esféricas

As estratégias existentes para aquisição de imagens e vídeos omnidirecionais envolvem o uso de uma ou mais câmeras regulares, potencialmente equipadas com componentes ópticos especiais [da Silveira et al. 2022]. Na verdade, ao contrário do que sugere o modelo de imagem esférica, não há um dispositivo de sensor único para capturar todas as informações da cena de uma só vez [Adarve and Mahony 2017]. Abaixo, é apresentada uma breve revisão dos três principais sistemas de captura: catadióptrico, polidióptrico e dispositivos de imagem de 360° com duas lentes olho de peixe.

(a) (b) (c)

Figura 2.3: Versões planificadas da imagem esférica ilustrada na Figura 2.1b: (a) Formato ERP, (b) CMP and (c) representação icosahedral.

Sistemas de imagem catadióptricos combinam uma câmera regular com um espelho de formato convexo (cônico, esférico, parabólico ou hiperbólico) e permitem capturar informações visuais de todo o campo de visão *horizontal* de uma cena [Nayar 1997]. Esse método sofre de auto-oclusão do sensor/espelho e normalmente gera imagens representadas em formato cilíndrico [Cruz-Mota et al. 2012]. Devido ao campo de visão vertical restrito e aos componentes frágeis do espelho, os dispositivos catadióptricos são raros em pesquisas e aplicações industriais recentes [Aggarwal et al. 2016].

Por outro lado, dispositivos de captura polidióptricos consistem em um número (variável) de câmeras regulares apontando para fora em uma estrutura (*rig*). Cada câmera captura uma parte estreita da cena (ou seja, tem um campo de visão estreito), e todas as visualizações são combinadas em um procedimento baseado em *software* chamado costura de imagens (*stitching* ou *mosaicking*) [Im et al. 2016]. Dispositivos de imageamento polidióptricos costumam ser volumosos e caros, mas podem produzir panoramas de alta resolução com campo de visão personalizado [Fangi et al. 2018].

Um tipo mais recente de dispositivo de captura, suportado por muitos fabricantes, combina dois sensores localizados em posições opostas equipados com lentes olho de peixe [Shan and Li 2018]. Cada sensor captura uma imagem "hemisférica" (com campo de visão ultra-grande) adequada para a costura esférica de duas visualizações [Lo et al. 2018]. Esses dispositivos portáteis e baratos simplificaram e democratizaram a aquisição de conteúdo real em 360° e impulsionaram a indústria e a pesquisa em áreas relacionadas à realidade aumentada/mista/virtual [Jung et al. 2019, da Silveira et al. 2022].

2.2.3. Representação de mídia esférica

Imagens e vídeos esféricos podem ser representados como um mapeamento na esfera unitária [Akihiko et al. 2005]. A imagem digital subjacente é obtida por meio de um procedimento de amostragem, que é normalmente realizado após a aplicação de uma ou mais funções de mapeamento de esfera para plano.

Uma função popular de mapeamento de esfera para plano é a chamada projeção equirretangular (ERP, abreviada de *EquiRectangular Projection*). A ERP - também conhecida como mapeamento latitude-longitude [Gava et al. 2018] - é considerada a representação planar padrão da esfera [Eder et al. 2019, da Silveira and Jung 2019b], e permite um relação simples entre *pixels* no plano e pontos amostrados na esfera.

Uma vez que um determinado ponto **x** está sobre a superfície de uma esfera unitária, ele pode ser reescrito em coordenadas esféricas usando dois parâmetros angulares (θ, ϕ) [Akihiko et al. 2005]:

$$\mathbf{x} = [x\ y\ z]^\top = [\cos(\theta)\sin(\phi)\ \sin(\theta)\sin(\phi)\ \cos(\phi)]^\top, \tag{2}$$

onde $\theta \in [0, 2\pi)$ e $\phi \in [0, \pi)$.

Além disso, cabe lembrar que uma câmera omnidirecional captura todo o conteúdo da cena e, portanto, há informações associadas a cada posição (θ, ϕ) na superfície esférica. Como tal, uma imagem omnidirecional pode ser representada em um plano de $[0, 2\pi) \times [0, \pi)$. A *imagem ERP* é, então, gerada a partir de uma discretização em (θ, ϕ), de modo que a intensidade de luz associada a um ponto de imagem **x** mapeia para a posição do pixel **p** dada por

$$\mathbf{p} = [u\ v]^\top = \left[\left\lfloor \frac{\theta w}{2\pi} \right\rfloor \ \left\lfloor \frac{\phi h}{\pi} \right\rfloor \right]^\top, \tag{3}$$

onde w e h são a largura e altura da imagem ERP em *pixels*, respectivamente. As imagens ERP frequentemente têm uma razão de aspecto 2:1, o que significa que a variação angular em θ e ϕ é a mesma.

Os parâmetros θ e ϕ são recuperados de **x** usando a relação inversa apresentada na Eq. (2):

$$\theta = \tan^{-1}(y, x) \tag{4}$$

e

$$\phi = \cos^{-1}(z), \tag{5}$$

onde $\tan^{-1}(\cdot, \cdot)$ representa a função arco-tangente sensível a quadrante.

A ERP é direta e simples de calcular, mas possui uma amostragem não-uniforme que distorce os objetos da cena dependendo de sua localização na imagem [Cruz-Mota et al. 2012], que se torna mais intensa próximo aos polos norte e sul [Ferreira et al. 2017]. Muitos outros mapeamentos de esfera para plano podem ser considerados, mas nenhum é livre de distorção [Zelnik-Manor et al. 2005, Su and Grauman 2017]. Como essas deformações dependem da magnitude do FoV usado na projeção [da Silveira et al. 2018], alguns autores propõem mapear a esfera em não apenas um, mas vários planos. Por exemplo, mapear a esfera em um cubo circunscrito resulta em seis imagens com FoV mais estreito e faces equi-angulares, conhecidas como mapeamento em cubo (CMP, de *Cube Map Projection*) [Dai et al. 2019, da Silveira et al. 2018]. A CMP reduz as distorções, mas o FoV de 90° de cada face ainda é maior do que o comumente encontrado em imagens de perspectiva [Su and Grauman 2017, Wang et al. 2018a]. Além disso, a conectividade das faces deve ser considerada ao processar *imagens CMP*. Representações emergentes baseadas em divisões sucessivas de uma forma geométrica 3D tentam mitigar ainda mais as distorções. Abordagens proeminentes incluem a icosfera/projeção em planos tangentes [Eder et al. 2020], que deriva de um icosaedro, e aquelas baseadas em um octaedro [Lee et al. 2020].

A Figura 2.3 ilustra o mapeamento da imagem omnidirecional da Figura 2.1b para suas representações ERP, mapa de cubo e baseadas em icosaedro (desdobrados). Deve-se mencionar que a troca de formatos de representação pode levar à perda de informações e introduzir artefatos [Azevedo et al. 2020], uma vez que esses mapeamentos requerem transformações *subpixel* [Coors et al. 2018].

Como uma nota final sobre a representação de imagens, artefatos de compressão também estão presentes em conteúdos esféricos armazenados digitalmente e podem gerar degradações visuais adicionais. O leitor é encaminhado ao [Xu et al. 2020] para uma análise detalhada sobre compressão de imagens esféricas e avaliação de qualidade.

2.3. Imagens esféricas a aprendizado profundo

Seguindo a tendência de aplicações que exploram imagens em perspectiva, técnicas de aprendizado profundo também têm sido exploradas para processar mídias esféricas. Entretanto, a amostragem não-uniforme gerada pelo processo de planificação exige cuidado quando se deseja aproveitar arquiteturas planejadas para imagens em perspectiva.

Este capítulo foca a análise no formato ERP por ser a representação planar padrão de imagens em 360°, amplamente empregada na indústria e na pesquisa [Su and Grauman 2017], mas considerações serão feitas sobre outras representações. É importante observar que, embora tais representações possam ser obtidas diretamente por amostragem de um domínio esférico, a maioria dos panoramas é armazenada no formato ERP. Assim, os mapeamentos multi-planares envolvem a interpolação do ERP para a esfera e, em seguida, a amostragem para a representação desejada.

2.3.1. Desafios com a representação ERP

Como discutido na Seção 2.2.3, a ERP amostra a esfera unitária de maneira *não-uniforme*. Esse procedimento resulta em um "efeito de alongamento" que se acentua nas proximidades dos polos. De fato, os polos são superamostrados, pois as primeiras e últimas linhas da imagem colapsam nos polos norte e sul da esfera, respectivamente (como nos paralelos com latitudes muito altas nos dois polos do globo terrestre). Portanto, essas linhas replicam as informações em todas as colunas. Em geral, o espaçamento entre pontos adjacentes ao longo de uma linha no formato ERP é proporcional a $\sin\phi$ [De Simone et al. 2017], resultando em um desequilíbrio acentuado entre a linha do equador ($\phi = \frac{\pi}{2}$) e os polos ($\phi = 0$ e $\phi = \pi$).

Além das distorções induzidas pela amostragem não-uniforme, as imagens ERP têm uma propriedade *cíclica* (ou *circular*) [da Silveira et al. 2022, Lee et al. 2020], o que significa que as bordas esquerda e direita se conectam. Portanto, os objetos podem ser divididos nas porções esquerda e direita de uma imagem ERP. A Figura 2.3a apresenta uma ilustração das questões mencionadas acima.

O amplo uso de imagens ERP decorre de sua simplicidade e porque elas contêm todas as informações da cena em um único plano. Explorar todo o contexto da cena a partir de uma única imagem com domínio retangular é muito atraente, especialmente na era do aprendizado profundo e o uso generalizado de redes neurais convolucionais (CNNs, de *Convolutional Neural Networks*) [Goodfellow et al. 2016]. A ideia central de uma

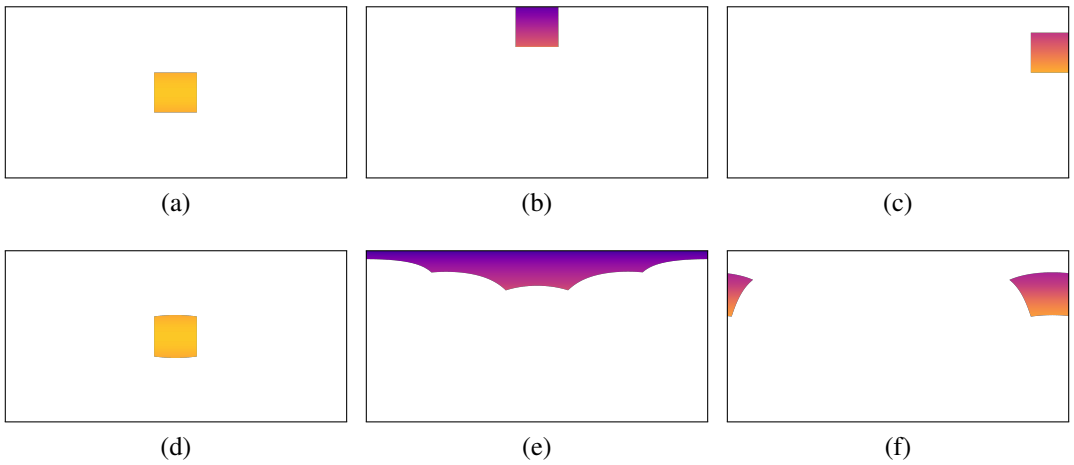


Figura 2.4: Suporte de um *kernel* (amplificado para fins de visualização) posicionado em posições diferentes de uma imagem ERP: (a)–(c) *kernels* retangulares, e (d)–(f) *kernels* ideais.

camada convolucional é que ela contém filtros com suporte espacialmente invariante – seu campo receptivo – e pesos [Goodfellow et al. 2016]. Os *kernels* convolucionais padrão são retangulares ou, mais comumente, quadrados, e são aplicados em toda a imagem usando um mecanismo de janela deslizante [Goodfellow et al. 2016]. Devido às distorções relacionadas à amostragem, aplicar esses filtros regulares a uma imagem ERP faz com que regiões desiguais da superfície da esfera sejam cobertas, dependendo da posição do filtro [Fernandez-Labrador et al. 2020]. Se quisermos que o suporte do *kernel* cubra a mesma área na superfície da esfera, ele deve ser ajustado dependendo da latitude ϕ da imagem em que está centrado [Su and Grauman 2017].

A Figura 2.4 ilustra a forma do suporte de um *kernel* regular e de um kernel "ideal" (maior do que o usual para fins de visualização) em diferentes partes de uma imagem ERP. A área relativa da superfície da esfera relacionada a cada pixel coberto pelo *kernel* é mostrada em um mapa de cores: cores arroxeadas representam valores pequenos e cores amareladas representam valores maiores (independentemente da resolução da imagem ERP, eles somam 4π, que é a área da superfície da esfera). Quando o centro do filtro ideal está na região do equador ($\phi = \frac{\pi}{2}$), como mostrado na Fig2.4d, seu suporte é pouco distorcido (compare com a Fig 2.4a, por exemplo). As distorções ficam aparentes à medida que movemos o filtro em direção aos polos. A Figura 2.4e ilustra o ajuste necessário para que o filtro cubra a mesma área da superfície da esfera que na Fig2.4d, quando se aproxima do polo norte ($\phi \to 0$) da imagem. Como podemos observar, a forma do filtro já não é mais retangular, com suporte mais amplo próximo ao polo (primeira linha). Por outro lado, a Figura 2.4b mostra o efeito de um kernel convolucional padrão, que é fixo e cobre uma região menor da superfície da esfera em comparação com a Fig2.4a. Note que a área em destaque na Figura 2.4b atinge o mesmo número de *pixels* que aquela na Figura 2.4a, mas a área coberta na superfície da esfera é menor. Finalmente, a Figura 2.4c retrata o que acontece quando um filtro regular toca uma borda lateral da imagem. O filtro é frequentemente aplicado após preenchimento com zeros ou extrapolação de dados em uma convolução regular. No caso esférico, um filtro ideal realiza uma convolução circular,

como mostrado na Figura 2.4f. A área da superfície da esfera coberta pelo *kernel* ideal em diferentes posições, ilustradas nas Figuras 2.4d, 2.4e e 2.4f, é fixa.

Mitigar o problema de circularidade em imagens ERP é simples. Isso pode ser implementado através de uma estratégia de preenchimento circular horizontal, que mantém a continuidade das informações espaciais ao longo dos paralelos da esfera nas bordas horizontais. Esses preenchimentos podem ser usados como etapa de pré-processamento, onde o panorama de entrada é preenchido circularmente antes de usar a CNN e, em seguida, recortado de volta ao tamanho original, conforme usado para o cálculo de fluxo óptico em [da Silveira and Jung 2019a]. Alternativamente, pode ser incorporado a uma camada convolucional circular, observando que o tamanho do preenchimento deve ser ajustado com base nas dimensões do *kernel* [Sun et al. 2019, Zioulis et al. 2021, Wang et al. 2018b, Zhuang et al. 2022].

A deformação espacial do *kernel* é mais difícil de lidar. Alguns trabalhos propõem ajustar as convoluções (e às vezes as operações de *pooling*) para lidar com as distorções induzidas pelas mapeamentos da esfera para o plano [Zioulis et al. 2018, Tateno et al. 2018, Su and Grauman 2017, Fernandez-Labrador et al. 2020]. Por exemplo, Su e Grauman [Su and Grauman 2017] propõem aprender pesos que ajustam as respostas de um filtro regular para acomodar as distorções do ERP. Convoluções sensíveis à distorção, propostas por Tateno *et al.* [Tateno et al. 2018], deformam seus campos receptivos para amostrar pontos dentro do suporte ideal (conforme discutido anteriormente). Uma ideia semelhante é explorada em [Fernandez-Labrador et al. 2020], onde convoluções deformáveis são usadas para a amostragem induzida pelo ERP. Uma desvantagem dessas abordagens é a sobrecarga computacional e a complexidade do código, uma vez que o uso de *kernels* que amostram irregularmente tende a ser mais lento do que convoluções 2D regulares que exploram diretamente o paralelismo em placas gráficas (GPUs).

Convoluções dilatadas foram introduzidas no contexto do aprendizado profundo para imagens planares por Fisher e Koltun [Yu and Koltun 2015], e são uma solução atraente para lidar com a amostragem não-uniforme de imagens ERP. De fato, elas foram exploradas por Zioulis e colegas [Zioulis et al. 2018] para ajustar o campo receptivo horizontal da convolução, dependendo da latitude, sendo maior próximo aos polos e menor próximo à linha do equador. As redes de convoluções dilatadas combinadas de forma adaptativa (ACDNets), propostas recentemente por Zhuang e colegas [Zhuang et al. 2022], consistem em aplicar um conjunto de convoluções dilatadas paralelas combinadas de forma adaptativa em relação aos canais através de pesos aprendíveis. Soluções baseadas em convoluções dilatadas são mais rápidas do que os *kernels* deformáveis. No entanto, eles não podem cobrir regiões exatamente de área igual na esfera, uma vez que cada linha requer uma deformação separada, como mostrado na Figura 2.4.

Outra estratégia atraente para mitigar a amostragem não-uniforme de imagens ERP é usar operadores não-locais. Ao contrário das CNNs, que apresentam um campo receptivo local com base no tamanho do *kernel*, os operadores não-locais potencialmente exploram todas as características espaciais de uma determinada camada. O representante mais conhecido de operador não local é o *Transformer*, inicialmente introduzido para processamento de texto [Vaswani et al. 2017] e posteriormente estendido para imagens. A extensão mais popular para imagens planares é o *Visual Transformer* (ViT) [Dosovitskiy

et al. 2020], que explora pequenas regiões (*patches*) não sobrepostas de uma imagem como *tokens* no *framework* original dos *Transformers*. No contexto de panoramas, Sun e colegas[Sun et al. 2021] propuseram uma arquitetura que gera primeiro características latentes na direção vertical e depois explora um *Transformer multi-head* na direção horizontal. O *Panorama Transformer* (Panoformer) [Shen et al. 2022] trabalha diretamente na representação ERP e usa planos tangentes para extrair os *tokens* baseados em *patches*. Ele também explora um *embedding* posicional relativo baseada em projeções ERP-esfera-ERP para levar em conta a localização espacial dos *tokens*. O *Parallel Convolutional Transformer* (PCFormer)[Xu et al. 2022] explora um ramo convolucional para extrair características locais e um ramo semelhante ao ViT para extrair interações de longo prazo, e, em seguida, explora um módulo de fusão de atenção dupla para fusão de características em múltiplas escalas. O modelo Trans4PASS [Zhang et al. 2022a] explora uma rede de pirâmide de características e introduz deslocamentos relativos dependentes dos dados usados para incorporar um agrupamento de *patches* deformáveis, com o objetivo de mitigar a amostragem não-uniforme da ERP. Em teoria, o suporte não local das abordagens baseadas em *Transformers* pode mitigar os problemas de circularidade e amostragem não-uniforme em representações ERP. No entanto, arquiteturas baseadas exclusivamente em *Transformers* geralmente requerem conjuntos de dados maiores e mais recursos computacionais. Dai e colegas [Dai et al. 2021] mostraram que a combinação de camadas convolucionais e *Transformers* pode ser usada para "combinar" conjuntos de dados de diferentes tamanhos para imagens de perspectiva, e o mesmo pode ser verdadeiro para panoramas.

2.3.2. Desafios com representações multi-plano e híbridas

A principal causa de distorção na representação ERP é o mapeamento de toda a esfera em um único plano. Outros mapeamentos de esfera para plano, como CMP ou baseados em icosaedro, aliviam as distorções, pois extraem projeções em planos tangentes com FoVs mais estreitos. No entanto, há um compromisso entre o FoV da imagem e a informação contextual sendo representada [da Silveira et al. 2018]: por um lado, representações multi-plano aliviam os problemas de distorção usando FoVs menores; por outro lado, cada plano contém apenas informações parciais sobre o conteúdo completo. Além disso, o problema de circularidade horizontal presente em formatos ERP é potencializado em representações multi-plano, uma vez que o mesmo conteúdo esférico pode se espalhar pelas bordas das projeções planares adjacentes. O leitor pode voltar às Figuras 2.3b e2.3c e perceber o quão intrincadas são as conexões das faces em representações esféricas multi-plano. Lidar com imagens multi-plano requer tratamento adequado, como preenchimento de faces [Wang et al. 2020a, Eder et al. 2020] ou costura [da Silveira et al. 2018, Rey-Area et al. 2022] para mitigar problemas de descontinuidade quando os planos são processados independentemente.

Outra estratégia para lidar com as bordas da representação multi-plano envolve ajustar o operador convolucional para o domínio desejado. Por exemplo, Lee *et al.* [Lee et al. 2019, Lee et al. 2020] geram uma representação icosaédrica da esfera, chamada SpherePHD, e definem *kernels* convolucionais e operadores de *pooling* que trabalham nos triângulos da tesselação que já levam em conta a conectividade das bordas. O uso conjunto de convoluções e operadores de *pooling* adaptados também expande o campo

Tabela 2.1: Análise de representações esféricas comuns para aprendizado profundo: prós, contras e estratégias para mitigação de problemas

Representação	Prós	Contras	Mitigação
ERP	Conteúdo completo (informação global) no mesmo plano	Deformações relacionadas à amostragem	Convoluções deformáveis; abordagens baseadas em *Transformers*
	Única imagem	Circularidade horizontal	Preenchimento horizontal adaptativo; convolução horizontal
Multi-plano	Menores distorções por plano (menor conforme o número de planos aumenta)	Informação local por plano (menos conteúdo conforme o número de planos aumenta)	Processamento conjunto dos planos; convoluções e *pooling* adaptados à representação
	Uso potencial de CNNs para cada plano	Conexão de bordas inter-plano	Preenchimento de bordas; pós-processamento baseado em costura; *embeddings* posicionais em *Transformers*
Híbridos	Potencial aprendizagem do melhor de cada representação	Modelos maiores e mais complexos	Destilação de conhecimento

receptivo dos *kernels* convolucionais em camadas mais profundas, o que ajuda a propagar as informações de um plano para os vizinhos. No entanto, essas abordagens geralmente não permitem transferir os pesos da rede treinada em imagens de perspectiva para o domínio esférico.

Como discutido para o formato ERP, *Transformers* também têm sido adotados para lidar com representações multi-plano. O *Cube-map Vision Transformer* (CViT) [Bai et al. 2022] visa aprender implicitamente as conexões das faces do CMP usando uma abordagem baseada em ViT. O CViT extrai *patches* planares das seis faces da CMP e usa incorporações posicionais aprendidas para manter implicitamente as informações espaciais dos *patches* e as conexões entre as faces. Li e colegas [Li et al. 2023] exploram uma amostragem *Hierarchical Equal Area isoLatitude Pixelization* (HEALPix), que é baseada em quadriláteros curvilíneos em vez de projeções planares, para gerar *tokens*. Em seguida, eles exploram uma incorporação posicional feita manualmente com base nas coordenadas esféricas do *patch* (vetor unitário na superfície da esfera) para lidar com o problema de conectividade das bordas.

Alguns autores propõem o uso de mais de um esquema de projeção para lidar melhor com a amostragem não-uniforme de imagens ERP. A rede *BiFuse* [Wang et al. 2020a] explora representações ERP e CMP através de um codificador-decodificador profundo em paralelo, com interconexões e um módulo de fusão de duas projeções. Li *et al.* [Li et al. 2022] propõem uma combinação de representações ERP, tri-cilíndrica e modificada de CMP (por meio do preenchimento das faces do cubo) no contexto de cálculo de fluxo óptico e usam um esquema de fusão profunda baseado em U-Net. O uso de várias projeções parece uma direção interessante para tirar o melhor proveito de cada uma delas. No entanto, o uso de várias representações e múltiplos ramos também aumenta o tamanho e a complexidade do modelo, o que tende a aumentar os requisitos de memória da GPU. Uma solução potencial para esse problema é o uso de técnicas de destilação de conhecimento [Gou et al. 2021], que transferem um modelo professor maior para um modelo aluno menor.

Em resumo, os desafios para explorar abordagens de aprendizado profundo dependem da representação escolhida. A Tabela 2.1 lista os prós e contras das representações ERP, multi-plano, híbridas e estratégias de mitigação representativas.

2.4. Algumas Aplicações

Embora praticamente qualquer aplicação possa se beneficiar do uso de imagens esféricas, essa seção foca em algumas aplicações que naturalmente requerem um FoV mais amplo.

2.4.1. Correção de orientação

Embora uma imagem esférica capture todo ambiente ao redor da câmera, a visualização é normalmente feita em dispositivos convencionais (como celulares e monitores), que possuem FoV limitado. Assim, é necessário recortar uma porção da imagem esférica e realizar uma projeção planar antes da visualização [da Silveira and Jung 2023]. Uma etapa de pré-processamento usual consiste em rotacionar o conteúdo da imagem esférica para que o horizonte da imagem se alinhe com o horizonte no mundo, obtendo-se uma orientação canônica de captura. Tal processo de correção de orientação é normalmente

(a) (b)

Figura 2.5: A (a) tilted 360° capture (in ERP format) of the same scene as in Figura 2.1b and (b) its upright aligned version using the estimates from the method in [Bergmann et al. 2021].

chamado de *gravity alignment*, *horizon alignment* ou *upright adjustment*.

O objetivo de uma abordagem correção de orientação é estimar uma matriz de rotação $\mathbf{R}^{\uparrow} \in SO(3)$ que alinha o plano do solo com o equador e posiciona os objetos da cena verticalmente. A correção da orientação de uma imagem ERP, por exemplo, é tipicamente realizada pelos seguintes passos. Primeiro, precisamos projetar a imagem (suas intensidades de luz) na esfera unitária usando a Eq.(2). Em seguida, temos que girar a esfera (ou seja, todos os pontos em sua superfície individualmente) usando $\mathbf{R}^{\uparrow\top}$. Por fim, precisamos projetar as intensidades de luz associadas a esses pontos de volta ao plano usando a Eq.(3). Vale ressaltar que a imagem de entrada, com orientação arbitrária, sofrerá diferentes operações de reamostragem para ser mapeada na imagem corrigida verticalmente de saída. Por exemplo, as informações nos polos da imagem de entrada podem ser reduzidas para se ajustarem às latitudes centrais da imagem de saída. *Pixels* originalmente no equador também podem ser mapeados para os polos da imagem de saída, perdendo grande parte das informações de alta frequência [Murrugarra-Llerena et al. 2022]. Também cabe notar que a rotação desejada envolve dois graus de liberdade, visto que qualquer rotação em torno do eixo vertical para uma imagem já alinhada segue gerando uma imagem alinhada. O terceiro grau de liberdade pode ser ajustado para colocar conteúdo de interesse à frente do panorama. Isso pode ser feito manualmente, onde o usuário ajusta a rotação de acordo com seus interesses pessoais, ou até de modo automático, através do uso de técnicas que identificam regiões "interessantes" na imagem esférica usando técnicas de saliência visual [Bernal-Berdun et al. 2022].

A tendência atual para abordar o problema de correção de orientação é usar técnicas de aprendizado de máquina, tipicamente usando redes profundas, para estimar o vetor de orientação (*upright vector*) da imagem de entrada. Com isso, se pode gerar a matriz de rotação e gerar uma versão canônica da imagem de entrada. Para tal, são necessárias bases de dados contendo imagens na orientação canônica, a partir das quais se pode gerar rotações arbitrárias para construir os dados de treinamento anotados: a imagem rotacionada é o dado de entrada, e o *upright vector* é o dado que a rede precisa inferir. Por exemplo, a base SUN360 [J. Xiao et al. 2012] forece cerca de 57,000 panoramas em ambientes internos e externos, e pode ser usada nessa tarefa.

Outro ponto importante se refere à avaliação dos resultados de correção de orientação. Como discutido em Jung *et al.* [Jung et al. 2019], discrepâncias angulares menores

que 5° são consideradas muito satisfatórias pelos seres humanos, enquanto aquelas menores que 12° são consideradas satisfatórias. Por exemplo, considere a imagem ERP mostrada na Figura 2.5a, que corresponde a uma captura inclinada do ambiente ilustrado na Figura 2.3a. Como se pode perceber, a imagem apresenta fortes distorções visuais, sendo inclusive difícil a identificação dos elementos da cena (cadeiras, quadro, teto, etc.). Aplicando o algoritmo de correção de orientação proposto em [Bergmann et al. 2021] gera a imagem mostrada na Figura 2.5b, que visualmente alinha o plano horizontal do mundo (sala de aula) com o equador da imagem esférica.

2.4.2. Reconstrução 3D

A reconstrução tridimensional de cenas desempenha um papel fundamental em uma ampla gama de aplicações, abrangendo campos que vão desde a realidade virtual/aumentada até a robótica, a análise de cenas forenses e a indústria do entretenimento [da Silveira and Jung 2023]. A habilidade de transformar uma cena bidimensional capturada em uma representação 3D precisa e detalhada é essencial para compreender e interagir com o ambiente de maneira mais rica e imersiva. Além disso, a reconstrução 3D é valiosa na preservação do patrimônio cultural, permitindo a digitalização precisa de artefatos históricos e locais arquitetônicos, o que facilita a documentação, a restauração e a disseminação do conhecimento. Na visão computacional tradicional, usando imagens em perspectiva, são necessárias diversas capturas com diferentes pontos de vista para que se tenha uma cobertura completa do ambiente. Por outro lado, uma única captura esférica já fornece toda informação visual em torno da câmera.

Uma imagem esférica armazena a informação visual (cor) ao longo dos raios emitidos em torno da câmera. Para gerar a reconstrução 3D da cena, é necessário estimar a distância de cada um desses raios, gerando uma representação RGB-D (cor + distância) que pode ser diretamente mapeada para uma nuvem de pontos colorida. Do ponto de vista físico, não é possível obter a informação de profundidade a partir de uma única captura esférica, visto que qualquer ponto ao longo de um raio 3D é mapeado na mesma posição da superfície da esfera unitária. Dessa forma, seriam necessárias duas ou mais capturas da cena para inferir a profundidade da cena, como na estereoscopia clássica [Hartley and Zisserman 2003]. Em particular, considerar múltiplas vistas (mais de duas) adiciona robustez na estimativa da profundidade [da Silveira and Jung 2019a]. Por outro lado, o uso de múltiplas imagens requer a obtenção de múltiplas capturas da mesma cena, o que pode ser um fator complicador sobretudo se houver objetos em movimento na cena: capturas em instantes de tempos distintos não geram vistas da mesma cena devido ao movimento relativo dos objetos dinâmicos.

Apesar de ser fisicamente implausível, se pode estimar a profundidade a partir de uma única captura. No caso de imagens em perspectiva, os seres humanos conseguem estimar distâncias a partir de uma única imagem. Para tal, usam relações entre objetos cujos tamanhos são conhecidos no mundo real, e inferem uma noção de distância a partir do tamanho projetado do objeto na imagem: quanto mais distante, menor será a projeção do objeto. Inspirados por essa característica, vários algoritmos de visão computacional baseados em aprendizado de máquina têm sido propostos para estimar a profundidade a partir de uma única imagem, problema comumente chamado de *single-image stereo* ou *monocular depth estimation* [Masoumian et al. 2022]. Seguindo essa estratégia baseada

em aprendizado de máquina, alguns autores têm atacado o problema de estimativa de profundidade usando uma única captura panorâmica.

A maioria das abordagens de estimativa de profundidade a partir de um único panorama adota imagens ERP [Tateno et al. 2018, Sun et al. 2021, Albanis et al. 2021], que contêm todas as informações contextuais, mas apresentam distorções acentuadas, ou imagens de múltiplos planos [da Silveira et al. 2018, Rey-Area et al. 2022], que aliviam as distorções, mas exigem que as projeções individuais sejam unidas de volta à esfera. Conforme discutido brevemente na Seção 2.3.2, alguns trabalhos utilizam duas representações planares da esfera e incorporam o mapeamento delas no processo de aprendizado para aliviar os problemas de cada uma individualmente [Wang et al. 2020a, Jiang et al. 2021]. Outras abordagens consideram representações de imagem baseadas em icosaedro e arquiteturas de rede especializadas [Lee et al. 2020].

As abordagens que exploram uma única representação ERP geralmente estão restritas a panoramas de baixa resolução (512×256 ou 1024×512) devido às limitações de memória da GPU, o que pode não ser adequado para aplicações de realidade virtual que envolvem HMDs de alta qualidade [Liu et al. 2021]. Para esses cenários, o uso de representações de múltiplos planos [da Silveira et al. 2018, Rey-Area et al. 2022] é uma solução possível, já que cada projeção planar contém uma imagem com um FoV estreito que pode ser tratado individualmente. No entanto, tais abordagens devem lidar com descontinuidades entre representações planares adjacentes e falta de informação contextual para processar cada projeção planar. Outro desafio na estimativa de profundidade monocular diz respeito a cenas capturadas em ambientes externos. A maioria dos métodos pode não generalizar para esses cenários devido à falta de conjuntos de dados anotados (necessários para o treinamento supervisionado), e eventual incapacidade para lidar com valores de profundidade infinitos, como no céu. Felizmente, novas abordagens começaram a abordar esse aspecto relevante [Bhanushali et al. 2022].

A Figura 2.6a mostra a nuvem de pontos colorida (vista de fora da sala) associada à estimativa de mapa de profundidade pelo modelo de aprendizado baseado em U-Net de[Albanis et al. 2021] usando a imagem na Figura 2.3a como entrada. Embora a estrutura externa do ambiente tenha sido capturado, percebe-se que que as paredes laterais e o teto não são exatamente planares.

Técnicas de estimativa de profundidade densas são capazes de estimar a posição 3D de cada ponto do panorama. Por um lado, permitem uma granularidade fina da cena; por outro lado, podem não considerar alguma informação geométrica pré-existente na cena e gerar modelos 3D com menor precisão, como ilustrado na Figura 2.6a (o fato de que paredes em um ambiente interno são normalmente planares não foi levado em consideração, e gerou "ondulações" na nuvem de pontos estimada).

Para algumas aplicações específicas, como a modelagem de ambientes internos cuja geometria segue padrões pré-definidos, se pode fazer o uso de técnicas de estimativa de *layout*. Os métodos de estimativa de *layout* têm como objetivo recuperar uma representação tridimensional esparsa a partir de um panorama capturado no interior de um ambiente, gerando informações sobre a geometria do mesmo (quintas e junções entre paredes, teto e piso, por exemplo).

(a) (b)

Figura 2.6: Duas maneiras de extração de um modelo 3D a partir da ERP ilustrada na Figura 2.3a: (a) estimativa de profundidade densa (ponto-a-ponto) usando [Albanis et al. 2021] e (b) estimativa do *layout* usando [Wang et al. 2021].

Métodos pioneiros para estimativa de *layout* eram semi-automáticos ou usavam explicitamente primitivas geométricas ou pontos/linhas de fuga [Jia and Li 2015]. Já abordagens mais recentes [Sun et al. 2019, Sun et al. 2021, Fernandez-Labrador et al. 2020, Wang et al. 2021, Jiang et al. 2022, Pintore et al. 2020a] atacam o problema a partir de uma perspectiva de aprendizado de máquina, e usam uma única imagem omnidirecional como entrada. As técnicas existentes exploram diferentes maneiras de representar o panorama (ERP, CMP, etc.) e adotam arquiteturas de rede compatíveis com o dado de entrada, como discutido na Seção 2.3. Destacamos que alguns métodos [Zou et al. 2018, Zhang et al. 2014] incluem uma etapa de pré-processamento que alinha as imagens esféricas verticalmente antes da inferência do *layout*, ressaltando a importância dos métodos discutidos na Seção 2.4.1. A maioria dos métodos de estimativa de *layout* opta por fazer a regressão das junções de paredes/teto/piso [Sun et al. 2019, Sun et al. 2021, Fernandez-Labrador et al. 2020] de ambientes internos, embora também seja possível inferir suas bordas [Pintore et al. 2020b].

Abordagens de estimativa de *layout* se baseiam em restrições geométricas que guiam o processo de otimização [da Silveira et al. 2022]. O modelo geométrico mais simples usado na estimativa de *layout* é um cuboide, que implica um *layout* em forma de caixa (também chamada de cuboide) [Zhang et al. 2014]. A forma de cuboide é um caso particular para a hipótese de "Mundo Manhattan", no qual o *layout* da sala tem paredes perpendiculares entre si [Fernandez-Labrador et al. 2020, Wang et al. 2021], ou seja, o plano de chão do ambiente é uma região poligonal com segmentos de reta ortogonais entre si. Assim, os modelos Manhattan compreendem ambientes mais complexos, como salas em forma de "L". Mundos Manhattan aumentados podem ter paredes que não são perpendiculares entre si [Pintore et al. 2018, Fernandez-Labrador et al. 2020]. Por fim, a restrição de *layout* mais genérica é chamada de suposição de Mundo Atlanta, onde até paredes curvas podem existir, desde que o teto e o piso sejam paralelos [Pintore et al. 2020a].

A Figura 2.6b mostra um exemplo de estimativa de *layout* a partir de um único panorama, onde o método de[Wang et al. 2021] utiliza a representação ERP da cena *Classroom* mostrada na Figura 2.3a como entrada. Embora o método de[Wang et al. 2021]

seja capaz de lidar com layouts genéricos de Manhattan, ele consegue detectar corretamente o *layout* em forma de cuboide da imagem de entrada, gerando paredes planares e ortogonais, contrastando com a técnica densa mostrada na Figura 2.6a. Por outro lado, objetos não-planares (como as mesas no chão) estão distorcidos na representação final e planificados na face do cuboide que representa o chão.

Estabelecer um *benchmark* para estimativas de profundidade e *layout* de panoramas é um desafio devido à diversidade de abordagens listadas nesta seção. O leitor pode consultar os trabalhos [da Silveira and Jung 2023] e [da Silveira et al. 2022] que compilam bases de dados e métricas de avaliação adequadas às variantes desses dois problemas.

2.5. Considerações Finais

Este capítulo tem como objetivo fornecer uma introdução sólida à computação visual omnidirecional. Inicialmente, ele revisou o modelo de imageamento esférico, *pipelines* de aquisição existentes e formatos de representação (multi-)planar proeminentes usados para armazenar e processar mídia omnidirecional. Em seguida, o artigo apresentou os principais desafios das representações omnidirecionais, focando nas distorções inerentes a essas representações e em seu impacto nas arquiteturas de aprendizado profundo, que são a abordagem atual para processar panoramas. Este capítulo também apresentou tendências recentes para mitigar esses desafios e mostrou os avanços em três cenários de aplicação que exploram plenamente imagens omnidirecionais.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

[Adarve and Mahony 2017] Adarve, J. D. and Mahony, R. (2017). Spherepix: A data structure for spherical image processing. *IEEE Robotics and Automation Letters*, 2(2):483–490.

[Aggarwal et al. 2016] Aggarwal, R., Vohra, A., and Namboodiri, A. M. (2016). Panoramic Stereo Videos with a Single Camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3755–3763.

[Akihiko et al. 2005] Akihiko, T., Atsushi, I., and Ohnishi, N. (2005). Two-and three-view geometry for spherical cameras. *Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*, 105:29–34.

[Albanis et al. 2021] Albanis, G., Zioulis, N., Drakoulis, P., Gkitsas, V., Sterzentsenko, V., Alvarez, F., Zarpalas, D., and Daras, P. (2021). Pano3d: A holistic benchmark and a solid baseline for 360° depth estimation. In *Conference on Computer Vision and Pattern Recognition Workshops*, pages 3722–3732.

[Azevedo et al. 2020] Azevedo, R. G. d. A., Birkbeck, N., De Simone, F., Janatra, I., Adsumilli, B., and Frossard, P. (2020). Visual Distortions in 360-degree Videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(8):2524–2537.

[Bai et al. 2022] Bai, J., Lai, S., Qin, H., Guo, J., and Guo, Y. (2022). Glpanodepth: Global-to-local panoramic depth estimation. *arXiv preprint arXiv:2202.02796*.

[Bergmann et al. 2021] Bergmann, M. A., Pinto, P. G. L., da Silveira, T. L. T., and Jung, C. R. (2021). Gravity alignment for single panorama depth inference. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 1–8. IEEE.

[Bernal-Berdun et al. 2022] Bernal-Berdun, E., Martin, D., Gutierrez, D., and Masia, B. (2022). Sst-sal: A spherical spatio-temporal approach for saliency prediction in 360° videos. *Computers & Graphics*, 106:200–209.

[Bhanushali et al. 2022] Bhanushali, J., Chakravarthula, P., and Muniyandi, M. (2022). OmniHorizon: In-the-wild outdoors depth and normal estimation from synthetic omnidirectional dataset.

[Coors et al. 2018] Coors, B., Condurache, A. P., and Geiger, A. (2018). SphereNet: Learning spherical representations for detection and classification in omnidirectional images. *European Conference on Computer Vision*, pages 525–541.

[Cruz-Mota et al. 2012] Cruz-Mota, J., Bogdanova, I., Paquier, B., Bierlaire, M., and Thiran, J. P. (2012). Scale invariant feature transform on the sphere: Theory and applications. *International Journal of Computer Vision*, 98(2):217–241.

[da Silveira and Jung 2023] da Silveira, T. L. and Jung, C. R. (2023). Omnidirectional visual computing: Foundations, challenges, and applications. *Computers & Graphics*, 113:89–101.

[da Silveira et al. 2018] da Silveira, T. L. T., Dalaqua, L. P., and Jung, C. R. (2018). Indoor Depth Estimation from Single Spherical Images. In *IEEE International Conference on Image Processing*, pages 2935–2939.

[da Silveira et al. 2021] da Silveira, T. L. T., de Oliveira, A. Q., Walter, M., and Jung, C. R. (2021). Fast and accurate superpixel algorithms for 360° images. *Signal Processing*, 189:108277.

[da Silveira and Jung 2019a] da Silveira, T. L. T. and Jung, C. R. (2019a). Dense 3D Scene Reconstruction from Multiple Spherical Images for 3-DoF+ VR Applications. In *IEEE Conference on Virtual Reality and 3D User Interfaces*, pages 9–18.

[da Silveira and Jung 2019b] da Silveira, T. L. T. and Jung, C. R. (2019b). Perturbation Analysis of the 8-Point Algorithm: A Case Study for Wide FoV Cameras. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11757–11766.

[da Silveira et al. 2022] da Silveira, T. L. T., Pinto, P. G. L., Murrugarra-Llerena, J., and Jung, C. R. (2022). 3d scene geometry estimation from 360° imagery: A survey. *ACM Comput. Surv.*, 55(4).

[Dai et al. 2019] Dai, F., Zhu, C., Ma, Y., Cao, J., Zhao, Q., and Zhang, Y. (2019). Freely Explore the Scene with 360° Field of View. In *IEEE Conference on Virtual Reality and 3D User Interfaces*, pages 888–889.

[Dai et al. 2021] Dai, Z., Liu, H., Le, Q. V., and Tan, M. (2021). Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977.

[De Simone et al. 2017] De Simone, F., Frossard, P., Wilkins, P., Birkbeck, N., and Kokaram, A. (2017). Geometry-driven quantization for omnidirectional image coding. *2016 Picture Coding Symposium, PCS 2016*.

[Dosovitskiy et al. 2020] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

[Ebrahimi et al. 2016] Ebrahimi, T., Foessel, S., Pereira, F., and Schelkens, P. (2016). JPEG Pleno: Toward an Efficient Representation of Visual Reality. *IEEE Multimedia*, 23(4):14–20.

[Eder et al. 2019] Eder, M., Moulon, P., and Guan, L. (2019). Pano Popups: Indoor 3D Reconstruction with a Plane-Aware Network. In *2019 International Conference on 3D Vision (3DV)*, pages 76–84. IEEE.

[Eder et al. 2020] Eder, M., Shvets, M., Lim, J., and Frahm, J.-M. (2020). Tangent images for mitigating spherical distortion. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[Fangi et al. 2018] Fangi, G., Pierdicca, R., Sturari, M., and Malinverni, E. S. (2018). Improving spherical photogrammetry using 360° OMNI-Cameras: Use cases and new applications. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42(2):331–337.

[Fernandez-Labrador et al. 2020] Fernandez-Labrador, C., Facil, J. M., Perez-Yus, A., Demonceaux, C., Civera, J., and Guerrero, J. (2020). Corners for layout: End-to-end layout recovery from 360 images. *IEEE Robotics and Automation Letters*, pages 1–1.

[Ferreira et al. 2017] Ferreira, L. S., Sacht, L., and Velho, L. (2017). Local Moebius transformations applied to omnidirectional images. *Computers & Graphics*, 68:77–83.

[Fujiki et al. 2007] Fujiki, J., Torii, A., and Akaho, S. (2007). Epipolar Geometry Via Rectification of Spherical Images. In *Computer Vision/Computer Graphics Collaboration Techniques*, volume 4418, pages 461–471. Springer Berlin Heidelberg.

[Gava et al. 2018] Gava, C. C., Stricker, D., and Yokota, S. (2018). Dense Scene Reconstruction from Spherical Light Fields. In *IEEE International Conference on Image Processing*, pages 4178–4182.

[Goodfellow et al. 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. The MIT Press.

[Gou et al. 2021] Gou, J., Yu, B., Maybank, S. J., and Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819.

[Guan and Smith 2017] Guan, H. and Smith, W. A. P. (2017). Structure-From-Motion in Spherical Video Using the von Mises-Fisher Distribution. *IEEE Transactions on Image Processing*, 26(2):711–723.

[Hartley and Zisserman 2003] Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge.

[Im et al. 2016] Im, S., Ha, H., Rameau, F., Jeon, H.-G., Choe, G., and Kweon, I. S. (2016). All-around depth from small motion with a spherical panoramic camera. In *European Conference on Computer Vision*, pages 156–172.

[J. Huang et al. 2017] J. Huang, Z. Chen, D. Ceylan, and H. Jin (2017). 6-DoF VR videos with a single 360-camera. In *IEEE Virtual Reality*, pages 37–44.

[J. Xiao et al. 2012] J. Xiao, E., K. A., Oliva, A., and Torralba, A. (2012). Recognizing scene viewpoint using panoramic place representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2695–2702.

[Jia and Li 2015] Jia, H. and Li, S. (2015). Estimating structure of indoor scene from a single full-view image. In *IEEE International Conference on Robotics and Automation*, pages 4851–4858.

[Jiang et al. 2021] Jiang, H., Sheng, Z., Zhu, S., Dong, Z., and Huang, R. (2021). Unifuse: Unidirectional fusion for 360° panorama depth estimation. *IEEE Robotics and Automation Letters*, 6(2):1519–1526.

[Jiang et al. 2022] Jiang, Z., Xiang, Z., Xu, J., and Zhao, M. (2022). LGT-net: Indoor panoramic room layout estimation with geometry-aware transformer network. In *Conference on Computer Vision and Pattern Recognition*.

[Jung et al. 2019] Jung, R., Lee, A. S. J., Ashtari, A., and Bazin, J.-C. (2019). Deep360Up: A Deep Learning-Based Approach for Automatic VR Image Upright Adjustment. In *IEEE Conference on Virtual Reality and 3D User Interfaces*, pages 1–8.

[Krolla et al. 2014] Krolla, B., Diebold, M., Goldlücke, B., and Stricker, D. (2014). Spherical light fields. *British Machine Vision Conference*, (67.1-67.12).

[Lee et al. 2019] Lee, Y., Jeong, J., Yun, J., Cho, W., and Yoon, K.-J. (2019). Spherephd: Applying cnns on a spherical polyhedron representation of 360 images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9181–9189.

[Lee et al. 2020] Lee, Y., Jeong, J., Yun, J., Cho, W., and Yoon, K.-J. (2020). Spherephd: Applying cnns on 360° images with non-euclidean spherical polyhedron representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

[Li et al. 2023] Li, M., Wang, S., Yuan, W., Shen, W., Sheng, Z., and Dong, Z. (2023). $\mathscr{S}^2$net: Accurate panorama depth estimation on spherical surface. *IEEE Robotics and Automation Letters*, 8(2):1053–1060.

[Li 2008] Li, S. (2008). Binocular spherical stereo. *IEEE Transactions on Intelligent Transportation Systems*, 9(4):589–600.

[Li et al. 2022] Li, Y., Barnes, C., Huang, K., and Zhang, F.-L. (2022). Deep 360° optical flow estimation based on multi-projection fusion. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 336–352. Springer.

[Liu et al. 2021] Liu, R., Peng, C., Zhang, Y., Husarek, H., and Yu, Q. (2021). A survey of immersive technologies and applications for industrial product development. *Computers & Graphics*, 100:137–151.

[Lo et al. 2018] Lo, I., Shih, K., and Chen, H. H. (2018). Image stitching for dual fisheye cameras. In *IEEE International Conference on Image Processing*, pages 3164–3168.

[Masoumian et al. 2022] Masoumian, A., Rashwan, H. A., Cristiano, J., Asif, M. S., and Puig, D. (2022). Monocular depth estimation using deep learning: A review. *Sensors*, 22(14):5353.

[Murrugarra-Llerena et al. 2022] Murrugarra-Llerena, J., da Silveira, T. L. T., and Jung, C. R. (2022). Pose estimation for two-view panoramas based on keypoint matching: A comparative study and critical analysis. In *Conference on Computer Vision and Pattern Recognition Workshops*, pages 5202–5211.

[Nayar 1997] Nayar, S. K. (1997). Catadioptric Omnidirectional Camera*. In *Conference on Computer Vision and Pattern Recognition*, pages 482–488.

[Pintore et al. 2020a] Pintore, G., Agus, M., and Gobbetti, E. (2020a). AtlantaNet: Inferring the 3D indoor layout from a single 360 image beyond the Manhattan world assumption. In *European Conference on Computer Vision*.

[Pintore et al. 2020b] Pintore, G., Mura, C., Ganovelli, F., Fuentes-Perez, L., Pajarola, R., and Gobbetti, E. (2020b). State-of-the-art in automatic 3d reconstruction of structured indoor environments. *Computer Graphics Forum*, 39(2).

[Pintore et al. 2018] Pintore, G., Pintus, R., Ganovelli, F., Scopigno, R., and Gobbetti, E. (2018). Recovering 3d existing-conditions of indoor structures from spherical images. *Computers & Graphics*, 77:16–29.

[Rey-Area et al. 2022] Rey-Area, M., Yuan, M., and Richardt, C. (2022). 360monodepth: High-resolution 360° monocular depth estimation. In *Conference on Computer Vision and Pattern Recognition*, pages 3762–3772.

[S. Li and Fukumori 2005] S. Li and Fukumori, K. (2005). Spherical stereo for the construction of immersive vr environment. In *IEEE Virtual Reality*, pages 217–222.

[Serrano et al. 2019] Serrano, A., Kim, I., Chen, Z., DIVerdi, S., Gutierrez, D., Hertzmann, A., and Masia, B. (2019). Motion parallax for 360° RGBD video. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):1817–1827.

[Shan and Li 2018] Shan, Y. and Li, S. (2018). Descriptor Matching for a Discrete Spherical Image With a Convolutional Neural Network. *IEEE Access*, 6:20748–20755.

[Shen et al. 2022] Shen, Z., Lin, C., Liao, K., Nie, L., Zheng, Z., and Zhao, Y. (2022). Panoformer: Panorama transformer for indoor 360° depth estimation. In *European Conference on Computer Vision*, pages 195–211. Springer.

[Su and Grauman 2017] Su, Y.-C. and Grauman, K. (2017). Learning Spherical Convolution for Fast Features from 360° Imagery. In *Conference on Neural Information Processing Systems*, pages 529–539.

[Sun et al. 2019] Sun, C., Hsiao, C.-W., Sun, M., and Chen, H.-T. (2019). HorizonNet: Learning Room Layout with 1D Representation and Pano Stretch Data Augmentation. pages 1047–1056.

[Sun et al. 2021] Sun, C., Sun, M., and Chen, H.-T. (2021). Hohonet: 360 indoor holistic understanding with latent horizontal features. In *Conference on Computer Vision and Pattern Recognition*, pages 2573–2582.

[Tateno et al. 2018] Tateno, K., Navab, N., and Tombari, F. (2018). Distortion-Aware Convolutional Filters for Dense Prediction in Panoramic Images. *European Conference on Computer Vision*, pages 732–750.

[Vaswani et al. 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008.

[Wang et al. 2018a] Wang, F.-E., Hu, H.-N., Cheng, H.-T., Lin, J.-T., Yang, S.-T., Shih, M.-L., Chu, H.-K., and Sun, M. (2018a). Self-supervised Learning of Depth and Camera Motion from 360° Videos. volume 11364, pages 53–68. Asian Conference on Computer Vision.

[Wang et al. 2020a] Wang, F.-E., Yeh, Y.-H., Sun, M., Chiu, W.-C., and Tsai, Y.-H. (2020a). Bifuse: Monocular 360 depth estimation via bi-projection fusion. In *Conference on Computer Vision and Pattern Recognition*.

[Wang et al. 2021] Wang, F.-E., Yeh, Y.-H., Sun, M., Chiu, W.-C., and Tsai, Y.-H. (2021). LED2-Net: Monocular 360° layout estimation via differentiable depth rendering. pages 12956–12965.

[Wang et al. 2020b] Wang, M., Lyu, X.-Q., Li, Y.-J., and Zhang, F.-L. (2020b). VR content creation and exploration with deep learning: A survey. *Computational Visual Media*, 6(1):3–28.

[Wang et al. 2018b] Wang, T.-H., Huang, H.-J., Lin, J.-T., Hu, C.-W., Zeng, K.-H., and Sun, M. (2018b). Omnidirectional CNN for visual place recognition and navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2341–2348. IEEE.

[Xu et al. 2022] Xu, C., Yang, H., Han, C., and Zhang, C. (2022). Pcformer: A parallel convolutional transformer network for 360° depth estimation. *IET Computer Vision*.

[Xu et al. 2020] Xu, M., Li, C., Zhang, S., and Callet, P. L. (2020). State-of-the-art in 360° video/image processing: Perception, assessment and compression. *IEEE Journal of Selected Topics in Signal Processing*, 14(1):5–26.

[Yang et al. 2018] Yang, W., Qian, Y., Kamarainen, J. K., Cricri, F., and Fan, L. (2018). Object Detection in Equirectangular Panorama. *International Conference on Pattern Recognition*, pages 2190–2195.

[Yu and Koltun 2015] Yu, F. and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

[Zelnik-Manor et al. 2005] Zelnik-Manor, L., Peters, G., and Perona, P. (2005). Squaring the circle in panoramas. In *IEEE International Conference on Computer Vision*, volume 2, pages 1292–1299 Vol. 2.

[Zhang et al. 2022a] Zhang, J., Yang, K., Ma, C., Reiß, S., Peng, K., and Stiefelhagen, R. (2022a). Bending reality: Distortion-aware transformers for adapting to panoramic semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 16917–16927.

[Zhang et al. 2014] Zhang, Y., Song, S., Tan, P., and Xiao, J. (2014). PanoContext: A whole-room 3D context model for panoramic scene understanding. In *European Conference on Computer Vision*.

[Zhang et al. 2021] Zhang, Y., Zhang, F.-L., Lai, Y.-K., and Zhu, Z. (2021). Efficient propagation of sparse edits on 360° panoramas. *Computers & Graphics*, 96:61–70.

[Zhang et al. 2022b] Zhang, Y., Zhang, F.-L., Zhu, Z., Wang, L., and Jin, Y. (2022b). Fast edit propagation for 360 degree panoramas using function interpolation. *IEEE Access*, 10:43882–43894.

[Zhuang et al. 2022] Zhuang, C., Lu, Z., Wang, Y., Xiao, J., and Wang, Y. (2022). Acdnet: Adaptively combined dilated convolution for monocular panorama depth estimation. In *AAAI Conference on Artificial Intelligence*, volume 36, pages 3653–3661.

[Zioulis et al. 2021] Zioulis, N., Alvarez, F., Zarpalas, D., and Daras, P. (2021). Single-shot cuboids: Geodesics-based end-to-end manhattan aligned layout estimation from spherical panoramas.

[Zioulis et al. 2018] Zioulis, N., Karakottas, A., Zarpalas, D., and Daras, P. (2018). OmniDepth: Dense Depth Estimation for Indoors Spherical Panoramas. In *European Conference on Computer Vision*, pages 453–471.

[Zou et al. 2018] Zou, C., Colburn, A., Shan, Q., and Hoiem, D. (2018). LayoutNet: Reconstructing the 3D Room Layout from a Single RGB Image. In *Conference on Computer Vision and Pattern Recognition*, pages 2051–2059.

Capítulo 3

Pensamento Computacional Paralelo: Desafios do Presente e do Futuro

Arthur F. Lorenzon e Lucas Mello Schnorr

Programa de Pós-Graduação em Computação (PPGC) - UFRGS

Abstract

Parallel processing becomes a fundamental piece in scientific and technological development. The advancement of artificial intelligence and machine learning relies on complex algorithms and models that require significant computational power for training and inference. In the scientific field, many computational simulations and modeling of natural phenomena demand immense processing power to perform complex calculations within a reasonable timeframe. Regarding emerging technologies like virtual reality and autonomous vehicles, such technologies require real-time processing power to provide immersive experiences and make swift decisions based on constantly changing inputs and data. Last but not least, the field of medicine and genomic researchbenefits from parallel computing to expedite DNA sequencing and identify genetic patterns relevant to diseases and treatments, leading to significant advances in personalized medicine.As the demand for computational power increases, there's a noticeable rise in the number of processing cores within a single chip, an increase in the scale of interconnected computer clusters through low-latency networks, and a higher quantity of general-purpose vector accelerators (GPGPUs). To harness the full potential of this performance, we need to develop applications capable of efficiently exploiting such architectures. The primary approach involves breaking down complex tasks into several smaller parts processed in parallel, potentially resulting in increased performance and the ability to handle larger workloads. Thus, this mini-course aims to introduce the concepts of parallel computational thinking and parallel programming, aiming to instill in participants the important reflexes necessary to break down the solution to a large problem into a set of smaller problems that can be calculated concurrently, effectively identifying the parallelism of the solution. We will also address technological trends for the future of parallel processing. Ultimately, it's

Vídeo com a apresentação do capítulo: `https://youtu.be/_eCci_5upLU`

expected that these elements will lead to the development of suitable parallel solutions for current societal problems.

Resumo

O processamento paralelo se torna uma peça fundamental no desenvolvimento científico e tecnológico. O avanço da inteligência artificial e aprendizagem de máquina dependem de algoritmos e modelos complexos que requerem poder computacional significativo para treinamento e inferência. Na área científica, muitas simulações computacionais e modelagem de fenômenos naturais exigem enormes quantidades de poder de processamento para executar cálculos complexos em um tempo razoável. Com relação às tecnologias emergentes, como realidade virtual e veículos autônomos, tais tecnologias requerem poder de processamento em tempo real para fornecer experiências imersivas e tomar decisões rápidas com base em entradas e dados em constante mudança. Por fim, mas não menos importante, a área da medicina e pesquisa genômica se beneficia da computação paralela para acelerar o sequenciamento de DNA e identificar padrões genéticos relevantes para doenças e tratamentos, levando a significativos avanços na medicina personalizada. Ao mesmo tempo que a demanda por poder computacional aumenta, observa-se um aumento da quantidade de núcleos de processamento em um único chip, um aumento na escala de clusters de computadores interconectados por redes de baixa latência e uma maior quantidade de aceleradores vetoriais de propósito geral (GPGPUs). Para extrair todo este potencial de desempenho, precisamos desenvolver aplicações capazes de explorar tais arquiteturas de maneira eficiente. A principal abordagem é a divisão de tarefas complexas em várias partes menores que são processadas em paralelo, resultando potencialmente em um aumento no desempenho e na capacidade de lidar com cargas de trabalho maiores. Assim, este minicurso tem por objetivo apresentar os conceitos de ***pensamento computacional paralelo*** *e* ***programação paralela****, com o objetivo de instigar nos participantes os reflexos importantes necessários para quebrar a solução de um problema grande em um conjunto de problemas menores que podem ser calculados de maneira concorrente, efetivamente identificando o paralelismo da solução. Abordaremos também as tendências tecnológicas para o futuro do processamento paralelo. Enfim, espera-se que estes elementos levem ao desenvolvimento de soluções paralelas adequadas para problemas atuais da sociedade.*

3.1. Introdução

A computação tem sido objeto de um interesse cada vez maior por nações ao redor do mundo para inclusão de seu currículo como elemento base de escolas. O intuito principal é que a computação ocupe espaços como já fazem a matemática, a física, a geografia, a filosofia e a língua oficial do país. O *pensamento computacional*, ou seja, o método onde quebra-se problemas maiores em subproblemas, é sem dúvida fundamental para os dias de hoje e sobretudo para o futuro, para a próxima geração da sociedade, pois permite resolver problemas de maneira mais eficiente com a programação de computadores. Enquanto saúda-se e encoraja-se esses esforços, ressaltamos aqui a importância coadjuvante mas cada vez mais essencial do **pensamento computacional paralelo**. Além da tradicional quebra em subproblemas, a forma de pensar "em paralelo" exige imaginar atividades e

tarefas que possam ser executadas de maneira concorrente. É inegável a importância desse tipo de método, pois o que se observa é uma tendência inexorável para plataformas computacionais cada vez mais paralelas, compostas de uma grande quantidade de núcleos de processamento. O pensamento computacional paralelo é portanto fundamental para imaginar soluções que possam usufruir das plataformas computacionais modernas com a criação de aplicações paralelas.

O projeto de aplicações paralelas é uma etapa primordial na resolução de problemas complexos da sociedade atual, tais como previsão climática, mitigação de cenários de catástrofe, a procura por fontes renováveis de energia e a simulação de inundações para identificar áreas alagadiças. Em geral essas aplicações exigem um enorme poder computacional pois envolvem equações matemáticas complexas que modelam comportamentos físicos. As simulações decorrentes podem inclusive substituir onerosos experimentos reais, permitindo a criação de soluções inovadoras por um número maior de pesquisadores e empreendedores.

Projetar de maneira adequada uma aplicação paralela envolve uma interdisciplinariedade muito grande, pois se de um lado exige conhecimentos profundos da solução candidata de um problema, de outro exige conhecimentos em poder adaptar os passos paralelos dessa solução para se executar adequadamente a uma plataforma de execução, seja esta um único computador (com múltiplos núcleos de processamento ou múltiplas placas aceleradoras) ou um *cluster* de computadores (vários nós computacionais interconectados por uma rede de interconexão). Enfim, as decisões na fase de projeto da aplicação paralela acabam sendo determinantes para se obter um bom desempenho e um uso eficiente dos recursos de processamento.

Este curso aborda de maneira ampla os conceitos de **pensamento computacional paralelo** e **programação paralela**, com o objetivo de instigar nos participantes os reflexos importantes necessários para quebrar a solução de um problema grande em um conjunto de problemas menores que podem ser calculados de maneira concorrente, efetivamente identificando o paralelismo da solução. Ainda que esta etapa seja independente da plataforma computacional subjacente, pretendemos também abordar escolhas importantes na identificação de paralelismo quando se define uma plataforma alvo específica, assim como as tendências tecnológicas na área de processamento paralelo. Enfim, espera-se que tais intuições do pensamento computacional paralelo levem ao desenvolvimento de soluções adequadas para problemas atuais da sociedade e possam contribuir para o avanço societal.

A Seção 3.2 apresenta o método PCAM [Foster 1995] para exercitar o pensamento computacional paralelo. A Seção 3.3 traz as principais tendências tecnológicas para o futuro do processamento paralelo. A Seção 3.4 traz uma conclusão com reflexões futuras.

3.2. Pensamento Computacional Paralelo: Método PCAM

O método **PCAM**, ilustrado na Figura 3.1, envolve as etapas: particionamento, comunicação, aglomeração e mapeamento. As duas primeiras etapas tem um enfoque na escalabilidade de uma solução para um problema, ou seja, procuramos definir algoritmos capazes de resolver o problema de maneira mais paralela possível, com maior concorrência entre as unidades de processamento. Nas duas últimas etapas, de aglomeração e mapeamento, a preocupação do projetista se foca na preocupação com o desempenho

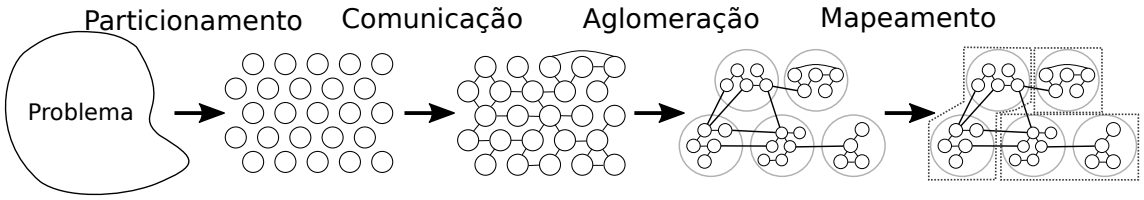


Figura 3.1. Metodologia PCAM com suas quatro etapas.

computacional (tempo de execução). São nestas duas últimas etapas que o conhecimento da configuração da plataforma de execução se torna vital.

Segue uma breve descrição de cada uma das etapas. No **Particionamento**, as operações que resolvem um determinado problema a devem ser quebradas em pedaços pequenos. O objetivo principal é detectar o paralelismo nestas operações. Na **Comunicação**, devemos definir quais são as atividades de comunicação necessárias para que a resolução de um problema, já dividida em pedaços, funcione de maneira apropriada e sem erros. Na **Aglomeração**, devemos avaliar se a solução respeita requisitos de desempenho computacional e custos de implementação. Enfim, no **Mapeamento**, devemos atribuir as tarefas, estática ou dinamicamente, às unidades de processamento. Aqui devemos maximizar o uso de recursos computacionais e minimizar atividades de comunicação.

Em um cenário ideal, espera-se que uma aplicação paralela criada no âmbito do processo metodológico PCAM seja capaz de explorar de maneira eficiente uma quantidade indeterminada de unidades de processamento. Mesmo assim, observa-se frequentemente a aplicação do método especificamente para uma plataforma computacional, como aquelas que serão abordadas na Seção 3.3 onde veremos tendências tecnológicas. Criar uma aplicação com portabilidade de desempenho é uma tarefa bastante difícil pois envolve utilizar algoritmos adaptativos em função da execução e do ambiente. Este tópico permanece como objeto de investigação.

3.2.1. Particionamento

A etapa de particionamento envolve a descoberta de oportunidades para execução paralela. A ideia é identificar o maior número possível de pequenas tarefas. Com isso, procura-se estabelecer qual o menor subproblema possível, consistindo de operações que são executadas sequencialmente. Este menor subproblema possível é então chamado de tarefa. Como consequência desse esforço, esse grão pequeno permitirá uma maior flexibilidade para a criação de algoritmos paralelos que suportem uma variadade maior de plataformas computacionais.

Normalmente, um tamanho de tarefa demasiadamente pequeno pode incutir em uma perda de desempenho no que diz respeito a quantidade de comunicações e ao gerenciamento da enorme quantidade de tarefas pequenas resultantes. Isso leva naturalmente a uma junção das operações de várias tarefas pequenas, efetivamente mudando a *granularidade* das tarefas. No âmbito do modelo PCAM, esta reflexão é relegada para mais tarde, na etapa de aglomeração.

Existem dois tipos de particionamento: de dados e de operações. O particionamento de dados é mais comum, sendo conhecida também por **decomposição de domínio**. Ela tem

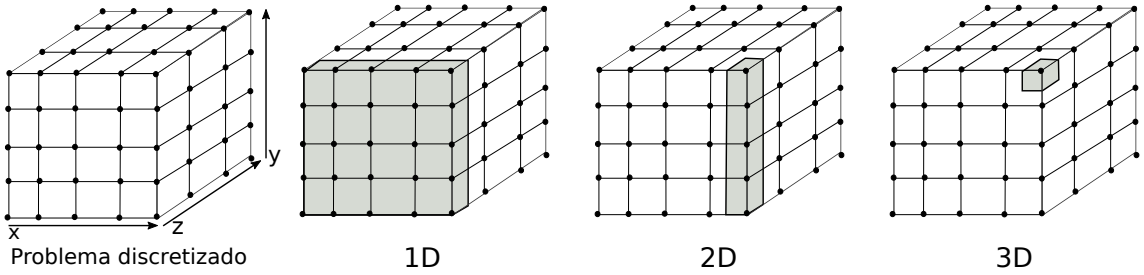


Figura 3.2. Três tipos de decomposição de domínio para o problema tridimensional à esquerda, com uma (1D), duas (2D) e três (3D) dimensões. O tamanho do dado na abordagem tridimensional é o menor possível (representado pelo conjunto de pontos na grade), pois envolve apenas um ponto na grade.

por objetivo quebrar o problema em pedaços suficientemente pequenos. Por simplicidade, esse processo é conduzido de forma a obter pedaços que sejam também de tamanhos idênticos, de forma a facilitar as etapas seguintes do método PCAM. Cada pedaço terá portanto os dados, resultante da partição pelo método, e as operações associadas. É importante quantificar o custo destas operações de forma que elas sejam consideradas, ainda que de maneira secundária, na definição do tamanho da partição de dados. O segundo tipo de particionamento envolve os tipos de operações (instruções) que devem ser computadas pela aplicação paralela, sendo conhecida por **decomposição funcional**. Neste caso, o projetista deve identificar partes no futuro código da aplicação que são funcionalmente independentes, prevendo sua execução de maneira concorrente. Ainda que idealmente o processo de particionamento possa se preocupar com a divisão dos *dados* e das *operações* conjuntamente, é comum adotar um ou outro tipo de decomposição de maneira independente.

A Figura 3.2 demonstra exemplos de decomposição de domínio de um problema tridimensional que já foi discretizado conforme ilustração na esquerda da figura. Esta discretização é representada pelos pontos do espaço tridimensional, com cinco coordenadas no eixo `x` e no eixo `y` e quatro no eixo `z`. O problema foi portanto discretizado em 100 pontos. Esta discretização pode então ser particionada em uma dimensão (1), com planos ao longo do eixo `z`, ou através de colunas (2D) ao longo do eixo `y`, ou através de uma partição tridimensional (3D) que envolve apenas um ponto. É esta última opção que permite a maior flexibilidade nas próximas etapas PCAM pois o tamanho da partição engloba um único ponto do domínio discretizado.

3.2.2. Comunicação

É comum, em um programa paralelo, que as tarefas necessitem trocar informações para realizar suas operações. Em PCAM, a etapa de comunicação envolve justamente o projeto destas atividades de troca de dados. Cenários onde as tarefas são independentes, portanto sem a necessidade de comunicação, são chamados de soluções *trivialmente paralelizáveis*. Nestes casos, basta realizar o particionamento e as etapas de aglomeração e mapeamento de PCAM.

Uma das principais preocupações com as atividades de comunicação é que elas possam ocorrer da maneira mais concorrente possível. Esse estado ideal pode ser atingido de

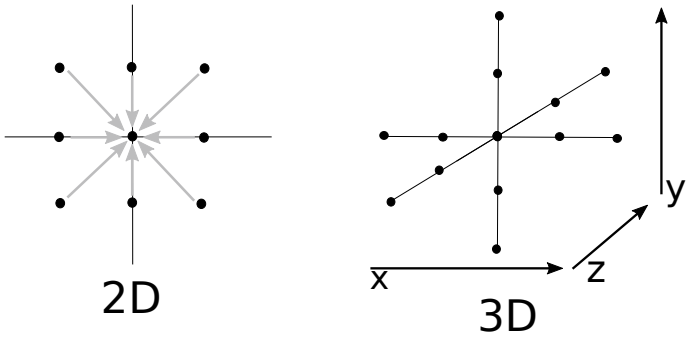


Figura 3.3. Comunicação local em particionamentos 2D (esquerda) e 3D (direita).

diferentes formas, através de variados padrões de comunicação. Uma classificação destes padrões pode seguir os seguintes eixos: comunicação local ou global, estruturada ou não, estática ou dinâmica, e síncrona ou não. Estes eixos são detalhados abaixo.

Local e Global. Uma comunicação local é obtida quando a operação de computação de uma determinada tarefa necessita dados de um pequeno número de tarefas vizinhas. A operação é conhecida por *stencil*, podendo ser configurada para um ambiente de variadas dimensões, de acordo com o domínio do problema. Por exemplo, na Figura 3.2 a quantidade de vizinhos imediatos no 3D é de seis, quatro de cada lado, um acima e outro abaixo. Neste caso, seis comunicações seriam necessárias antes da operação de cálculo. Quando o particionamento é bidimensional, como representado na ilustração 2D da Figura 3.3, a comunicação de todos os vizinhos pode ser necessária para um passo de simulação. Neste caso, existem oito operações de comunicações necessárias antes de efetuar as operações de cálculo da tarefa central. A localidade das comunicações podem variar bastante em função da complexidade da aplicação. Alguns cálculos podem exigir, por exemplo, dados de vizinhos de segunda ordem, conforme a ilustração 3D da Figura 3.3.

Uma operação de comunicação global pode envolver muitas tarefas, potencialmente todas aquelas que participam da aplicação paralela. Operações frequentemente reconhecidas como globais são aquelas de difusão massiva (*broadcast*) onde uma tarefa envia um dado para todas as outras, ou uma tarefa de redução, onde os dados de todas as tarefas são reduzidos por intermédio de um operador binário até uma única tarefa. Uma comunicação global pode ser implementada através de um algoritmo mestre-trabalhador. Neste caso, uma determinada tarefa fica responsável por recepcionar os dados de todas as outras que participam da computação. Este algoritmo tem duas características que tornam-no incapaz de atingir uma boa escalabilidade. Ele é centralizado e sequencial, uma vez que a tarefa mestre recebe as informações em uma determinada ordem. Uma forma mais eficiente de obter a mesma funcionalidade deste algoritmo é empregar uma árvore N-ária para difundir, ou receber, o dado mais rapidamente. Neste caso, as comunicações nas folhas e nós intermediários da árvore podem acontecer simultaneamente. O resultado é uma estrutura de comunicação regular na qual cada tarefa se comunica com poucos vizinhos próximos.

Estruturada ou não estruturada. As situações apresentadas até o momento são exemplos de uma estrutura de comunicação estática, ondes as tarefas tem vizinhos claramente definidos e imutáveis em função do particionamento estabelecido na etapa anterior. Neste

contexto as comunicações são frequentemente fixas, ditas estruturadas, e não evoluem ao longo da execução da aplicação paralela. Em outros casos, a grade de discretização pode seguir padrões mais complexos, conhecidos como não estruturados e irregulares. Por exemplo, um objeto irregular tal como o pulmão de uma pessoa pode ser melhor modelado por uma grade composta por formas simples, tais como triângulos, tetraedros, etc. Estas grades podem ser descritas por grafos, onde os vértices representam as tarefas e as arestas representam comunicação. Nestes casos, as atividades de comunicação entre as tarefas são mais complexas, envolvendo por vezes mais vizinhos em determinadas regiões.

Estática ou dinâmica. Grades de discretização podem ser regulares (veja exemplo na Figura 3.2) ou irregulares (exemplo na Figura 3.4), tais como um objeto modelado por formas como triângulos, etc. Uma diferença fundamental que pode afetar as demais etapas do projeto PCAM é se tais grades são estáticas ou dinâmicas. No caso de grades estáticas, a discretização é fixa desde a concepção na etapa de particionamento do projeto até a execução do código. No caso de grades dinâmicas, a discretização pode mudar em função da execução da aplicação paralela. Programas complexos podem aumentar a fidelidade de simulação em torno de objetos móveis em uma simulação ou em determinadas regiões de interesse, como bordas ou objetos relevantes.

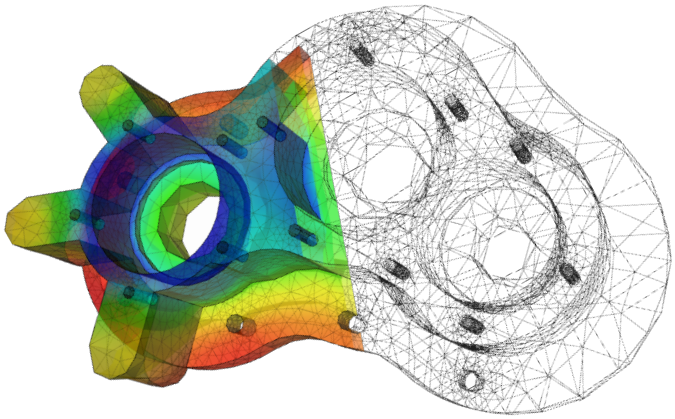


Figura 3.4. Grade de particionamento irregular tridimensional, representada por um grafo onde cada nó é uma tarefa e cada aresta é uma operação de comunicação (Artigo da Wikipedia em Alemão sobre Elementos Finitos).

Padrões irregulares de comunicação normalmente não afetam a etapa de particionamento. No exemplo da Figura 3.4, pode-se observar que algumas regiões tem uma intensidade de tarefas maior (pela proximidade física) que outras. O particionamento de um grafo como este pode implicar que cada nó de um grafo se torne uma tarefa e suas arestas se tornem comunicações. No entanto, uma grade irregular pode complicar bastante a condução das etapas de aglomeração e mapeamento. Por exemplo, ainda que a grade seja estática, a etapa de aglomeração pode ser complicada pois envolve antecipar o custo computacional das tarefas e a quantidade de dados da comunicação, de forma a criar grupos de tarefas que tenham por um lado um peso similar e que minimizem as comunicações. No caso da grade ser dinâmica, os algoritmos que realizam a aglomeração podem ser necessários durante a execução do programa incutindo em sobrecargas que devem ser pesados contra os benefícios trazidos por um melhor agrupamento de tarefas.

Síncrona ou Assíncrona. Até agora, vimos conceitos que consideram comunicação síncrona, onde as duas tarefas envolvidas na troca de dados estão cientes quando a operação acontece. Na comunicação assíncrona, por outro lado, as tarefas que possuem os dados, e que são responsáveis pelo envio, não estão cientes do momento quando as tarefas receptoras precisarão efetivamente dos dados da comunicação. Sendo assim, as tarefas receptoras devem registrar a necessidade de um dado que eventualmente será satisfeito, de maneira assíncrona, pela tarefa responsável por enviar o dado.

A grande vantagem de comunicações assíncronas é que elas podem ser utilizadas para esconder as comunicações. Em *middlewares* sofisticados de comunicação, a troca de dados de maneira assíncrona pode ser implementada de uma forma que a aplicação não fica bloqueada em nenhum momento esperando a conclusão do envio/recepção. Sendo assim, a aplicação paralela pode antecipar o registro da necessidade de um dado de forma que quando ele for necessário esse dado já tenha sido recebido. Esse conceito serve também do lado do envio, onde o gerador do dado registra o envio para quem precisa do dado assim que ele for gerado, potencializando a comunicação assíncrona.

3.2.3. Aglomeração

As etapas precedentes da metodologia PCAM permitem o particionamento e a definição das comunicações necessárias para a resolução paralela de um problema. O resultado destas etapas é um algoritmo abstrato que contém potencialmente muitas tarefas, tendo em vista que o objetivo é identificar a menor operação possível que possa ser executada concorrentemente com as demais. Esse algoritmo abstrato, distante da realidade, é normalmente ruim por ter tarefas demais, visto que somente o gerenciamento dessa quantidade enorme de tarefas é prejudicial para o desempenho. A etapa de *aglomeração* tem por objetivo tornar o algoritmo abstrato das etapas precedentes em algo mais realista, de acordo com os limites impostos pela configuração da plataforma de execução alvo. O objetivo principal é obter um programa eficiente nesta plataforma. Para atingir tal objetivo, é importante avaliar, analítica ou experimentalmente, o benefício da aglomeração de tarefas através do seu impacto em diretivas de comunicação e no tempo de execução. Abaixo são apresentados tópicos relacionados a granularidade de tarefas, a relação entre superfície e volume no particionamento, e uma discussão sobre replicação de cálculo e formas de evitar a comunicação.

Granularidade de tarefas. Na etapa de particionamento o objetivo é baseado na premissa de quanto mais tarefas melhor, ou seja, deve-se encontrar o menor conjunto de operações que possa ser executada sequencialmente. No entanto, observa-se que esse tipo de particionamento fino pode levar a elevados custos de comunicação que prejudicam o desempenho da aplicação, tendo em vista que a unidade de processamento para de executar código útil para se ocupar de envios e recepção de dados. A aglomeração permite então tornar as tarefas maiores, e na medida que isso ocorre, pode haver um efeito positivo da redução do custos de comunicação.

A Figura 3.5 mostra exemplos de aglomeração de tarefas a partir do particionamento original com uma tarefa por ponto, ilustrada na esquerda da figura. As quatro opções de aglomeração, ilustradas no centro esquerda, ilustram uma aglomeração de pontos horizontal e vertical (na parte superior da figura), e dois planos possíveis (na parte inferior).

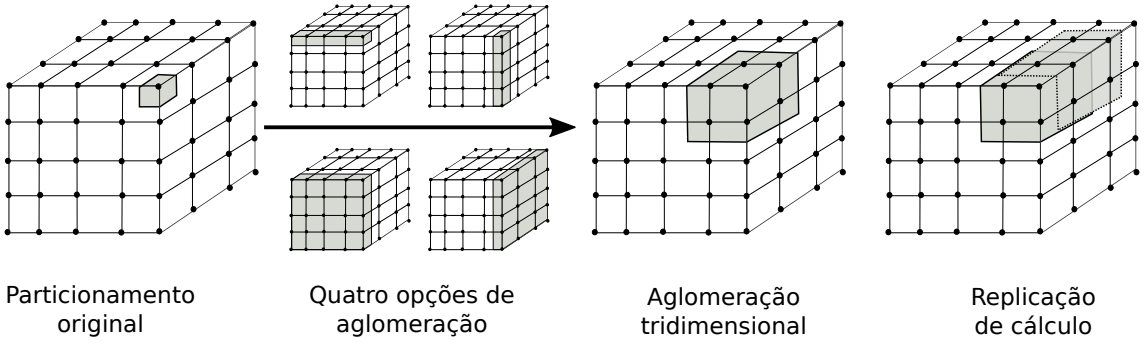


Figura 3.5. Exemplos de aglomeração de tarefas a partir do particionamento original (esquerda), e replicação de cálculo através da sobreposição de dois conjuntos aglomerados na decomposição de domínio (direita).

Enfim, uma aglomeração mais efetiva é a aquela tridimensional, como na ilustração do centro direita onde o bloco aglomera pontos em todas as três dimensões, reduzindo o perímetro do bloco, ou seja, as bordas que exigem comunicação com as tarefas vizinhas.

A aglomeração serve sobretudo para escolher o nível certo de concorrência que extrai o máximo de desempenho da plataforma de execução. Isso envolve então a redução das comunicações mas também pode ser influenciada por outras estratégias. Por exemplo, pode-se agregar dados a serem comunicados de forma que o envio seja feito com uma única operação ao invés de múltiplos envios. Isso permite evitar a latência da rede de interconexão. Pode-se também encontrar a menor quantidade de tarefas que maximize o desempenho, tendo em vista que que o excesso de tarefas pode ser penalizado pelo custo de gerenciamento das mesmas.

Outro ponto relevante na etapa de aglomeração é que a decisão sobre a granularidade das tarefas deve ser configurável. O programa deve ser capaz de permitir uma certa adaptabilidade tendo em vista a evolução dos computadores que podem tornar obsoleta uma determinada decisão de aglomeração.

Relação entre superfície e volume. A etapa de aglomeração traz o benefício de poder reduzir a quantidade de comunicação, como ilustrado no exemplo da Figura 3.6. Do lado esquerdo nós temos um particionamento fino de 8×8, com um total de 64 tarefas (representadas pelos círculos). Considerando que cada tarefa deve enviar 1 dado para cada um dos quatro vizinhos imediatos (conforme ilustrado nas tarefas em tons de cinza mais escuro), nós temos um total de 256 operações de comunicação cada uma com 1 dado. Ao realizar uma aglomeração bidimensional com um fator de 16 para 1, obtemos uma grade como aquela ilustrada na direita da figura, com quatro tarefas. Nesta configuração, são reduzidas não somente a quantidade de operações de comunicação para apenas 16 (pois cada uma das quatro tarefas se comunica com quatro vizinhos), mas também a quantidade de dados comunicados, pois envolve apenas o perímetro dos dados bidimensionais gerenciados por uma tarefa (os quadrados em tons cinza escuro na figura). Essa relação entre superfície e volume, ilustrada na figura através de um exemplo 2D, permite reduzir as necessidades de comunicação.

O efeito da etapa de aglomeração em grades não-estruturadas, como aquela exemplificada na Figura 3.4, são mais complexas de serem realizadas. Existem técnicas especializadas

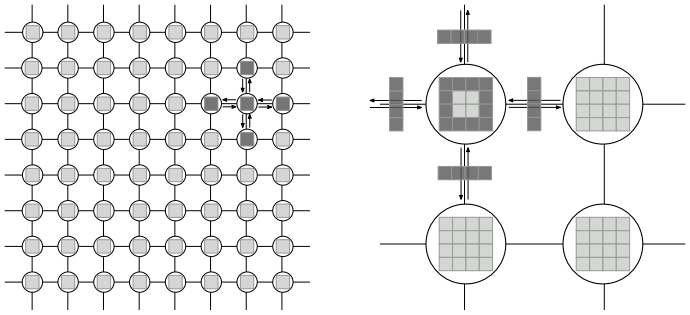


Figura 3.6. Efeito do aumento da granularidade nos custos de comunicação: em uma grade 8×8 (esquerda), o custo total de comunicação é de 256 mensagens (cada tarefa realiza 4 comunicação) cada uma com 1 dado (256 dados no total); em uma grade 2×2 com 4 tarefas (direita), apenas 16 comunicações são necessárias, cada uma com 4 dados para um total de 64 dados.

que tentam equalizar o peso das partições ao mesmo tempo que reduzem as bordas de comunicação. Essas técnicas permitem portanto o balanceamento da carga computacional e são rapidamente apresentadas na Seção 3.2.4 sobre Mapeamento.

3.2.4. Mapeamento

A quarta e última etapa da metodologia PCAM, chamado mapeamento, consiste em definir onde cada tarefa será executada. O mapeamento em si em um problema difícil pois precisa ser explícito em supercomputadores de alto desempenho. Inexiste um método automático para realizar o mapeamento, embora soluções simples possam ser aplicadas, sem que o desempenho seja o melhor possível. Os requisitos fundamentais na atividade explícita de mapeamento envolve (a) colocar tarefas concorrentes em unidades de processamento diferentes, de forma que tais tarefas sejam de fato executadas em paralelo e (b) alocar tarefas que se comunicam frequentemente em locais próximos na topologia de interconexão, tanto física quanto lógica. Estas duas condições podem entrar em conflito. Por exemplo, se considerarmos apenas a localidade podemos ser levados a colocar todas as tarefas em uma unidade de processamento, algo que certamente não é bom visto que as tarefas competiriam pelo mesmo recurso.

Em alguns casos a etapa de mapeamento é simples. Por exemplo, aplicações que possuem tarefas homogêneas entre si e ao longo da execução são apropriadas para mapear em supercomputadores com capacidade homogênea (todos as unidades de processamento são idênticas). Nestes casos, pode-se inclusive aglomerar as tarefas de maneira que tenhamos apenas uma tarefa por processador.

Por outro lado, a etapa de mapeamento se torna mais complexa quando as tarefas tem custos computacionais diferentes, ainda que estes custos sejam estáticos ao longo do tempo. Nestes cenários, algoritmos de *balanceamento de carga* podem ser úteis para equilibrar os custos entre os recursos computacionais. Métodos descentralizados de balanceamento de carga tem mais chance de se adaptar a evolução dos supercomputadores, especialmente no quesito de escalabilidade, pois não há uma única entidade controladora. O cenário mais complexo para a etapa de mapeamento ocorre quando a carga computacional é heterogênea tanto entre as tarefas quanto ao longo da execução, ou seja, o custo de uma tarefa evolui conforme a execução da aplicação avança. Neste caso, deve-se aplicar pre-

ferencialmente algoritmos de *balanceamento de carga dinâmicos*, capazes de monitorar a evolução da carga computacional ao longo do tempo. Algoritmos que exigem apenas um conhecimento local são preferíveis pois não requerem possíveis comunicações coletivas globais entre todas as tarefas.

Enfim, os algoritmos oriundos de decomposição funcional tem uma abordagem diferente de mapeamento. Eles podem ser mapeados preferencialmente por algoritmos de escalonamento de tarefas, antecipando tempo de ociosidade em processadores.

Balanceamento de carga. Algoritmos de balanceamento de carga são também conhecidos por algoritmos de particionamento. Eles tem por objetivo aglomerar tarefas finas (oriundas da etapa de particionamento) de uma partição inicial até encontrar uma tarefa cujo tamanho seja apropriado para uma determinada plataforma de execução. Existem quatro técnicas principais de balanceamento de carga: métodos baseados em bisseção recursiva, algoritmos locais, métodos probabilistas e mapeamento cíclicos.

Os métodos baseados em *bisseção recursiva* particionam o domínio do problema de maneira iterativa, com informações globais, sempre levando-se em conta o custo de um subdomínio e a minimização da comunicação entre as partições. Esses métodos são considerados algoritmos de divisão e conquista. Um exemplo é o algoritmo de Barnes-Hut [Barnes and Hut 1986]. Existem várias variantes dos métodos de bisseção recursiva. A forma mais simples, que não considera o custo e a quantidade das comunicações, consista em realizar a bisseção recursiva unicamente baseada nas coordenadas do domínio: sempre se divide a coordenada mais larga. Uma segunda variante do método de bisseção se chama de método desbalanceado pois tem um enfoque unicamente no controle das comunicações, reduzindo o perímetro das partições. Enfim, uma terceira variante mais sofisticada e mais geral é a bisseção recursiva de grafo, útil para grades não-estruturadas (veja Figura 3.4). Esta variante usa a informação de conectividade do grafo para reduzir o número de arestas que cruzam a fronteira entre dois subdomínios.

Uma técnica alternativa com menor intrusão consiste nos algoritmos de *balanceamento de carga locais*. Eles são relativamente baratos pois necessitam apenas de informações da tarefa em questão e de seus vizinhos. Isso possibilita também uma execução paralela, ou seja, todas as tarefas podem executar o algoritmo local simultaneamente. No entanto, a falta de coordenação global leva em geral a um particionamento pior daquele obtido por particionadores globais.

O terceiro tipo de método de balanceamento de carga consiste em *métodos probabilistas*. Com um custo baixo de execução e uma boa escalabilidade, ao mesmo tempo que ignora completamente o custo e quantidade de comunicações, algoritmos probabilistas alocam as tarefas nos recursos computacionais de maneira aleatória. Essa abordagem funciona melhor quando há muitas tarefas, pois as chances são maiores de fazer com que os recursos computacionais recebam carga de trabalho similar.

Enfim, os *mapeamentos cíclicos* são uma quarta forma de realizar o mapeamento da carga nos recursos computacionais. Baseado em um particionamento já definido na primeira etapa do método PCAM, a técnica distribui aos recursos, de maneira cíclica, as partições. Parte-se do princípio que existem bastante tarefas, que os custos com comunicação são baixos através de uma localidade de operações e dados reduzidas. Um mapeamento cí-

clico pode ser realizado utilizando como entrada os blocos de tarefas, criados na etapa de aglomeração.

Escalonamento de tarefas. Os algoritmos de escalonamento de tarefas podem ser usados em situações com requisitos de localidade fracos. Em geral, eles consideram as tarefas como um conjunto de problemas que devem ser resolvidos, sendo colocados em uma "piscina" de subproblemas. Uma heurística de escalonamento deve então decidir, durante a execução e de maneira dinâmica, em qual recurso computacional um determinado problema, recuperado da piscina, será alocado. O desenvolvimento de uma heurística que englobe de um lado a necessidade de reduzir os custos de comunicação e de outro o conhecimento global do sistema (para efetuar um bom balanceamento de carga) é o principal desafio. Embora heurísticas centralizadas tem um bom conhecimento da plataforma, em geral eles não são escaláveis para centenas de unidades de processamento. Para mitigar esse problema, existem heurísticas que criam uma estrutura hierárquica de gerenciadores, permitindo uma alternativa mais escalável com uma visão semi-global do estado da plataforma. Enfim, no outro extremo existem heurísticas totalmente descentralizadas: cada processador mantém uma "piscina" de tarefas e trabalhadores podem requisitar mais tarefas quando se tornam ociosos. Heurísticas probabilistas, potencialmente hierárquicas de acordo com a topologia da plataforma computacional, e associadas a roubo de tarefas se enquadram nesta classe de algoritmos.

3.3. Tendências Tecnológicas

Esta seção aborda as tendências tecnológicas modernas que moldam o presente e o futuro da área de processamento paralelo. Serão abordadas desdes conceitos estabelecidos, tais como computação de alto desempenho na nuvem, até conceitos mais inovadores como computação neuromórfica de alto desempenho.

3.3.1. Computação de Alto Desempenho na Nuvem

A computação em nuvem é uma forma de disponibilizar recursos de computação, como armazenamento, processamento, rede, software, entre outros, por meio da internet. Ela permite o acesso a uma infraestrutura escalável, flexível, econômica e segura para executar aplicações paralelas de alto desempenho. Alguns dos benefícios da computação em nuvem para a computação paralela de alto desempenho incluem a redução de custos operacionais e de manutenção; aumento da disponibilidade e confiabilidade; adaptação dinâmica à demanda e ao desempenho; acesso a tecnologias avançadas e inovadoras; entre outros. Alguns dos exemplos de serviços de computação em nuvem voltados para a computação paralela de alto desempenho disponíveis compreendem a *Amazon Web Services* (AWS), *Google Cloud Platform* (GCP), *Microsoft Azure*, *IBM Cloud*, entre outros.

Neste cenário, a computação de alto desempenho está sendo empregada como um serviço na nuvem (*HPCaaS*), conforme ilustrado na Figura 3.7. Ela mostra um exemplo de infraestrutura para execução de cargas de trabalho paralelas na nuvem, onde os usuários finais submetem a carga de trabalho através da *Internet* e o ambiente de nuvem (*HPCaaS*) é responsável por alocar máquinas e implantar a carga de trabalho para execução em sistemas computacionais de alto desempenho [Paillard et al. 2015][Navaux et al. 2023].

Um dos principais desafios associados à computação de alto desempenho em ambientes

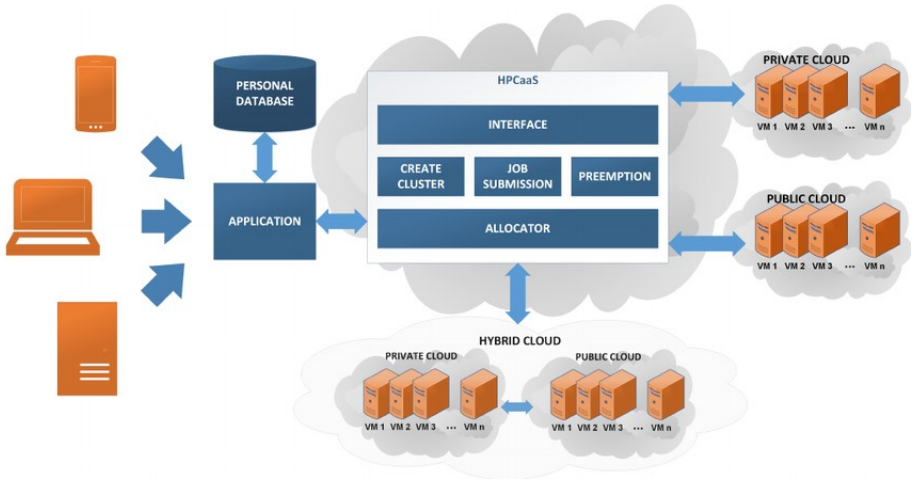


Figura 3.7. Computação de Alto Desempenho *as a service* na Computação na Nuvem [Paillard et al. 2015]

de nuvem está relacionado à eficiência do desempenho. Embora a nuvem ofereça escalabilidade em termos de recursos computacionais, a necessidade de virtualização e compartimentalização pode introduzir uma sobrecarga adicional, levando a alta latência e perda de desempenho. Consequentemente, a otimização das cargas de trabalho de HPC para uma execução eficaz em ambientes virtualizados exige ajustes e otimizações específicas para mitigar essa sobrecarga.

De maneira similar, a gestão de recursos é outra área desafiadora. Enquanto ambientes HPC tradicionais têm controle detalhado sobre os recursos do sistema, permitindo ajustes precisos para otimizar o desempenho, a natureza compartilhada da nuvem pode resultar em conflitos de recursos e competição entre diferentes instâncias de máquinas virtuais. Portanto, uma alocação e monitoramento eficazes dos recursos são e continuarão sendo essenciais para garantir que as cargas de trabalho HPC possam acessar a capacidade de computação necessária.

Dado que as aplicações HPC frequentemente dependem de comunicação intensiva entre nós de processamento, a latência da rede no contexto da nuvem pode também se tornar um problema crítico para o desempenho das aplicações. Nesse sentido, o projeto de uma arquitetura de rede de alto desempenho na nuvem, caracterizada por baixa latência e alta largura de banda, torna-se fundamental para atender às exigências de comunicação das aplicações HPC. Além disso, garantir que o processo de comunicação durante a migração de dados e cargas de trabalho para a nuvem não comprometa a privacidade dos usuários é um desafio adicional, especialmente considerando que as aplicações HPC frequentemente lidam com dados sensíveis ou confidenciais.

Uma tendência adicional é a proliferação significativa de provedores de computação em nuvem de alto desempenho. Consequentemente, a escolha de um provedor específico pode restringir a interoperabilidade e portabilidade das aplicações HPC, dificultando a transferência de cargas de trabalho entre diferentes nuvens ou para ambientes locais. Além disso, uma vez que as aplicações HPC normalmente são desenvolvidas para ambientes específicos, sua adaptação para aproveitar a escalabilidade e recursos da nuvem

pode se tornar um processo complexo e demorado devido à diversidade de recursos heterogêneos disponíveis.

Resumidamente, a integração da computação de alto desempenho em ambientes de nuvem oferece diversas oportunidades, mas também apresenta desafios substanciais. Otimização de desempenho, alocação eficaz de recursos, segurança sólida e adaptação de aplicações são elementos críticos que requerem abordagens cuidadosas para garantir o sucesso da computação de alto desempenho na nuvem. Além disso, à medida que a tecnologia evolui, soluções inovadoras e adaptativas continuarão a ser desenvolvidas para superar esses desafios e permitir que as aplicações HPC alcancem todo o seu potencial na nuvem.

3.3.2. Computação Quântica de Alto Desempenho

A computação quântica é uma forma de explorar os fenômenos da mecânica quântica para realizar operações que seriam impossíveis ou muito lentas em computadores clássicos. A computação quântica para alto desempenho é um campo que busca aproveitar as vantagens dos computadores quânticos para resolver problemas complexos e desafiadores que exigem muitos recursos computacionais. Os computadores quânticos são dispositivos que usam as propriedades da física quântica, como a superposição e o emaranhamento, para manipular unidades de informação chamadas *qubits*. Os *qubits* podem representar simultaneamente os valores 0 e 1, o que permite que os computadores quânticos realizem operações paralelas e explorem um espaço de soluções muito maior do que os computadores clássicos.

Ela tem a promessa de revolucionar diversos campos da ciência, tecnologia e negócios, como criptografia, otimização, aprendizado de máquina, química, física, medicina, entre outros. Embora ainda esteja em desenvolvimento, a computação quântica já vem mostrando avanços significativos e desafiando os limites da computação tradicional. Conforme destacado pelos autores em [Fu et al. 2016], um computador quântico irá sempre consistir de componentes de computação convencional e quântica pois algoritmos quânticos consistirão de partes clássicas e quânticas e, portanto, serão executadas por seus respectivos blocos de computação. Além disso, o computador quântico requer monitoramento muito próximo e, se necessário, correção pela lógica clássica. A Figura 3.8 ilustra uma visão geral da pilha do sistema de um computador quântico, onde as camadas superiores representam os algoritmos para os quais as linguagens e compiladores precisam ser desenvolvidos para que aplicações possam explorar o hardware quântico subjacente. Neste ponto, os *qubits* são definidos como *qubits* lógicos. A próxima camada é o conjunto de instruções da arquitetura (*Q Instruction Set Architecture* – QISA). Assim, o compilador irá traduzir instruções lógicas em instruções físicas que pertence a QISA. A camada de correção de erro (*Quantum Error Correction*) é responsável pela detecção e correção de erro, onde recebe dados para identificar possíveis erros e irá realizar as correções necessárias.

Existem vários desafios para desenvolver e implementar essa tecnologia, como a fragilidade dos *qubits*, a correção de erros, a escalabilidade, a programação e a integração com sistemas clássicos. Algumas empresas e instituições estão trabalhando para superar esses desafios e oferecer soluções de computação quântica para alto desempenho. Por exemplo, a IBM anunciou recentemente a criação de um processador quântico avançado chamado

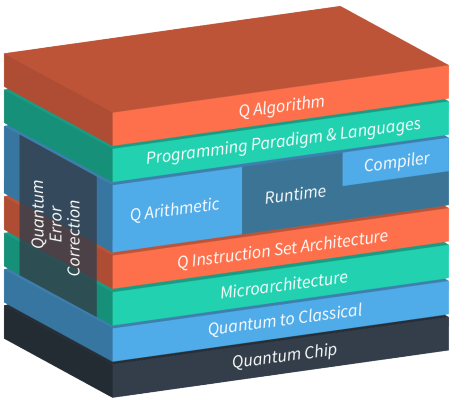


Figura 3.8. Visão geral da pilha do sistema de um computador quântico [Fu et al. 2016]

Eagle, que tem 127 *qubits* e ultrapassa o limite de processamento existente em máquinas quânticas [Chow et al. 2021]. A Microsoft oferece uma plataforma aberta chamada Azure Quantum, que permite aos usuários acessar diferentes tipos de hardware quântico e desenvolver algoritmos quânticos. Outras empresas que estão na vanguarda da computação quântica para alto desempenho são *Google*, *Amazon*, *Intel*, *Alibaba*, *Fujitsu* e *D-Wave*.

As interfaces de programação paralela que podem ser utilizadas para programar processadores quânticos dependem do tipo de hardware quântico e do modelo de computação quântica que se deseja usar. Existem diferentes paradigmas de computação quântica, como o modelo de circuitos quânticos, o modelo de computação adiabática, o modelo de máquinas de Turing quânticas e o modelo de computação topológica. Cada um desses modelos requer uma forma diferente de representar e manipular os *qubits* e os algoritmos quânticos. Assim, algumas das interfaces de programação paralela mais conhecidas e usadas para programar processadores quânticos são:

- *Q#*: É uma linguagem de programação quântica desenvolvida pela Microsoft, que faz parte do Quantum Development Kit[1]. O Q# é baseado no modelo de circuitos quânticos e permite a criação de programas quânticos híbridos, que combinam código quântico e clássico. O Q# também oferece ferramentas para simular, depurar, testar e otimizar os programas quânticos. Ele pode ser usado com o Azure Quantum, que é uma plataforma aberta para executar programas quânticos em diferentes tipos de hardware quântico.
- *Qiskit*: É um framework de software para programação quântica desenvolvido pela IBM, que faz parte do IBM Quantum Experience[2]. O *Qiskit* é baseado no modelo de circuitos quânticos e permite a criação de programas quânticos usando linguagens de programação como *Python* e *C++*. O *Qiskit* também oferece ferramentas para simular, visualizar, analisar e otimizar os programas quânticos. O Qiskit pode ser usado com o

[1]https://github.com/microsoft/Quantum
[2]https://www.ibm.com/quantum

IBM Quantum Cloud, que é uma plataforma que fornece acesso a diferentes tipos de hardware quântico.

- *Cirq*: É um framework de software para programação quântica desenvolvido pelo Google, que faz parte do Google Quantum AI[3]. O *Cirq* é baseado no modelo de circuitos quânticos e permite a criação de programas quânticos usando linguagens de programação como *Python*. Ele também oferece ferramentas para simular, depurar, testar e otimizar os programas quânticos. O *Cirq* pode ser usado com o Google Quantum Cloud, que é uma plataforma que fornece acesso a diferentes tipos de hardware quântico.

- Existem outras interfaces de programação paralela para programar processadores quânticos, como o *Quil* (desenvolvido pela Rigetti[4]), o *Ocean* (desenvolvido pela D-Wave[5]), o *Strawberry Fields* (desenvolvido pela Xanadu[6]) e o *ProjectQ* (desenvolvido pela ETH Zurich[7]). Cada uma dessas interfaces tem suas próprias características, vantagens e desvantagens, dependendo do tipo de problema que se quer resolver e do tipo de hardware quântico que se quer usar.

Portanto, a programação paralela para computadores quânticos apresentará desafios únicos à medida que essa tecnologia continua a avançar. Um desafio central será a complexidade inerente das operações quânticas, que envolvem estados superpostos e entrelaçados. Assim, a efetiva divisão e coordenação de tarefas entre qubits para realizar cálculos paralelos precisará levar em consideração as propriedades delicadas dos *qubits* e os requisitos específicos das operações quânticas. Além disso, a natureza não determinística dos computadores quânticos pode tornar a sincronização de processos paralelos mais desafiadora, exigindo novas abordagens para garantir resultados consistentes. A escalabilidade também será um fator limitante, já que a adição de mais *qubits* a um sistema pode aumentar as complexidades de comunicação e controle. Por fim, a heterogeneidade dos dispositivos quânticos disponíveis, com diferentes tempos de coerência e taxas de erro, exigirá estratégias de programação paralela adaptativas e otimizadas para garantir o melhor desempenho em um ambiente quântico.

3.3.3. Computação Neuromórfica de Alto Desempenho

A computação neuromórfica é uma forma de inspirar-se na estrutura e no funcionamento do cérebro humano para projetar e construir sistemas de computação paralela de alto desempenho. Ela tem o objetivo de criar dispositivos que possam aprender, adaptar-se e interagir com o ambiente de forma autônoma e eficiente. A computação neuromórfica também busca superar as limitações dos sistemas convencionais em termos de consumo de energia, latência, precisão, robustez, entre outros. Alguns dos exemplos de aplicações da computação neuromórfica incluem reconhecimento de padrões, visão computacional, processamento de linguagem natural e controle robótico. As plataformas de computação

[3]https://quantumai.google/
[4]https://github.com/quil-lang/quil
[5]https://www.dwavesys.com/solutions-and-products/ocean/
[6]https://strawberryfields.ai/
[7]https://github.com/ProjectQ-Framework/ProjectQ

neuromórfica têm evoluído nos últimos anos. Exemplos destas arquiteturas incluem *Intel Loihi* [Davies et al. 2018], *IBM TrueNorth* [Akopyan et al. 2015] e *SpiNNaker* [Mayr et al. 2019], conforme discutidos a seguir.

O processador *Intel Loihi 2* é uma evolução do *Loihi*, com a promessa de ser até 10 vezes mais rápido e mais econômico em energia. Além disso, o *Loihi 2* foi fabricado com a tecnologia de 4 nanômetros da Intel, sendo um dos primeiros chips a usar esse nó de produção. Este processador possui 128 núcleos de processamento neuromórficos com 8x mais neurônios e sinapses que a versão anterior. Cada núcleo tem 192KB de memória flexível e o neurônio pode ser completamente programável, assim como um FPGA. O processador *IBM TrueNorth* é composto por 4096 núcleos, cada um contendo 256 neurônios artificiais e 256 milhões de sinapses. Esses elementos são interconectados por uma infraestrutura de roteamento baseada em eventos. O *TrueNorth* foi fabricado com a tecnologia de 28 nanômetros da IBM, e consome apenas 65 miliwatts de energia.

A Figura 3.9 destaca a arquitetura *TrueNorth* [Akopyan et al. 2015], inspirada pela estrutura e função do (*a*) cérebro humano e (*b*) coluna cortical, uma pequena região de neurônios densamente interconectados e funcionalmente relacionados que formam a unidade canônica do cérebro. De forma análoga, o (*c*) núcleo neurossináptico é o bloco básico de construção da arquitetura *TrueNorth*, contendo 256 axônios de entrada, 256 neurônios e uma barra transversal sináptica de 64k. Composto por computação fortemente acoplada (neurônios), memória (sinapses) e comunicação (axônios e dendritos), um núcleo ocupa $240x390\mu m$ de área de sílicio (*d*). 4096 núcleos neurossinápticos organizados em uma matriz 2-D forma um chip *TrueNorth* (*e*), que ocupa $4,3cm^2$ em um processo CMOS de 28nm e consome apenas 65mW ao executar uma aplicação típica de visão computacional (*f*). O resultado é uma aproximação eficiente da estrutura cortical dentro das restrições de um subtrato de silício (*g*).

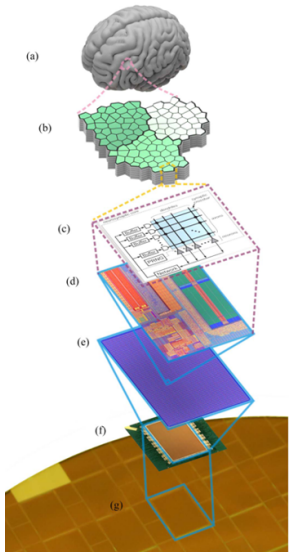


Figura 3.9. Arquitetura TrueNorth, inspirada pela estrutura e funções do cérebro humano [Akopyan et al. 2015]

O processador *SpiNNaker* é uma plataforma para computação massivamente paralela. Ele é composto por meio milhão de elementos computacionais simples, que imitam as

sinapses, controlados pelo seu próprio software. Ele pode realizar até 200 milhões de ações por segundo e tem mais 100 milhões de peças móveis. Durante o seu projeto, três dos axiomas do projeto de máquinas paralelas (coerência de memória, sincronicidade e determinismo) foram descartados sem comprometer a capacidade de realizar cálculos significativos.

Para programar em paralelo para esses processadores, é preciso levar em conta algumas características e técnicas específicas, como por exemplo, o uso de linguagens de programação funcionais que permitem expressar algoritmos de forma mais abstrata e declarativa, facilitando a paralelização neste tipo de arquitetura; estruturas de dados imutáveis para garantir que os dados não serão modificados por outros processos paralelos, evitando problemas de sincronização e consistência; e modelos de programação específicos para processadores neuromórficos.

Estes modelos de programação específicos oferecem abstrações e construções adequadas para processadores neuromórficos. Por exemplo, a linguagem *PyNN* (Python for Neural Networks [Davison et al. 2009]) permite definir modelos de neurônios e sinapses, criar redes neurais e executá-las em diferentes plataformas neuromórficas, como o *SpiNNaker*. A linguagem *Nengo*[8] também permite criar e simular redes neurais em arquiteturas neuromórficas, usando um paradigma baseado em fluxo de dados.

Outra maneira de programar paralelo para arquiteturas neuromórficas é usar bibliotecas ou *frameworks* que facilitam a integração entre as aplicações e as plataformas neuromórficas. Por exemplo, o framework *NeuCube* permite desenvolver aplicações de aprendizado de máquina baseadas em redes neurais esparsas, que podem ser executadas em arquiteturas neuromórficas como o *TrueNorth* [Kasabov 2014]. O framework *NeuGen* permite gerar modelos detalhados de circuitos neuronais, que podem ser simulados em arquiteturas neuromórficas como o *Neurogid* [Eberhard et al. 2006].

No entanto, a programação paralela para computadores neuromórficos apresenta desafios distintos à medida que essa tecnologia evolui. Um desafio central reside na tradução eficiente de algoritmos tradicionais para modelos neuroinspirados, aproveitando a capacidade desses sistemas de processar informações de maneira semelhante ao cérebro humano. A programação eficaz também exigirá a exploração das arquiteturas altamente paralelas e distribuídas desses sistemas, garantindo que a computação ocorra de maneira otimizada e que as interconexões complexas entre neurônios artificiais sejam devidamente coordenadas. Além disso, lidar com a variabilidade inerente dos componentes neuromórficos e as taxas de falha associadas requer estratégias de tolerância a falhas e adaptação dinâmica. A coexistência de múltiplos tipos de neurônios e sinapses em um único chip também exige uma alocação eficiente de recursos e uma programação que leve em conta as características específicas dos diferentes elementos. Em última análise, a programação paralela bem-sucedida para computadores neuromórficos dependerá da colaboração entre a comunidade de neurociência computacional e a comunidade de ciência da computação para criar abordagens que aproveitem totalmente o potencial desses sistemas inovadores.

[8]https://www.nengo.ai/

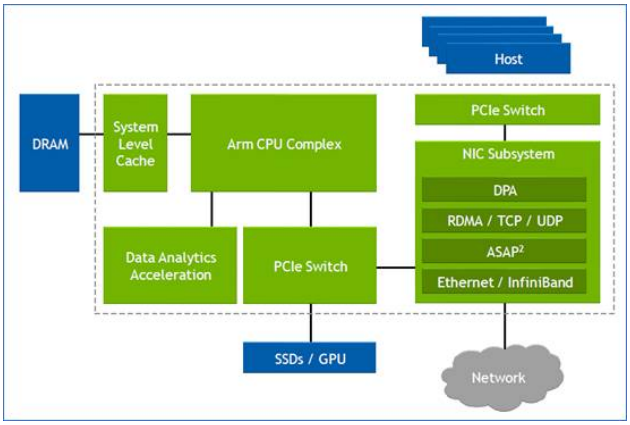


Figura 3.10. Diagrama de blocos da DPU Bluefield-3 da NVidia [Burstein 2021]

3.3.4. Computação Paralela em Adaptadores de Rede Inteligentes

Os adaptadores de rede inteligentes, também conhecidos como *SmartNICs*, são dispositivos que possuem unidades de processamento de dados (DPUs – *data processing units*) integradas capazes de executar funções de rede e segurança no próprio hardware, sem depender da CPU do sistema. Estes dispositivos têm sido bastante úteis para cenários de computação de alto desempenho, como ambiente de nuvem, aprendizado de máquina e *big data*.

Diferentes são as maneiras de empregar *SmartNICs* para computação de alto desempenho. Uma delas consiste no uso das DPUs para executar funções de rede e segurança no hardware. Outra forma consiste em usar as tecnologias de transferência direta de dados entre os dispositivos de armazenamento e as GPUs, como RoCE e GPUDirect Storage[9]. Essas tecnologias permitem que os dados sejam processados em paralelo pelas GPUs, sem passar pela CPU ou pela memória do sistema, reduzindo a latência, o consumo de energia e a complexidade do sistema. Além disso, os *SmartNICs* podem fornecer um serviço de sincronização extremamente preciso para as aplicações de centro de dados e infraestrutura subjacente. Isso pode facilitar a coordenação e a comunicação entre os processos paralelos que dependem de sincronização de tempo, como os algoritmos distribuídos e os sistemas de consenso.

A Figura 3.10 exemplifica os componentes de *hardware* do adaptador de rede inteligente *Bluefild-3* (em verde), da NVIDIA. Este adaptador está conectado à internet (*Network*) e a máquina *host*. Adicionalmente, ele possui conexões diretas com a memória DRAM e unidades de armazenamento para acelerar o acesso aos dados sem a necessidade de interferência da CPU do *host*. Esta arquitetura possui 16 núcleos ARM-A78 (*ARM CPU Complex*) e aceleradores de dados com 16 núcleos e 256 *threads* (*Data Analytics Acceleration*). Deste modo, para programar em paralelo para *SmartNICs* tais como *Bluefield-3*, é preciso conhecer os recursos e as limitações das unidades de processamento de dados que eles possuem, bem como as bibliotecas e as ferramentas que permitem acessar e controlar essas DPUs. Assim, diferentes bibliotecas surgem como alternativas:

[9]https://docs.nvidia.com/gpudirect-storage/overview-guide/index.html

- ***NVIDIA DOCA***[10] é um framework de desenvolvimento de software para *SmartNICs* baseados em DPUs NVIDIA BlueField. Ele oferece uma interface de programação para criar e gerenciar aplicações de rede e segurança que executam nas DPUs. Ele também fornece um ambiente de execução (*runtime*) que abstrai os detalhes de baixo nível do hardware e do sistema operacional.
- ***Mellanox SmartNIC SDK***[11] é um kit de desenvolvimento de software para *SmartNICs* baseados em DPUs *Mellanox ConnectX*. Ele permite o desenvolvimento de aplicações personalizadas que executam nas DPUs, usando linguagens como C, C++ e Python. Ele também inclui exemplos de código, documentação e ferramentas de depuração.
- ***Intel Data Plane Development Kit (DPDK)*** é um conjunto de bibliotecas e *drivers* para acelerar o processamento de pacotes em plataformas baseadas em processadores Intel. Ele pode ser usado para programar *SmartNICs* baseados em DPUs Intel I/O Processing Units (IOPUs), usando linguagens como C e *Rust*. Ele também oferece suporte a várias arquiteturas de rede, como memória compartilhada, troca de mensagens e RDMA [Zhu 2020].

Um desafio central da programação paralela em *SmartNICs* é a exploração eficiente das capacidades de processamento altamente paralelo desses controladores, que são projetados para acelerar funções de rede e processamento de dados em níveis próximos ao hardware. A programação precisa encontrar o equilíbrio entre a distribuição de tarefas em vários núcleos de processamento e a otimização do uso dos recursos específicos do *SmartNIC* para obter um desempenho máximo. Além disso, a coordenação entre os núcleos de processamento e as unidades de rede especializadas exige abordagens eficazes para evitar gargalos de comunicação e latência excessiva. Lidar com a heterogeneidade de cargas de trabalho e requisitos variados de aplicativos também é um desafio, visto que diferentes cenários de uso demandarão estratégias de programação personalizadas. A garantia de segurança e isolamento, especialmente quando várias funções de rede são executadas no mesmo dispositivo, também será crucial. Em última análise, a programação paralela bem-sucedida para *SmartNICs* dependerá da compreensão profunda da arquitetura desses dispositivos e da adaptação das técnicas de programação existentes para tirar o máximo proveito de suas capacidades específicas

3.3.5. Computação de Alto Desempenho em Arquiteturas Massivamente Paralelas

Arquiteturas massivamente paralelas são projetos de sistemas de computação que se destacam pela capacidade de executar um grande número de tarefas ou instruções simultaneamente, aproveitando um grande conjunto de unidades de processamento interconectadas. Essa abordagem visa a obtenção de um alto nível de desempenho computacional para lidar com cargas de trabalho intensivas e complexas, como simulações científicas, análise de grandes conjuntos de dados e processamento de inteligência artificial.

Essas arquiteturas se diferenciam das abordagens convencionais, que contam com um ou alguns poucos núcleos de processamento central. Em vez disso, as arquiteturas massivamente paralelas incorporam centenas ou mesmo milhares de núcleos de processamento

[10] https://developer.nvidia.com/networking/doca

[11] https://docs.nvidia.com/networking/display/NEOSDKv25

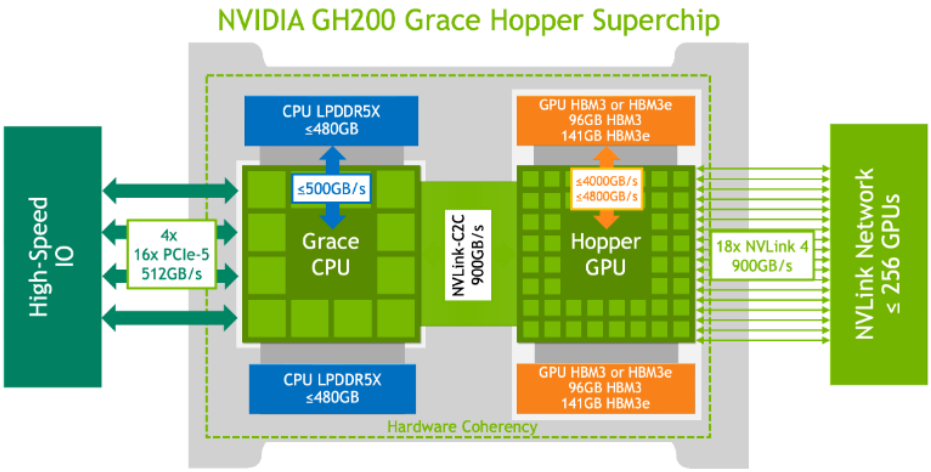


Figura 3.11. Superchip Grace Hopper da NVIDIA

independentes, trabalhando em conjunto para acelerar a execução de tarefas. Um exemplo notável é o uso de unidades de processamento gráfico (GPUs - *graphical processing units*), projetadas originalmente para renderização gráfica. A arquitetura altamente paralela dessas unidades se mostrou extremamente eficaz em tarefas de computação intensiva, como aprendizado de máquina e simulações científicas. Logo, é natural que diferentes empresas têm voltado seus projetos para o desenvolvimento de arquiteturas massivamente paralelas, como o caso da NVIDIA, Intel e AMD.

Recentemente, a NVIDIA projetou a arquitetura *GH200 Grace Hopper*. Ela consiste de uma CPU acelerada projetada do zero para computação de alto desempenho e inteligência artificial. Conforme ilustrado na Figura 3.11[12], ela combina as arquiteturas *Grace* e *Hopper* usando a tecnologia *NVIDIA NVLink-C2C* para oferecer um modelo de memória coerente de CPU e GPU com alta largura de banda (até 900GB/s). Considerando estações de trabalho que demandam alto desempenho gráfico, a Intel projetou a família de GPUs Intel Arc, com suporte a *Ray Tracing* acelerado por *hardware*, codificação de vídeo com o *codec* AV1 e outras características. De modo similar, a GPU AMD ROCm é uma plataforma de software aberto para programação de GPUs, suportando ambientes em vários fornecedores e arquiteturas de aceleradores. Ela oferece suporte às principais estruturas de aprendizado de máquina para ajudar os usuários a acelerar as cargas de trabalho de IA.

No entanto, aproveitar ao máximo essa capacidade computacional disponível requer uma mudança na forma como os programas são projetados e implementados. A programação paralela nestas arquiteturas massivamente paralelas visa aproveitar as vantagens de cada dispositivo para resolver problemas complexos de forma eficiente e escalável. Assim, diversas interfaces de programação permitem o desenvolvimento de aplicações paralelas para tais arquiteturas, cada uma com suas próprias características, vantagens e desvantagens. Algumas das mais populares são:

- *CUDA*: É uma linguagem baseada em C/C++ que permite a programação direta de GPUs da NVIDIA. É uma linguagem de baixo nível que oferece um controle fino sobre

[12]https://resources.nvidia.com/en-us-grace-cpu/nvidia-grace-hopper

os recursos da GPU, mas também exige um conhecimento profundo da arquitetura e do modelo de execução.

- *OpenCL*: É uma linguagem baseada em C/C++ que permite a programação de diversos tipos de dispositivos, como CPUs, GPUs, FPGAs, entre outros. É uma linguagem de baixo nível que oferece flexibilidade e portabilidade, mas também exige um cuidado maior com a compatibilidade e a otimização do código para cada dispositivo.
- *OpenMP*: É uma interface de programação paralela para C/C++ que permite a programação paralela por meio da inserção de diretivas e funções que controlam a criação e a sincronização de threads em um código sequencial. Diferentemente de *OpenCL* e *CUDA*, ela visa exploração de paralelismo objetivando simplicidade e produtividade.
- *OpenACC*: É uma extensão da linguagem C/C++ que permite a programação paralela por meio de diretivas que indicam quais regiões do código devem ser executadas em arquiteturas multicore e dispositivos aceleradores, como GPUs e FPGAs. Assim com o *OpenMP*, é uma linguagem de alto nível que oferece abstração e portabilidade, mas também depende da qualidade do compilador para gerar código eficiente para cada dispositivo.
- *Intel OneAPI*: é um modelo de programação unificado, baseado em padrões abertos, que permite o desenvolvimento de aplicações que podem ser executadas em diferentes tipos de arquiteturas aceleradoras, como CPUs, GPUs e FPGAs. Seu objetivo é oferecer uma experiência de desenvolvimento que aumenta a produtividade, desempenho e inovação dos desenvolvedores.

3.3.6. Computação de Alto Desempenho em Processadores de Inteligência Artificial

Nos últimos anos, avanços notáveis na convergência entre a IA e HPC tem resultado em uma simbiose promissora que impulsiona a fronteira da capacidade computacional. Assim, o uso de processadores otimizados para IA em ambientes de HPC está se tornando uma tendência proeminente, trazendo consigo a promessa de acelerar não apenas as cargas de trabalho tradicionais de HPC, mas também possibilitando abordagens inovadoras para problemas complexos. Esses processadores, muitas vezes equipados com unidades especializadas de hardware para tarefas intensivas de IA, como treinamento e inferência de redes neurais profundas, demonstraram a capacidade de lidar com a crescente demanda por processamento de dados complexos em campos que variam desde a pesquisa científica até a análise de grandes conjuntos de dados. Deste modo, esta fusão entre IA e HPC promete catalisar a descoberta, resolução de problemas complexos e tomada de decisões mais informadas, moldando o futuro da computação de alto desempenho.

Diferentes arquiteturas focadas em IA têm sido propostas ao longo dos anos, conforme discutido a seguir. O processador SambaNova é baseado em uma arquitetura chamada *Reconfigurable Dataflow Unit* (RDU), que consiste em um arranjo de unidades aritméticas que podem se comunicar entre si de forma assíncrona e adaptativa, sem a necessidade de um barramento ou uma memória compartilhada . Essa arquitetura permite que o processador se adapte dinamicamente às características e aos requisitos das diferentes aplicações de inteligência artificial, otimizando o uso dos recursos e o consumo de energia. O processador SambaNova pode ser usado tanto para treinamento quanto para inferência de

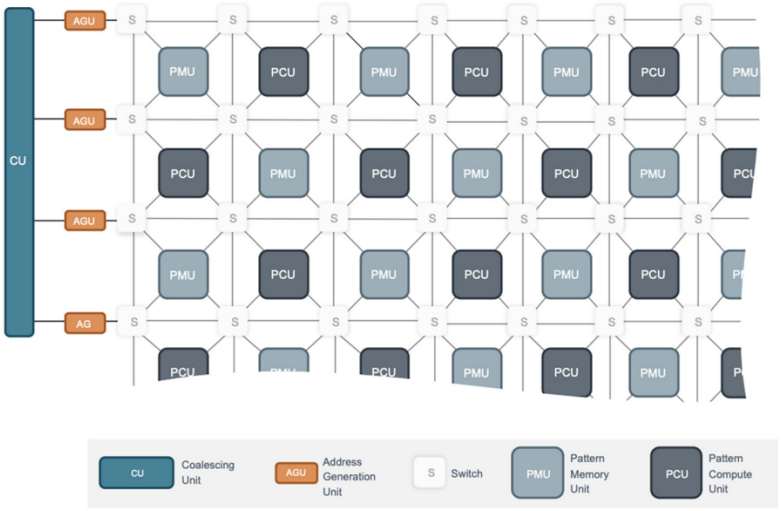


Figura 3.12. Reconfigurable Dataflow Unit (RDU) do sistema SambaNova [Emani et al. 2021]

modelos de inteligência artificial, sendo capaz de executar redes neurais profundas com milhões ou bilhões de parâmetros.

A Figura 3.12 destaca uma pequena parte da arquitetura RDU [Emani et al. 2021]. Ela consiste de uma matriz de unidades de processamento e memória reconfiguráveis conectadas por uma malha de comutação 3-D. Quando uma aplicação inicia a execução, o *software SambaFlow* configura os elementos RDU para executar um fluxo de dados otimizado para a aplicação. A PCU (unidade de computação padrão) foi projetada para executar uma única operação paralela de uma aplicação. Seu caminho de dados é organizado como um pipeline SIMD reconfigurado de vários estágios. A PMU (unidade de memória padrão) consiste de *scrathpads* especializadas que fornecem capacidade de memória e executam uma série de funções específicas para minimizar a movimentação de dados, reduzir a latência e fornecer alta largura de banda. O componente de *Switch* (S) é uma estrutura de alta velocidade que conecta PCUs e PMUs, sendo composta por três redes de comutação: escalar, vetorial e de coontrole. Por fim, as unidades geradoras de endereço (AGU) e unidades coalescentes (CU) fornecem a interconexão entre RDUs e o resto do sistema, incluindo a memória DRAM, outras RDUs, e o processador *host* para o processamento eficiente de problemas maiores.

Cerebras Systems é um processador baseado em uma arquitetura chamada *Wafer Scale Engine* (WSE), que consiste em um único chip que ocupa toda a superfície de um *wafer* de silício. Um *wafer* é um disco de silício usado para fabricar vários chips menores, mas o processador Cerebras Systems usa o wafer inteiro como um chip gigante, com trilhões de transistores e milhares de núcleos otimizados para IA. Outras soluções incluem o processador *Graphcore* e *Groq*. *Graphcore* é baseado em uma arquitetura chamada *Intelligence Processing Unit* (IPU), que consiste em um chip que contém milhares de núcleos de processamento e uma grande quantidade de memória interna [Knowles 2021]. Já o processador *Groq* é baseado em uma arquitetura chamada Tensor Streaming Processor (TSP), que consiste em um chip que contém milhares de unidades de processamento e

uma grande quantidade de memória interna [Gwennap 2020].

As interfaces de programação paralela que podem ser utilizadas para programar tais processadores são as seguintes:

- *SambaNova* oferece uma plataforma de software chamada *SambaFlow*, que é um conjunto de ferramentas e bibliotecas para desenvolver, treinar e implantar modelos de aprendizado de máquina em seu sistema *DataScale*. O *SambaFlow* suporta linguagens de programação como Python, C++ e Java, e frameworks de aprendizado de máquina como *TensorFlow*, *PyTorch* e *MXNet*. Ele também permite a integração com outras ferramentas de software, como *Kubeflow*, *Spark* e *Ray*.

- *Cerebras* oferece uma plataforma de software chamada *Cerebras Software Platform*, um ambiente integrado para desenvolver, treinar e executar modelos de aprendizado de máquina em seu sistema *CS-2*. Esta plataforma suporta linguagens de programação como Python e C++, e *frameworks* de aprendizado de máquina como *TensorFlow*, *PyTorch* e *JAX*. O Cerebras Software Platform também permite a integração com outras ferramentas de software, como *Horovod*, *MPI* e *NCCL*.

- *GraphCore* oferece uma plataforma de software chamada *Poplar*, um framework gráfico para desenvolver, treinar e implantar modelos de aprendizado de máquina em seu sistema IPU. O Poplar suporta linguagens de programação como Python e C++, e frameworks de aprendizado de máquina como *TensorFlow*, *PyTorch* e *ONNX*. O *Poplar* também permite a integração com outras ferramentas de software, como *Kubeflow*, *Spark* e *Dask*.

- *Groq* oferece uma plataforma de software chamada *Groq SDK*, que suporta linguagens de programação como Python e C++, e frameworks de aprendizado de máquina como TensorFlow, PyTorch e ONNX. O *Groq SDK* também permite a integração com outras ferramentas de software, como *TVM*, *MLIR* e *Glow*.

3.4. Conclusão e Reflexões Futuras

O processamento paralelo é uma peça fundamental no desenvolvimento técnico-científico pois é necessário para resolver problemas em inúmeras áreas da ciência e da tecnologia. Torna-se portanto fundamental criar intuições a partir do pensamento computacional paralelo e da programação paralela. Este minicurso procurou trazer as peças básicas necessárias para incutir no leitor os reflexos necessários para projetar soluções paralelas eficientes, além de trazer uma moderno apanhado de tendências tecnológicas que moldam o futuro da área. Espera-se que com esta combinação de conceitos básicos e tendências o leitor possa ser motivado a trabalhar nesta enriquecedora e multidisciplinar área de pesquisa, com problemas concretos e de alto impacto na sociedade.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

[Akopyan et al. 2015] Akopyan, F., Sawada, J., Cassidy, A., Alvarez-Icaza, R., Arthur, J., Merolla, P., Imam, N., Nakamura, Y., Datta, P., Nam, G.-J., Taba, B., Beakes, M., Brezzo, B., Kuang, J. B., Manohar, R., Risk, W. P., Jackson, B., and Modha, D. S. (2015). Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(10):1537–1557.

[Barnes and Hut 1986] Barnes, J. and Hut, P. (1986). A hierarchical o (n log n) force-calculation algorithm. *nature*, 324(6096):446.

[Burstein 2021] Burstein, I. (2021). Nvidia data center processing unit (dpu) architecture. In *2021 IEEE Hot Chips 33 Symposium (HCS)*, pages 1–20.

[Chow et al. 2021] Chow, J., Dial, O., and Gambetta, J. (2021). Ibm quantum breaks the 100-qubit processor barrier. *IBM Research Blog*, 2.

[Davies et al. 2018] Davies, M., Srinivasa, N., Lin, T.-H., Chinya, G., Cao, Y., Choday, S. H., Dimou, G., Joshi, P., Imam, N., Jain, S., et al. (2018). Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1):82–99.

[Davison et al. 2009] Davison, A. P., Brüderle, D., Eppler, J. M., Kremkow, J., Muller, E., Pecevski, D., Perrinet, L., and Yger, P. (2009). Pynn: a common interface for neuronal network simulators. *Frontiers in neuroinformatics*, 2:388.

[Eberhard et al. 2006] Eberhard, J. P., Wanner, A., and Wittum, G. (2006). Neugen: a tool for the generation of realistic morphology of cortical neurons and neural networks in 3d. *Neurocomputing*, 70(1-3):327–342.

[Emani et al. 2021] Emani, M., Vishwanath, V., Adams, C., Papka, M. E., Stevens, R., Florescu, L., Jairath, S., Liu, W., Nama, T., and Sujeeth, A. (2021). Accelerating scientific applications with sambanova reconfigurable dataflow architecture. *Computing in Science & Engineering*, 23(2):114–119.

[Foster 1995] Foster, I. (1995). *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Fu et al. 2016] Fu, X., Riesebos, L., Lao, L., Almudever, C. G., Sebastiano, F., Versluis, R., Charbon, E., and Bertels, K. (2016). A heterogeneous quantum computer architecture. In *Proceedings of the ACM International Conference on Computing Frontiers*, pages 323–330.

[Gwennap 2020] Gwennap, L. (2020). Groq rocks neural networks. *Microprocessor Report, Tech. Rep., jan*.

[Kasabov 2014] Kasabov, N. K. (2014). Neucube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. *Neural Networks*, 52:62–76.

[Knowles 2021] Knowles, S. (2021). Graphcore. In *2021 IEEE Hot Chips 33 Symposium (HCS)*, pages 1–25.

[Mayr et al. 2019] Mayr, C., Höppner, S., and Furber, S. B. (2019). Spinnaker 2: A 10 million core processor system for brain simulation and machine learning. *CoRR*, abs/1911.02385.

[Navaux et al. 2023] Navaux, P. O. A., Lorenzon, A. F., and Serpa, M. d. S. (2023). Challenges in high-performance computing. *Journal of the Brazilian Computer Society*, 29(1):51â€“62.

[Paillard et al. 2015] Paillard, G. A. L., Coutinho, E. F., de Lima, E. T., and Moreira, L. O. (2015). An architecture proposal for high performance computing in cloud computing environments. In *4th International Workshop on ADVANCEs in ICT INfrastructures and Services (ADVANCE 2015), Recife*.

[Zhu 2020] Zhu, H. (2020). *Data Plane Development Kit (DPDK): A Software Optimization Guide to the User Space-Based Network Applications*. CRC Press.

Capítulo 4

Desafios para a Computação Energeticamente Eficiente

Luigi Carro e Gabriel Luca Nazar

Programa de Pós-Graduação em Computação (PPGC) - UFRGS

Abstract

New and relevant applications impose great challenges to computational systems, since they demand complex processing applied to large and growing amounts of data. With the weakening of Moore's law seen in recent years, new approaches are needed to deliver the required performance. An additional challenge is the need to contain the growing energy consumption of computing devices and infrastructures to reduce operating costs, for environmental reasons and, for mobile devices, also due to battery limitations. In this chapter, we will address the challenges of developing such systems and possible solutions, traversing the multiple levels of abstraction in computing.

Resumo

Novas e relevantes aplicações impõem grandes desafios a sistemas computacionais, já que demandam processamento complexo aplicado a grandes, e crescentes, quantidades de dados. Com a perda de força da lei de Moore observada nos últimos anos, novas abordagens são necessárias para oferecer o desempenho necessário. Soma-se a esse desafio a necessidade de contermos o crescente consumo energético de dispositivos e infraestruturas computacionais para redução de custos operacionais, por questões ambientais e, em dispositivos móveis, também por limitações de bateria. Nesse capítulo, iremos abordar os desafios de desenvolvimento de tais sistemas e possíveis soluções, atravessando os múltiplos níveis de abstração da computação.

Vídeo com a apresentação do capítulo: `https://youtu.be/OsXT15-deO8`

4.1. Introdução

Enquanto a lei de Moore esteve presente, as aplicações da informática floresceram, de tal modo que a vida cotidiana hoje em dia é fortemente baseada em algum tipo de iteração com computadores, ocultos ou explícitos. A escalada tecnológica (ou a lei de Moore) permitiu a integração de mais dispositivos em um único circuito integrado, com benefícios em velocidade de operação e energia. À medida que a lei de Moore perde a força, para que a evolução permanente da Computação se mantenha, deve-se encontrar um substituto para esta melhoria tecnológica, que forneça as mesmas vantagens. Isto significa que, à medida que o problema aumenta, é preciso permitir a escalabilidade do hardware para resolver um problema maior em um tempo constante, ou para resolver o mesmo problema em um período de tempo menor, sem alterar a base do software ou mesmo exigir modificações significativas no software.

Apenas fornecer mais hardware para execução paralela, por exemplo, claramente não é a solução escalável que se precisa. O paralelismo não favorece energia, apenas EDP, ou o produto de energia vezes tempo de execução (ou seja, executando mais rápido com a mesma energia gasta, ou gastando menos energia no mesmo tempo de execução). Além disso, são necessárias modificações severas ou mesmo radicais no software para que o paralelismo possa ser explorado em diferentes aplicações.

As aplicações atuais do domínio de Computação são caracterizadas pelo seu enorme tamanho, medido em milhões de linhas de código, como apresentado na Tabela 4.1. A transposição dessas aplicações para uma versão paralela é uma tarefa mais do que hercúlea. Sem o impulso da tecnologia, como a comunidade poderá sustentar a crescente demanda por mais desempenho?

Tabela 4.1. Tamanho, em linhas de código fonte, de diferentes aplicação. Fonte: [Desjardins 2017]

Aplicação	Milhões de linhas de código
Linux kernel 2.2.0	2,0
Windows 3.1 (1992)	2,3
Drone militar dos EUA	3,5
Photoshop C.C.6	4,5
HealthCare.gov	5,0
Google Chrome	6,0
Boeing 787, somente aviônica e sistemas de suporte on-line	6,5
Chevy Volt	10,0
Android	11,5
Boeing 787, total	14,0
Caça F-35	24,0
Carro de alto padrão	100,0

Ao mesmo tempo em que se vivencia uma crise tecnológica, pela ausência da lei de Moore como todo o setor de Ciência da Computação estava acostumado pelos últimos 40 anos, tem-se outro problema real e premente: o consumo energético excessivo, tanto

de aplicações em nuvem quanto de aplicações portáteis. Muitas vezes estes domínios aparecem na mesma aplicação. Por exemplo, um estudo recente mostra que se todos os usuários do Google usassem seu serviços de reconhecimento de voz por apenas 3 minutos por dia, toda a capacidade computacional de seus servidores teria de dobrar [Jones 2018]. Mundialmente, ao redor de 12% da energia total é gasta em datacenters e computadores pessoais, celulares e TVs, e este número tende a subir até 20,9% até 2030 [Jones 2018].

Do ponto de vista da Ciência da Computação, para reduzir os custos de datacenters e para aumentar a duração da bateria de celulares, tem-se de focar, neste momento tecnológico, no desenvolvimento de soluções de baixa energia e alto desempenho. Isto porque, embora o problema de consumo seja muito importante, o suporte que a evolução do hardware forneceu nos últimos 40 anos não estará mais presente, pelo fim da lei de Moore, como mencionado acima.

Como mostra a Figura 4.1, as camadas de abstração que usamos nos últimos 50 anos ainda estão presentes. Em alguns domínios, ao atravessar essas camadas, resultados favoráveis foram obtidos. Por exemplo, a síntese de alto nível (HLS) [Nane et al. 2016, Coussy et al. 2009] é uma transformação da camada do programa para a camada lógica, cada vez mais popular para desenvolvimento para dispositivos como *Field-Programmable Gate Arrays* (FPGAs). Infelizmente, embora a HLS tenha existido durante anos, ainda existem entraves, como a necessidade de inclusão manual de diretivas de otimização para obter implementações de melhor qualidade [AMD 2023a, Cong et al. 2022], que limitam sua aplicabilidade e produtividade em projetos de larga escala. Fazer caches visíveis para o programador também já foi usado como uma estratégia de cruzar camadas no domínio da computação de alto desempenho, para conseguir mais *hits* de cache e aumentar o desempenho, ao mesmo tempo em que se reduz a dissipação de energia. Isto, no entanto, coloca mais pressão sobre a habilidade do programador de entender as camadas mais baixas da pilha de abstração, com severas penalidades em tempo de projeto e custos de software.

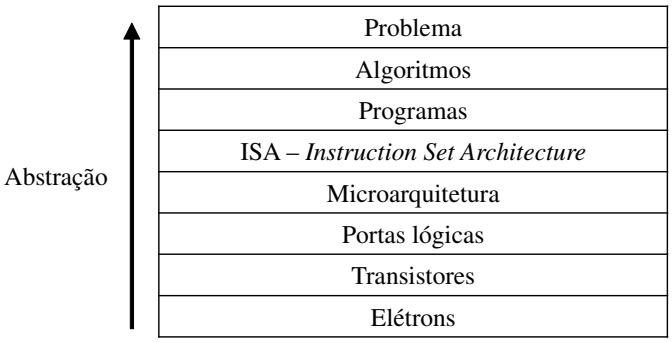


Figura 4.1. Níveis de abstração.

A única maneira de se sustentar a lei de Moore sem tecnologia, e incluindo a eficiência energética tão necessária hoje em dia, passa pela transformação mais eficaz de algoritmos em hardware, atravessando várias camadas de abstração de forma automática. A idéia central é manter-se as abstrações atuais de desenvolvimento de software, ao mesmo tempo em que se fornecem os mesmos efeitos que a evolução da tecnologia traria

ao cenário, explorando habilmente o desenvolvimento de hardware e transformações de software entre as camadas da pilha de abstração.

O que se propõe neste curso é mostrar que é possível o desenvolvimento, para o domínio de aplicações caracterizadas pelo uso intensivo de dados com e sem localidade, de uma mistura de soluções de software e hardware que possam substituir os avanços tecnológicos com os quais a comunidade de software se acostumou. A fim de proporcionar escalabilidade e melhor desempenho, é preciso investigar e integrar diferentes áreas.

Primeiro, deve-se revisar e discutir a arquitetura atual do sistema, de modo que os aceleradores para domínios específicos possam ser mais facilmente desenvolvidos e integrados ao sistema de computação. Geralmente, aplicações enormes e complexas possuem diferentes partes, cujo comportamento é muito diferente. Se, para cada parte de código é necessário desenvolver um acelerador, isso significa que cada acelerador deve ser facilmente integrado na plataforma de hardware, sem comprometer as outras partes de software. Além disso, um segundo passo importante é observar que, uma vez que a integração deve ser perfeita para o programador, um conjunto de ferramentas que possa ajudar essa adaptação da plataforma deve ser usado para suportar esta execução combinada, de modo que a tarefa de programação em si não seja aumentada. Por fim, destaca-se que o paradigma de computação na nuvem, ou, de forma mais generalizada, de *offloading* de carga computacional para uma infraestrutura compartilhada, é um modelo de sucesso que dificilmente se tornará obsoleto no futuro próximo. Assim, soluções para oferecer processamento energeticamente eficiente devem estar cientes da natureza distribuída dos recursos computacionais, dos custos e limitações de cada nodo computacional e da comunicação entre eles.

A crescente demanda por aplicações computacionalmente custosas em dispositivos de baixo custo e com severas restrições em capacidade de processamento e armazenamento popularizou, ao longo dos últimos anos, o uso de *cloud computing*. Essa abordagem permite que tarefas complexas sejam realizadas em grandes infraestruturas de alto desempenho, aliviando a carga computacional dos dispositivos finais dos usuários. Aplicações, por exemplo, de realidade aumentada e de aprendizado de máquina beneficiam-se de cloud computing para serem oferecidas em dispositivos de baixo custo, como aqueles que compõem a Internet das Coisas (*Internet of Things* - IoT). Quando tais aplicações demandam baixa latência, entretanto, cloud computing pode ser uma solução inviável, devido à grande distância entre os recursos computacionais e seus usuários.

A decentralização dos recursos computacionais tradicionalmente oferecidos em cloud computing os aproxima dos dispositivos dos usuários, habilitando seu uso em aplicações que demandam latência baixa e previsível. Além disso, outras vantagens são obtidas, como redução de tráfego no núcleo da rede e potenciais benefícios em segurança. Esse processo de decentralização dá origem a novos paradigmas que podem receber diferentes nomes, como *fog* ou *edge computing* [Satyanarayanan 2017, Long et al. 2018, Shi et al. 2016]. Mais recentemente, uma variação tipicamente chamada *in-network computing* também tem sido investigada [Kianpisheh and Taleb 2023].

Existem, entretanto, variações quanto às definições e à ênfase dada a cada tipo de abordagem por diferentes autores. Em [Satyanarayanan 2017] é defendida uma definição de edge restrita a servidores nas bordas da infraestrutura rede, também conhecidos como

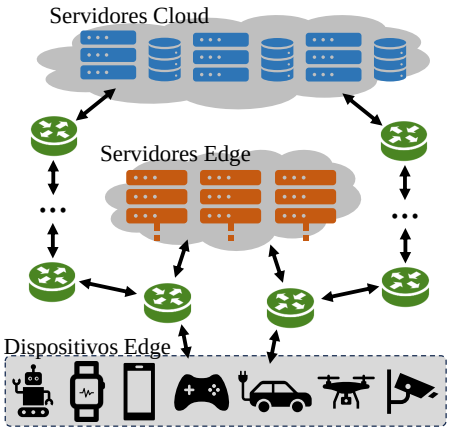


Figura 4.2. Cenário de dispositivos edge, elementos de rede, servidores edge e servidores cloud.

cloudlets [Satyanarayanan et al. 2009], sob o argumento que dispositivos com limitações mais severas de tamanho e potência não oferecem o desempenho necessário. Essa definição, entretanto, exclui diversas abordagens que habilitam aplicações relevantes, como a plataforma apresentada em [Long et al. 2018]. Em [Shi et al. 2016], qualquer dispositivo no caminho entre a fonte dos dados e os servidores centralizados da nuvem pode ser considerado edge. Isso inclui, portanto, tanto dispositivos edge de baixo custo e com severas restrições de energia e potência, como *smartphones* ou *wearables*, quanto cloudlets de maior capacidade dispostos nas bordas da rede, entre os quais as tarefas computacionais podem ser divididas. Esta definição, embora abrangente, permite pouca diferenciação de abordagens adequadas para diferentes nichos de aplicação.

Aqui, adotaremos uma definição similar à apresentada em [Kianpisheh and Taleb 2023], que permite uma distinção clara entre *in-network computing* e *edge computing*. A Figura 4.2 apresenta a relação destes paradigmas: os dispositivos edge produzem e consomem dados, que trafegam através de elementos de rede (em verde), como roteadores e *switches*. Tais elementos podem realizar processamento sobre esses dados, configurando *in-network computing*. Os dados podem, ainda, ser encaminhados até servidores edge, próximos dos dispositivos finais, ou até servidores cloud, de maior capacidade e potencialmente mais distantes. A escolha mais adequada dependerá dos requisitos da aplicação e da disponibilidade de recursos.

Além dessas diversas possibilidades de local, existem diferentes opções para o dispositivo responsável pela realização da tarefa em si, criando efetivamente um cenário de grande heterogeneidade arquitetural. Uma vez que cada dispositivo é mais adequado para aplicações com determinadas características, uma infraestrutura heterogênea potencialmente poderá oferecer o dispositivo mais adequado para cada tarefa, com benefícios em desempenho, custo e consumo de energia [Cooke and Fahmy 2020]. Dentre os dispositivos mais comumente utilizados, destacamos processadores de propósito geral (*General-Purpose Processors* - GPPs), unidades de processamento gráfico (*Graphics Processing Units* - GPUs) e dispositivos reconfiguráveis como *Field-Programmable Gate Arrays* (FPGAs).

4.2. Domínio de Aplicação

A necessidade de maior desempenho ainda está fortemente presente no mercado: veículos auto-guiados, indústria do entretenimento, compras on-line, automação de serviços usando inteligência artifical. Os algoritmos subjacentes que suportam essas aplicações importantes exigem muitos recursos de processamento e dados, e sua futura implantação ou escalabilidade é limitada pela quantidade de hardware que se pode usar para calcular as funções necessárias.

Essas aplicações requerem o processamento de uma enorme quantidade de dados, de forma estruturada (armazenados em banco de dados) mas também desestruturada (armazenadas na web). Os dados estruturados aqui são aqueles com uma alta localidade espacial, que pode ser observada na execução de aplicativos multimídia, por exemplo. Desestruturado significa que os dados estão espalhados na memória, no disco, ou mesmo na rede, sem localidade espacial, e onde uma solução simples usando arquiteturas *Single Instruction Multiple Data* (SIMD) como GPUs não pode ser facilmente implantada.

A arquitetura de processadores, até recentemente, sempre seguiu as necessidades do mercado. O co-processador numérico de ponto flutuante, as instruções MMX, SSE, AVX, todos tornaram-se realidade porque havia uma clara necessidade de mercado. À medida que a lei de Moore continuava, os dispositivos extras que poderiam ser integrados foram usados para suportar melhor os requisitos de software. O primeiro desvio desse padrão ocorreu apenas recentemente, com máquinas *multicore*. À medida que a lei de Moore perdia momentum, e a frequência não podia ser aumentada devido às limitações de fabricação e potência, a indústria de hardware de computação forneceu ao mercado mais núcleos, e não um desempenho maior do processador individualmente. Olhando para o presente e o futuro próximo, o tamanho de dados de aplicativos futuros só pode aumentar. À medida que a automação substitui a fabricação e os serviços, a quantidade de dados que podem ser coletados e analisados pode alcançar facilmente a escala dos terabytes [Jun et al. 2017, Blat et al. 2016, Algur and Sakri 2015].

A análise e implantação de tal dilúvio de dados são atualmente observadas em sistemas de recomendação, propagação de crenças, detecção de fraude, reconhecimento de imagem, processamento de sinais, compressão multimídia e outros. Essas aplicações, por sua vez, popularizaram os algoritmos subjacentes, como redes neurais, aprendizagem baseada em árvores de decisão, decomposição de valor único, decomposição de tensor e autovalor, gradiente descendente estocástico e muitos outros. O que o hardware deve fornecer para essas aplicações, que abrangem domínios tão amplos? Nos exemplos acima, os dados não são apenas grandes, mas também têm muitas lacunas. Ainda, observa-se que para muitas dessas aplicações, há uma tolerância intrínseca a imprecisões nos resultados, ou seja, um resultado aproximado ainda pode ser considerado aceitável. Por exemplo, em aplicações de áudio e vídeo, lidamos com as limitações sensoriais humanas. Já em aplicações de reconhecimento de imagens, divergências numéricas podem ocorrer e ainda assim termos uma classificação correta. A partir de todas essas características, identificamos que qualquer acelerador de hardware para esses domínios deve:

- possibilitar com que os dados sejam processados onde eles são gerados, sem movimentos desnecessários entre processador e memória que drenam energia [Santos

et al. 2021a, Santos et al. 2021b];

- fornecer uma maneira de lidar com dados que, por vezes, apresentam alta localidade (aplicativos de vídeo, por exemplo), onde um SIMD de baixa potência pode suportar o desenvolvimento, mas também acelerar a aplicação onde os dados apresentam uma localização muito baixa (uma característica fundamental da maioria dos algoritmos acessando árvores ou grafos) [de Lima et al. 2022];
- explorar eficientemente possibilidades de redução de precisão para atender diferentes requisitos não funcionais, como uso de recursos [Leipnitz and Nazar 2019b], processamento em tempo real [Leipnitz and Nazar 2019a] ou vazão total de processamento [Leipnitz and Nazar 2020];
- aproveitar oportunidades de computação com tecnologias alternativas, que embora garantam baixa energia, trazem outro conjunto de problemas a serem resolvidos [de Lima et al. 2022, de Lima and Carro 2022];
- prover tolerância a *soft errors*, cada vez mais presentes em tecnologias próximas ao limite da lei de Moore [Nazar et al. 2021a, dos Santos et al. 2022, Nazar et al. 2021b];
- permitir a alteração automática de algoritmos, para aproveitar as características dos dados do problema de modo a realizar oportunidades de computação com baixa energia [Gonçalves et al. 2019].

O mercado atual encontra-se muito concentrado em aplicativos no domínio da análise de dados, aprendizado de máquinas, automação de serviços, onde a quantidade de dados é enorme e a localidade variável dos mesmos dados exclui o uso de uma arquitetura única. Este conjunto de aplicações possui as características comuns que podem ser exploradas, pois:

- são complexas o suficiente para serem significativas como estudo de caso, mas ao mesmo tempo factíveis para uma equipe universitária;
- possuem características próprias de sistemas complexos que precisam ser escaláveis, pelo uso de hardware e software, e por envolver algoritmos de diferentes domínios e uma massa de dados sempre crescente;
- devem ser realizadas com alto grau de automação, para serem controladas por uma equipe pequena de projeto;
- podem ser prototipadas em placas facilmente disponíveis, como GPUs e/ou com FPGAs de alto desempenho, a um custo relativamente baixo.

4.3. Detalhamento do problema

Nesta seção, detalharemos problemas atuais que trataremos nesse curso. Na seção 4.3.1, apresentaremos limitações nas arquiteturas atuais que impõem custos de desenvolvimento e limitam a eficiência dos sistemas computacionais. Na seção 4.3.2, discutiremos dificuldades na adoção de arquiteturas promissoras, como FPGAs, em infraestruturas compartilhadas.

4.3.1. Limitações das arquiteturas atuais

Quando se chega ao fim da lei de Moore, extrair mais desempenho dos atuais processadores torna-se cada vez mais difícil. A exploração do paralelismo utilizando GPUs e múltiplos núcleos fora de ordem, que são as soluções atuais propostas pela indústria, são claramente viáveis se e somente se a tecnologia continuar a permitir mais núcleos e/ou maior frequência, e, para ambos os casos, as perspectivas tecnológicas são desafiadoras. A tecnologia pode ajudar, mas em diferentes setores: a integração com biologia, sensores, processamento quântico, processamento com tubos de nanocarbono e outras técnicas estão no horizonte. No entanto, em cada exemplo, perde-se o que tem sido o estilo fundamental de desenvolvimento de hardware e software, que é a integração total de dispositivos de computação em um único dispositivo.

Outra solução atual é o uso extensivo do paralelismo. No entanto, existem vários impedimentos para esta abordagem. O primeiro e mais óbvio é o desafio de desenvolvimento de programas paralelos. A maioria dos programadores não sabe como programar em paralelo, e muitas aplicações são simplesmente muito complexas para serem convertidas em uma versão paralela. Outro aspecto importante é que, ao usar máquinas paralelas, o ganho é limitado pelo número de processadores ou unidades paralelas, e ainda é mais limitado pela lei de Amdahl (a parte serial dominará o tempo de execução). Além disso, em termos energéticos, o paralelismo é inócuo: como $Energia = Potência \times Tempo$, se alguém pudesse usar N processadores, o melhor fator de redução de tempo seria N, mas a energia gasta é a mesma, já que os N processadores vão dissipar N vezes mais potência. Dadas as limitações reais impostas pela lei de Amdahl, é provável que a energia aumente. Em um cenário onde as demandas de aplicativos aumentam e o desenvolvimento de hardware encontra-se estagnado, há apenas uma maneira de aumentar o desempenho com menor energia: mudar o tempo de execução, reduzindo os estrangulamentos realmente presentes nas transformações de algoritmos a serem executados em um determinado dispositivo de hardware.

Neste curso esses estrangulamentos serão identificados, e propõem-se soluções que podem ser automatizadas, permitindo que a pilha de software seja tão abstrata quanto os programadores assim o desejarem. Desta forma, pode-se sustentar a lei de Moore para o futuro próximo, enquanto outros avanços tecnológicos podem vir a ajudar em domínios diferentes.

Dedicar mais tempo ao desenvolvimento de algoritmos não pode ser a solução: os programadores de primeira linha não são apenas caros, são difíceis de encontrar. Além disto, atualmente os aplicativos podem chegar facilmente à região de milhões de linhas de código (ver Tabela 1). Este aumento nas linhas de código é explicado por vários fatores, mas dois são os mais óbvios: a complexidade natural da aplicação final e o uso de linguagens de programação mais abstratas por programadores para aumento de produtividade. Como apontado em [Cass 2022], as cinco linguagens mais demandadas pelo mercado são SQL, Java, Python, JavaScript e C#. A complexidade do tamanho, no domínio do software, sempre foi resolvida por uma maior abstração, o que explica a popularidade das linguagens acima mencionadas.

Infelizmente, é claro também que uma maior abstração não ajuda o desempenho, pelo contrário: a memória é quase completamente abstraída, quase nenhuma construção

de paralelismo está disponível, já que o objetivo de uma maior abstração é exatamente ocultar detalhes de hardware, que são necessários quando se pensa em desempenho e melhores algoritmos. Quando se acrescenta estagnação de hardware com uma maior complexidade de aplicação exigindo linguagens de abstração mais altas, o cenário para a execução do software em um futuro próximo é menor desempenho ou um tempo de desenvolvimento mais longo. Ambos os casos estão contra a tradição dos últimos 40 anos na computação, pelo menos, e têm um efeito severo no ambiente econômico moderno, cada vez mais dependente do desenvolvimento de software e hardware.

Os circuitos de processadores atuais utilizam a maior parte de sua área na memória incorporada, suportando vários níveis de armazenamento em cache [Nalamalpu et al. 2015, Kurd et al. 2010, Lotfi-Kamran et al. 2012, Santos et al. 2016]. É evidente que uma questão interessante é como se pode usar melhor esta área, para favorecer a computação de baixa energia nas aplicações com uso intensivo de dados.

O centro da estratégia é a identificação de que, para algoritmos com diferentes características, diferentes estruturas de hardware devem estar disponíveis. Este é o princípio do que é ter um conjunto de aceleradores dedicados, prontamente disponíveis e suportados por uma linguagem de programação de alto nível. A generalização deste princípio "acelerador quando necessário" não é claramente uma tarefa fácil. Os tipos de dados mais abstratos não são tão facilmente mapeados para uma estrutura de hardware. No passado recente, esta estratégia tem sido utilizada para grandes mercados. Por exemplo, as GPUs usam vários processadores e se destacam no suporte a aplicações SIMD [Abraham et al. 2015, Liu et al. 2013]. Os programadores podem alcançar alto desempenho usando uma linguagem de suporte de alto nível (neste caso, CUDA [NVIDIA 2023]). Outro bom exemplo pode ser encontrado analisando-se FPGAs nos domínios de telecomunicações e finanças, onde suas operações de bit permitem que operadores customizados e canais massivamente paralelos sejam processados de uma maneira eficiente em termos de energia [Lindtjorn et al. 2011, Park et al. 2015, Giefers et al. 2014].

Dado o tamanho e a complexidade das aplicações atuais e futuras, é improvável que um único modelo de computação, como SIMD para GPUs ou operações maciças MIMD para FPGAs possa abranger todos os aspectos de um programa complexo. Existem alguns trabalhos que tentam mesclar, por exemplo, CPUs e GPUs de forma perfeita, como em [Jablin et al. 2011, Kim et al. 2012]. No entanto, a dissipação de potência desta estratégia é significativa, comprometendo a escalabilidade [Deshpande and Draper 2016, Gamell et al. 2013]. Outra estratégia foi utilizada no processamento em memória (*Processing In Memory* - PIM), delegando algumas operações com pouca localidade à memória, como em [Siegl et al. 2016, Scrbak et al. 2017, Santos et al. 2021a, Santos et al. 2021b, de Lima et al. 2022]. Como não há localidade, o sistema de caches torna-se altamente ineficiente e um enorme dreno de energia.

Os exemplos acima indicam a direção correta, pois eles suportam aceleradores especializados em mercados de massa. O conjunto de aplicações atual diz respeito ao processamento massivo de dados com ou sem localidade espacial, com um mínimo gasto de energia. O principal conceito a ser explorado é que o processamento deve ser desenvolvido onde é menos dispendioso fazê-lo. Além disso, sempre que possível, cruzam-se as camadas de abstração, para que se possa superar o processador commoditizado de última

geração construído na mesma tecnologia.

4.3.2. Arquiteturas heterogêneas em infraestruturas compartilhadas

O uso de dispositivos heterogêneos traz grandes vantagens para infraestruturas compartilhadas, como as encontradas em edge e in-network computing. Como mencionado, aplicações distintas, com requisitos diversos, podem ser atendidas com melhor eficiência por um conjunto de dispositivos heterogêneos. Nesse contexto, FPGAs são uma solução promissora, já que aliam alta capacidade de processamento e eficiência com uma grande flexibilidade e a possibilidade de reprogramação após implantação, particularmente relevante em infraestruturas onde a totalidade das aplicações de interesse não é conhecida a priori. Entretanto, desafios importantes ainda devem ser endereçados para permitir o usufruto pleno dessas vantagens.

FPGAs são dispositivos reconfiguráveis que integram blocos lógicos configuráveis (*Configurable Logic Blocks* - CLBs), capazes de realizar qualquer função booleana com uma quantidade limitada de entradas, e uma malha de circuitos de interconexão, também reconfigurável [Boutros and Betz 2021]. Assim, através de um arquivo de reconfiguração, também chamado de *bitstream*, o dispositivo pode ser reconfigurado para implementar circuitos lógicos complexos. FPGAs típicos incluem, ainda, circuitos dedicados para funções aritméticas, memórias e blocos de entrada e saída para conexão com dispositivos externos.

A geração do *bitstream* é, em geral, feita por ferramentas de software fornecidas pelo próprio fabricante, a partir de descrições no nível de transferência de registradores (*Register Transfer Level* - RTL), codificadas em uma linguagem de descrição de hardware (*Hardware Description Language* - HDL) como Verilog ou VHDL. Como alternativa a esse processo, geralmente bastante oneroso, pode ser utilizada a síntese de alto nível (*High-Level Synthesis* - HLS) [Cong et al. 2022]. HLS consiste em gerar modelos RTL de um circuito, tipicamente a partir de código em alguma linguagem de programação de mais alto nível, como C/C++. Ferramentas de HLS vêm em constante evolução e aumento de popularidade, devido às melhorias na qualidade dos circuitos produzidos e aos importantes ganhos em produtividade [Liang et al. 2012].

Um aspecto relevante de HLS é a possibilidade de utilizar diretivas para guiar o processo de síntese quanto ao uso de técnicas de otimização como *loop unrolling*, *pipelining* e *array partitioning* [AMD 2023a]. Tais técnicas têm impacto direto nas métricas de desempenho e uso de recursos e de energia e, através de seu uso, desenvolvedores podem encontrar soluções mais adequadas para as restrições da aplicação em questão. Essa capacidade é particularmente relevante no contexto de edge e in-network computing, já que, conforme mencionado, frequentemente teremos restrições de desempenho ou de uso de recursos variadas para cada aplicação e a possibilidade de termos múltiplas versões de uma mesma funcionalidade pode ser vantajosa. Observa-se, porém, que a inserção manual de tais diretivas volta a onerar o processo de desenvolvimento, na contramão dos benefícios esperados de HLS.

Particularmente para o caso de FPGAs, o seu uso em infraestruturas compartilhadas impõe algumas dificuldades. A troca da configuração de um FPGA, realizada através da transferência de um novo bitstream, é um processo substancialmente mais longo que

uma troca de contexto em um GPP, o que dificulta o uso de compartilhamento temporal para aplicações que continuamente processam *streams* de dados, bastante comuns em edge e in-network computing [Gobatto et al. 2022]. Embora *overlays* possam facilitar a gerência do dispositivo em tais aplicações [Bachini Lopes et al. 2023], para contornar essa limitação, o compartilhamento espacial do dispositivo, ou seja, utilizar partes dos recursos reconfiguráveis concomitantemente alocadas a diferentes aplicações, torna-se uma funcionalidade importante [Vaishnav et al. 2018]. Para permitir a alocação independente de aplicações em regiões do FPGA, a reconfiguração parcial dinâmica (*Dynamic Partial Reconfiguration* - DPR) é uma funcionalidade suportada pelos FPGAs modernos dos principais fabricantes [AMD 2023b, Intel 2023].

DPR consiste em utilizar bitstreams parciais para reconfigurar parte dos recursos do FPGA, enquanto os demais componentes permanecem em operação ininterrupta. Essa funcionalidade permite compor múltiplas combinações de funções a partir de um conjunto enxuto de bitstreams parciais. Além disso, cada funcionalidade pode ser substituída independentemente, permitindo a instanciação de novas aplicações sem interromper aquelas já em operação. Para sua utilização, entretanto, DPR demanda um particionamento prévio dos recursos do dispositivo, criando partições dinamicamente reconfiguráveis fixas. O adequado dimensionamento dessas partições dá origem a uma variação do problema de *floorplanning*, tradicionalmente encontrado em projeto de dispositivos eletrônicos e já abordado para FPGAs em trabalhos como [Seyoum et al. 2019, Galea et al. 2018, Tang et al. 2020], porém para conjuntos fixos e previstos de módulos que compartilham o dispositivo.

Após o particionamento dos recursos no FPGA, existe ainda o desafio de definir qual tarefa será realizada em cada partição, ou seja, o posicionamento e escalonamento de tarefas dentro do dispositivo, como realizado em [Tang et al. 2020, Yao et al. 2022, Dhar et al. 2022], porém para um único dispositivo. Em uma escala mais abrangente, esse problema ainda deve considerar que temos diversos nodos na infraestrutura, potencialmente com dispositivos de diferentes capacidades, dispostos com variadas distâncias do usuário final, e com diferentes cargas de ocupação. Ou seja, o uso de FPGAs com DPR adiciona mais um aspecto a ser considerado pelo problema de posicionamento de tarefas em infraestruturas de edge e in-network computing. A modelagem dos recursos dos dispositivos é frequentemente simplificada em trabalhos da área, que consideram o FPGA como uma coleção de recursos sem uma estrutura fixa [Sharma et al. 2020] ou não consideram os diferentes recursos demandados por cada tarefa [Cooke and Fahmy 2020]. Essas simplificações podem levar a soluções teóricas ineficientes ou que não são implementáveis na prática devido às restrições geométricas do dispositivo que devem ser respeitadas [AMD 2023b]. Em [Sun et al. 2019] é apresentado um algoritmo para o posicionamento de tarefas em múltiplos FPGAs que modela os recursos heterogêneos e as restrições geométricas de particionamento, entretanto desconsiderando o possível caráter distribuído desses dispositivos, que pode introduzir restrições de latência e banda para as soluções.

Ainda, a complexidade de desenvolvimento de implementações que façam uso eficiente de FPGAs é um fator que pode desestimular seu uso. Embora HLS ofereça uma alternativa de maior produtividade, a aplicação de diretivas de síntese ainda é uma tarefa dispendiosa e essencialmente manual. Além da expertise necessária para identificar boas oportunidades de uso de cada técnica de otimização, longos ciclos de síntese são

necessários para avaliar cada solução candidata. O uso de heurísticas de exploração de espaço de projeto (*Design Space Exploration* - DSE) é uma alternativa investigada para automatizar esse processo e reduzir tempo de projeto [Schafer and Wang 2020]. Uma vez que a quantidade de recursos utilizados é afetada pelas decisões de tais ferramentas, elas também têm impacto nas dimensões mais adequadas das partições definidas durante o floorplanning, uma possibilidade ainda não investigada no contexto de infraestruturas de computação de borda e em rede.

Sumarizando, embora o uso de arquiteturas heterogêneas, em particular aquelas dotadas de FPGAs, seja promissor, diversos desafios ainda são encontrados. Do ponto de vista de desenvolvimento, ferramentas de HLS integradas a heurísticas de DSE são capazes de produzir múltiplas implementações de cada funcionalidade. Essas, entretanto, ainda não estão adequadamente integradas a ferramentas de gerência cientes das particularidades desses dispositivos. Em particular, o dimensionamento de partições com o objetivo de acomodar múltiplos designs produzidos automaticamente através de DSE em HLS ainda carece de solução. Ainda, o posicionamento de tarefas em infraestruturas heterogêneas deve ser realizado de forma ciente das partições dos FPGAs, garantindo que a disponibilidade de recursos de cada partição seja respeitada.

4.4. Possíveis soluções

Os desafios identificados para computação energeticamente eficiente estão espalhados por todas as camadas de abstração vistas na Figura 4.1. Assim, soluções adequadas para esse problema deverão estar igualmente em todas as camadas e, frequentemente, explorando múltiplas camadas de forma simultânea e colaborativa.

Quando olhamos para cada dispositivo isoladamente, temos que o enfraquecimento da lei de Moore e a dificuldade de explorarmos paralelismo de forma eficiente, são desafios prementes. Nesse nível, tanto abordagens arquiteturais, como processamento em memória (PIM) [Santos et al. 2021b] e arquiteturas heterogêneas, quanto abordagens em nível de circuito, como voltagens próximas à tensão de *threshold* dos dispositivos (*Near-Thresold Voltage* - NTV) [Tonetto et al. 2022], são promissoras.

Em um nível mais abrangente, considerando a integração de múltiplos dispositivos em sistemas compartilhados de computação de nuvem e borda, também identificamos desafios relevantes: *i*) como oferecer recursos computacionais flexíveis, porém de alto desempenho e eficiência energética; *ii*) como desenvolver aplicações para tais recursos, capazes de atender diferentes requisitos não funcionais, com custos de projeto reduzidos; e *iii*) como instanciar aplicações nessa infraestrutura distribuída e heterogênea de recursos, ocupando-os de forma eficiente e igualmente ciente dos requisitos de cada usuário.

A Figura 4.3 apresenta um conjunto de soluções promissor para tais desafios. Primeiramente, o uso de HLS permite a implementação de aceleradores dedicados por desenvolvedores de software. Ao utilizarmos dispositivos como FPGAs, mantemos uma infraestrutura flexível, capaz de atender diferentes aplicações à medida em que as demandas são recebidas ao longo do tempo. Ainda, através de heurísticas de exploração de espaço de projeto (DSE), é possível produzir diferentes designs, ocupando diferentes pontos da fronteira de Pareto que envolve uso de recursos e desempenho. Assim, é possível utilizar a implementação de menor custo que ainda atende cada requisição de usuário. Podemos,

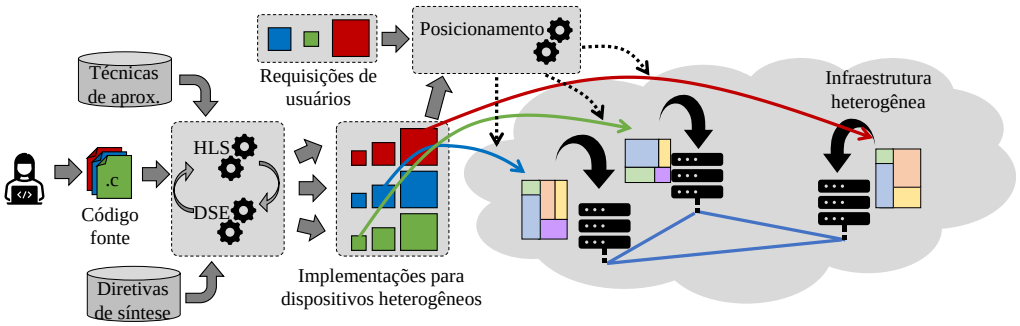


Figura 4.3. Fluxo de exploração de espaço de projeto e posicionamento de tarefas em uma infraestrutura heterogênea.

nesse ponto, considerar também a precisão demandada para cada aplicação: reduções substanciais de custo podem ser obtidas através de técnicas de computação aproximativa, que introduzem aproximações controladas na aplicação para reduzir uso de recursos ou de potência ou para aumentar o desempenho. A exploração do espaço conjunto que integra diretivas de síntese e técnicas de aproximação, embora complexa, pode encontrar soluções superiores em termos do *trade-off* entre custos e desempenho.

Após a elaboração desse conjunto diverso de implementações heterogêneas, resta ainda o problema de dispô-las adequadamente na infraestrutura. Uma vez que os enlaces de comunicação têm suas próprias latências e bandas limitadas, a garantia do atendimento dos requisitos do usuário não pode se dar somente a partir das características isoladas de cada acelerador. Deve-se, portanto, considerar o cenário completo do acelerador inserido em um ambiente de computação de borda e em rede. Através de mecanismos automatizados para esse posicionamento, cientes desses cenário completo, podemos facilitar a gerência dessas infraestruturas e identificar as configurações mais adequadas de acelerador para cada caso.

4.5. Conclusão

Neste capítulo, apresentamos os principais desafios vislumbrados para o desenvolvimento de sistemas computacionais energeticamente eficientes. Observamos que esses desafios englobam as mais diversas camadas desses sistemas: desde o desenvolvimento de software para novas aplicações, uma tarefa que demanda cada vez mais abstração para garantir produtividade, passando pela gerência de infraestruturas heterogêneas, e chegando até novos paradigmas arquiteturais. Este cenário desafiador portanto, dificilmente será resolvido por soluções limitadas a uma única camada de abstração dos sistemas, demandando soluções que cooperativamente reduzem consumo energético em todas as camadas.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

[Abraham et al. 2015] Abraham, M. J., Murtola, T., Schulz, R., Páll, S., Smith, J. C., Hess, B., and Lindahl, E. (2015). Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1-2:19–25.

[Algur and Sakri 2015] Algur, S. P. and Sakri, L. I. (2015). Parallelized genomic sequencing model: A big data approach for bioinformatics application. In *2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pages 69–74.

[AMD 2023a] AMD (2023a). Vitis High-Level Synthesis User Guide.

[AMD 2023b] AMD (2023b). Vivado Design Suite User Guide - Dynamic Function eXchange.

[Bachini Lopes et al. 2023] Bachini Lopes, F., Schaeffer-Filho, A. E., and Nazar, G. L. (2023). Modular vnf components acceleration with fpga overlays. *IEEE Transactions on Network and Service Management*, 20(1):846–857.

[Blat et al. 2016] Blat, J., Evans, A., Kim, H., Imre, E., Polok, L., Ila, V., Nikolaidis, N., Zemčík, P., Tefas, A., Smrž, P., Hilton, A., and Pitas, I. (2016). Big data analysis for media production. *Proceedings of the IEEE*, 104(11):2085–2113.

[Boutros and Betz 2021] Boutros, A. and Betz, V. (2021). Fpga architecture: Principles and progression. *IEEE Circuits and Systems Magazine*, 21(2):4–29.

[Cass 2022] Cass, S. (2022). Top Programming Languages 2022.

[Cong et al. 2022] Cong, J., Lau, J., Liu, G., Neuendorffer, S., Pan, P., Vissers, K., and Zhang, Z. (2022). Fpga hls today: Successes, challenges, and opportunities. *ACM Trans. Reconfigurable Technol. Syst.*, 15(4).

[Cooke and Fahmy 2020] Cooke, R. A. and Fahmy, S. A. (2020). A model for distributed in-network and near-edge computing with heterogeneous hardware. *Future Generation Computer Systems*, 105:395–409.

[Coussy et al. 2009] Coussy, P., Gajski, D. D., Meredith, M., and Takach, A. (2009). An introduction to high-level synthesis. *IEEE Design & Test of Computers*, 26(4):8–17.

[de Lima et al. 2022] de Lima, J. a. P. C., Brandalero, M., Hübner, M., and Carro, L. (2022). Stap: An architecture and design tool for automata processing on memristor tcams. *J. Emerg. Technol. Comput. Syst.*, 18(2).

[de Lima and Carro 2022] de Lima, J. P. C. and Carro, L. (2022). Quantization-aware in-situ training for reliable and accurate edge ai. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1497–1502.

[Deshpande and Draper 2016] Deshpande, A. M. and Draper, J. T. (2016). A new metric to measure cache utilization for hpc workloads. In *Proceedings of the Second International Symposium on Memory Systems*, MEMSYS '16, page 10–17, New York, NY, USA. Association for Computing Machinery.

[Desjardins 2017] Desjardins, J. (2017). How Many Millions of Lines of Code Does It Take?

[Dhar et al. 2022] Dhar, A., Richter, E., Yu, M., Zuo, W., Wang, X., Kim, N. S., and Chen, D. (2022). Dml: Dynamic partial reconfiguration with scalable task scheduling for multi-applications on fpgas. *IEEE Transactions on Computers*, 71(10):2577–2591.

[dos Santos et al. 2022] dos Santos, F. F., Malde, S., Cazzaniga, C., Frost, C., Carro, L., and Rech, P. (2022). Experimental findings on the sources of detected unrecoverable errors in gpus. *IEEE Transactions on Nuclear Science*, 69(3):436–443.

[Galea et al. 2018] Galea, F., Carpov, S., and Zaourar, L. (2018). Multi-start simulated annealing for partially-reconfigurable fpga floorplanning. In *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 1335–1338.

[Gamell et al. 2013] Gamell, M., Rodero, I., Parashar, M., and Poole, S. (2013). Exploring energy and performance behaviors of data-intensive scientific workflows on systems with deep memory hierarchies. In *20th Annual International Conference on High Performance Computing*, pages 226–235.

[Giefers et al. 2014] Giefers, H., Plessl, C., and Förstner, J. (2014). Accelerating finite difference time domain simulations with reconfigurable dataflow computers. *SIGARCH Comput. Archit. News*, 41(5):65–70.

[Gobatto et al. 2022] Gobatto, L., Saquetti, M., Diniz, C., Zatt, B., Cordeiro, W., and Azambuja, J. R. (2022). Improving content-aware video streaming in congested networks with in-network computing. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1813–1817.

[Gonçalves et al. 2019] Gonçalves, L. R., Moura, R. F. D., and Carro, L. (2019). Aggressive energy reduction for video inference with software-only strategies. *ACM Trans. Embed. Comput. Syst.*, 18(5s).

[Intel 2023] Intel (2023). Intel Quartus Prime Software Features Partial Reconfiguration.

[Jablin et al. 2011] Jablin, T. B., Prabhu, P., Jablin, J. A., Johnson, N. P., Beard, S. R., and August, D. I. (2011). Automatic cpu-gpu communication management and optimization. In *Proceedings of the 32nd ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '11, page 142–151, New York, NY, USA. Association for Computing Machinery.

[Jones 2018] Jones, N. (2018). How to stop data centres from gobbling up the world's electricity. *Nature*, 561(7722):163–167.

[Jun et al. 2017] Jun, S.-W., Xu, S., and Arvind (2017). Terabyte sort on fpga-accelerated flash storage. In *2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pages 17–24.

[Kianpisheh and Taleb 2023] Kianpisheh, S. and Taleb, T. (2023). A survey on in-network computing: Programmable data plane and technology specific applications. *IEEE Communications Surveys & Tutorials*, 25(1):701–761.

[Kim et al. 2012] Kim, J., Seo, S., Lee, J., Nah, J., Jo, G., and Lee, J. (2012). Snucl: An opencl framework for heterogeneous cpu/gpu clusters. In *Proceedings of the 26th ACM International Conference on Supercomputing*, ICS '12, page 341–352, New York, NY, USA. Association for Computing Machinery.

[Kurd et al. 2010] Kurd, N. A., Bhamidipati, S., Mozak, C., Miller, J. L., Wilson, T. M., Nemani, M., and Chowdhury, M. (2010). Westmere: A family of 32nm ia processors. In *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, pages 96–97.

[Leipnitz and Nazar 2019a] Leipnitz, M. T. and Nazar, G. L. (2019a). High-level synthesis of approximate designs under real-time constraints. *ACM Trans. Embed. Comput. Syst.*, 18(5s).

[Leipnitz and Nazar 2019b] Leipnitz, M. T. and Nazar, G. L. (2019b). High-level synthesis of resource-oriented approximate designs for fpgas. In *2019 56th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6.

[Leipnitz and Nazar 2020] Leipnitz, M. T. and Nazar, G. L. (2020). Throughput-oriented spatio-temporal optimization in approximate high-level synthesis. In *2020 IEEE 38th International Conference on Computer Design (ICCD)*, pages 316–323.

[Liang et al. 2012] Liang, Y., Rupnow, K., Li, Y., Min, D., Do, M. N., and Chen, D. (2012). High-level synthesis: Productivity, performance, and software constraints. *JECE*, 2012.

[Lindtjorn et al. 2011] Lindtjorn, O., Clapp, R., Pell, O., Fu, H., Flynn, M., and Mencer, O. (2011). Beyond traditional microprocessors for geoscience high-performance computing applications. *IEEE Micro*, 31(2):41–49.

[Liu et al. 2013] Liu, Y., Wirawan, A., and Schmidt, B. (2013). Cudasw++ 3.0: accelerating smith-waterman protein database search by coupling cpu and gpu simd instructions. *BMC bioinformatics*, 14:1–10.

[Long et al. 2018] Long, C., Cao, Y., Jiang, T., and Zhang, Q. (2018). Edge computing framework for cooperative video processing in multimedia iot systems. *IEEE Transactions on Multimedia*, 20(5):1126–1139.

[Lotfi-Kamran et al. 2012] Lotfi-Kamran, P., Grot, B., Ferdman, M., Volos, S., Kocberber, O., Picorel, J., Adileh, A., Jevdjic, D., Idgunji, S., Ozer, E., and Falsafi, B. (2012). Scale-out processors. In *Proceedings of the 39th Annual International Symposium on Computer Architecture*, ISCA '12, page 500–511, USA. IEEE Computer Society.

[Nalamalpu et al. 2015] Nalamalpu, A., Kurd, N., Deval, A., Mozak, C., Douglas, J., Khanna, A., Paillet, F., Schrom, G., and Phelps, B. (2015). Broadwell: A family of ia 14nm processors. In *2015 Symposium on VLSI Circuits (VLSI Circuits)*, pages C314–C315.

[Nane et al. 2016] Nane, R., Sima, V.-M., Pilato, C., Choi, J., Fort, B., Canis, A., Chen, Y. T., Hsiao, H., Brown, S., Ferrandi, F., Anderson, J., and Bertels, K. (2016). A survey and evaluation of fpga high-level synthesis tools. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(10):1591–1604.

[Nazar et al. 2021a] Nazar, G. L., Kopper, P. H., Leipnitz, M. T., and Juurlink, B. (2021a). Lightweight dual modular redundancy through approximate computing. In *2021 XI Brazilian Symposium on Computing Systems Engineering (SBESC)*, pages 1–8.

[Nazar et al. 2021b] Nazar, G. L., Kopper, P. H., Leipnitz, M. T., and Juurlink, B. (2021b). Precep: Automatic insertion of partial redundancy based on critical error probability. *Microelectronics Reliability*, 126:114226. Proceedings of ESREF 2021, 32nd European Symposium on Reliability of Electron Devices, Failure Physics and Analysis.

[NVIDIA 2023] NVIDIA (2023). CUDA C++ Programming Guide.

[Park et al. 2015] Park, S.-W., Park, J., Bong, K., Shin, D., Lee, J., Choi, S., and Yoo, H.-J. (2015). An energy-efficient and scalable deep learning/inference processor with tetra-parallel mimd architecture for big data applications. *IEEE Transactions on Biomedical Circuits and Systems*, 9(6):838–848.

[Santos et al. 2016] Santos, P. C., Alves, M. A. Z., Diener, M., Carro, L., and Navaux, P. O. A. (2016). Exploring cache size and core count tradeoffs in systems with reduced memory access latency. In *2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, pages 388–392.

[Santos et al. 2021a] Santos, P. C., de Lima, J. P. C., de Moura, R. F., Alves, M. A. Z., Beck, A. C. S., and Carro, L. (2021a). Enabling near-data accelerators adoption by through investigation of datapath solutions. *International Journal of Parallel Programming*, 49:237–252.

[Santos et al. 2021b] Santos, P. C., Forlin, B. E., and Carro, L. (2021b). Providing plug n' play for processing-in-memory accelerators. In *2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 651–656.

[Satyanarayanan 2017] Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1):30–39.

[Satyanarayanan et al. 2009] Satyanarayanan, M., Bahl, P., Caceres, R., and Davies, N. (2009). The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Computing*, 8(4):14–23.

[Schafer and Wang 2020] Schafer, B. C. and Wang, Z. (2020). High-level synthesis design space exploration: Past, present, and future. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(10):2628–2639.

[Scrbak et al. 2017] Scrbak, M., Islam, M., Kavi, K. M., Ignatowski, M., and Jayasena, N. (2017). Exploring the processing-in-memory design space. *Journal of Systems Architecture*, 75:59–67.

[Seyoum et al. 2019] Seyoum, B. B., Biondi, A., and Buttazzo, G. C. (2019). Flora: Floorplan optimizer for reconfigurable areas in fpgas. *ACM Trans. Embed. Comput. Syst.*, 18(5s).

[Sharma et al. 2020] Sharma, G. P., Tavernier, W., Colle, D., and Pickavet, M. (2020). Vnf-aapc: Accelerator-aware vnf placement and chaining. *Computer Networks*, 177:107329.

[Shi et al. 2016] Shi, W., Cao, J., Zhang, Q., Li, Y., and Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646.

[Siegl et al. 2016] Siegl, P., Buchty, R., and Berekovic, M. (2016). Data-centric computing frontiers: A survey on processing-in-memory. In *Proceedings of the Second International Symposium on Memory Systems*, MEMSYS '16, page 295–308, New York, NY, USA. Association for Computing Machinery.

[Sun et al. 2019] Sun, Z., Zhang, H., and Zhang, Z. (2019). Resource-aware task scheduling and placement in multi-fpga system. *IEEE Access*, 7:163851–163863.

[Tang et al. 2020] Tang, Q., Wang, Z., Guo, B., Zhu, L.-H., and Wei, J.-B. (2020). Partitioning and scheduling with module merging on dynamic partial reconfigurable fpgas. *ACM Trans. Reconfigurable Technol. Syst.*, 13(3).

[Tonetto et al. 2022] Tonetto, R. B., Beck, A. C. S., and Nazar, G. L. (2022). Snap: Selective ntv heterogeneous architectures for power-efficient edge computing. In *2022 25th Euromicro Conference on Digital System Design (DSD)*, pages 357–364.

[Vaishnav et al. 2018] Vaishnav, A., Pham, K. D., and Koch, D. (2018). A survey on fpga virtualization. In *2018 28th International Conference on Field Programmable Logic and Applications (FPL)*, pages 131–1317.

[Yao et al. 2022] Yao, R., Zhao, Y., Yu, Y., Zhao, Y., and Zhong, X. (2022). Fast search and efficient placement algorithm for reconfigurable tasks on modern heterogeneous fpgas. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 30(4):474–487.

Capítulo 5

Realidade Virtual: Potencialidades de uma Nova Plataforma Interativa

Luciana Nedel e Carla M.D.S. Freitas

Programa de Pós-Graduação em Computação (PPGC) - UFRGS

Abstract

High-quality virtual reality head-mounted displays (HMDs) are currently available at the same price as a smartphone. They have emerged at the same time as the concept of the metaverse has become popular. However, the acceptance of this new technology in the everyday life of the average citizen is still uncertain and a gamble. In this chapter, we present the key concepts and challenges in the field and their use in two application areas: simulation and information visualization.

Resumo

*Óculos de realidade virtual (*head mounted display *– HMD) de alta qualidade estão atualmente disponíveis pelo mesmo valor que um* smartphone*. Eles surgem ao mesmo tempo em que o conceito de* metaverso *se torna popular. Entretanto, a aceitação desta nova tecnologia no dia a dia do cidadão comum ainda é uma incógnita e uma aposta. Neste capítulo apresentamos os principais conceitos e desafios da área e seu uso em duas áreas de aplicação: simulação e visualização de informações.*

5.1. Introdução

Durante anos, o projeto de aplicações interativas se baseou unicamente na metáfora *desktop* e no uso de sistemas baseados em janelas para intermediar a comunicação humano-computador em soluções comerciais. A partir da década de noventa, porém, com o surgimento da Web, grande parte dos sistemas migraram para esta plataforma, se beneficiando das facilidades oferecidas (independência de máquina, acesso através de navegadores, etc.). Mais recentemente, os dispositivos móveis passaram também a ser uma plataforma

Vídeo com a apresentação do capítulo: `https://youtu.be/gbGGFYmAs2Q`

popular para hospedar aplicativos interativos, oferecendo interfaces mais intuitivas, naturais e inclusivas, acessíveis por uma parcela crescente da população.

Recentemente, com a popularização de dispositivos de realidade virtual de boa qualidade comercializados com preço semelhante a telefones celulares, vemos surgir o metaverso como uma nova plataforma de comunicação entre seres humanos e entre humanos e computadores. Ainda que estudos em realidade virtual e técnicas 3D de interação venham sendo desenvolvidos em laboratórios de pesquisa desde a década de 90, é atualmente que se reúnem as condições necessárias para o seu uso por empresas e pela população em geral.

Neste capítulo, aprofundamos o conceito de ambientes imersivos, ou seja, metaverso, e discutimos suas potencialidades como um espaço de interação que irá coexistir com plataformas desktop, Web e mobile. São discutidos os requisitos de hardware e software, o conceito de presença e imersão, exemplos de aplicações que se beneficiariam de uma plataforma imersiva, bem como aspectos éticos e sociais envolvidos.

São ainda abordados dois domínios de aplicação para a realidade virtual e aumentada: simuladores imersivos para ensino e treinamento de habilidades e ferramentas de visualização imersiva de informações.

O primeiro domínio é o da simulação em ambientes imersivos. Simuladores de voo já são largamente utilizados na formação de pilotos, sendo aceitos como um passo formal no processo de aprendizagem. Da mesma forma, simuladores de cirurgia são importantes no treinamento das habilidades necessárias a diversos procedimentos médicos, desde os mais simples até cirurgias minimamente invasivas. O desenvolvimento de simuladores imersivos tem grande potencial para contribuir na formação de estudantes e profissionais através do treinamento de atividades complexas e que envolvam habilidades espaciais importantes. Neste capítulo, são mostrados exemplos, identificados os elementos importantes no processo de concepção de simuladores imersivos, discutidas as potencialidades, limitações e oportunidades de concepção de novos produtos.

O segundo domínio é decorrente da crescente geração e disponibilização de dados. Ao mesmo tempo em que são gerados ou coletados grandes volumes de dados em pouco tempo, a transformação desses dados em conhecimento se torna cada vez mais difícil. Técnicas de visualização de informações têm sido propostas para facilitar a exploração e a análise e entendimento de dados [Munzner 2014]. Nesse contexto, a realidade virtual extrapola o limite de visualização estabelecido pelos displays 2D, permitindo a visualização de informações em 3D, com dados dispostos no espaço e facilmente acessíveis através de uma interação mais natural com os dados, posto que se passa em nosso ambiente e, frequentemente, com gestos. Enquanto as vantagens das técnicas de visualização em displays 2D para análise de dados levaram ao surgimento da área denominada *visual analytics* [Thomas and Cook 2005], no contexto de realidade virtual, temos as técnicas e aplicações hoje conhecidas como *immersive analytics* [Marriott et al. 2018b].

5.2. Realidade Virtual e Metaverso

O conceito de realidade virtual foi introduzido pela primeira vez por Sutherland em seu trabalho seminal “The Ultimate Display”, de 1965. Naquela época, Sutherland já sugeria

que um display poderia ser construído para fornecer imagens geradas por computador tão realistas que seriam indistinguíveis das coisas reais. Três anos depois, em 1968, ele produziu o primeiro capacete de visualização acoplado a um computador. Ele era composto por dois pequenos monitores de CRT montados em uma bandana e rastreava a posição da cabeça do usuário. A ideia por trás desse "novo" produto é reproduzida no parágrafo a seguir [Sutherland 1965]:

> *Não pense nisso como uma tela, pense nisso como uma janela, uma janela através da qual alguém olha para um mundo virtual. O desafio da computação gráfica é fazer com que esse mundo virtual tenha aparência real, soe real, se mova, dê respostas em tempo real, e até mesmo cause sensações reais.*

5.2.1. Realidade Virtual

Durante a última década, o termo "realidade virtual" tornou-se popular e tem sido usado indiscriminadamente para caracterizar desde simples aplicativos gráficos interativos até experiências totalmente imersivas, às vezes misturando objetos virtuais e reais. Percebendo a falta de uma taxonomia para distinguir tais aplicações interativas, Milgram e Kishino [Milgram and Kishino 1994] propuseram o continuum da virtualidade, que está relacionado com a mistura de classes de objetos apresentados em qualquer situação de exibição. Ambientes reais são mostrados em uma extremidade do continuum, enquanto ambientes virtuais estão na extremidade oposta.

Nesse contexto, esses autores propuseram um conceito claro e simples para a realidade virtual. Para eles, "um ambiente de Realidade Virtual (RV) é aquele em que o participante-observador está totalmente imerso e é capaz de interagir com um mundo completamente sintético". Em outras palavras, podemos dizer que a realidade virtual é um meio composto por simulações computadorizadas e interativas que detectam a posição e as ações do participante e substituem ou aumentam o feedback para um ou mais sentidos, proporcionando a sensação de estar mentalmente imerso ou presente na simulação (um mundo virtual).

Quatro elementos-chave são necessários para uma experiência de realidade virtual: mundo virtual, sensação de imersão, feedback sensorial e interatividade.

Enquanto o conceito de **mundo virtual** é bastante óbvio e se refere a um espaço imaginário composto por um conjunto de objetos acrescido de regras que governam esses objetos, a **imersão** nesse mundo pode ser tanto física quanto mental. O estado de imersão mental, ou seja, estar profundamente envolvido, é frequentemente referido como ter "um senso de presença" em um ambiente, sendo um desafio na área medir esse "um senso de presença" [Souza et al. 2021]. A imersão física, por outro lado, requer o uso de estímulos sintéticos por meio de tecnologia que ajuda o corpo a sentir o ambiente virtual.

O **feedback sensorial** é uma característica essencial das aplicações de realidade virtual. Um sistema de RV fornece feedback sensorial direto aos participantes com base em sua posição física. Na maioria dos casos, o feedback é visual, embora ambientes de realidade virtual ideais devam estimular todos os sentidos humanos (audição, visão, tato, olfato e paladar). Para gerar a saída sensorial do sistema de RV na posição do participante,

o sistema deve rastrear seus movimentos. Um sistema de RV típico rastreará a cabeça dos participantes e pelo menos as mãos ou objetos segurados pelas mãos.

Para que a realidade virtual pareça autêntica, ela deve responder às ações do usuário, ou seja, ser responsiva e, mais do que isso, **interativa**. Geralmente, os computadores permitem isso, mas alguns requisitos tecnológicos devem ser considerados: hardware de geração de imagens 3D em tempo real e geração de som estéreo de alta qualidade; dispositivos de entrada e saída específicos que simulam e estimulam os sentidos humanos; e software para simular ambientes virtuais, muitas vezes com alto nível de realismo, que permitem uma resposta imediata às ações do usuário. Esses requisitos tecnológicos compõem os elementos centrais das interfaces humano-computador em RV.

5.2.2. Metaverso

O termo "metaverso" se refere a um conceito que descreve um espaço virtual tridimensional coletivo e compartilhado, que tenta replicar ou simular a realidade através de dispositivos digitais. Ele representa a possibilidade de uma espécie de realidade paralela onde as pessoas podem interagir, criar, explorar e socializar [Wikipedia 2023]. A ideia central de metaverso é criar um ambiente digital imersivo e expansível, semelhante a um universo virtual, onde os usuários podem se conectar e interagir como se estivessem fisicamente presentes. Esses ambientes podem variar em termos de realismo, desde mundos virtuais altamente detalhados até espaços mais estilizados e simplificados. Em outras palavras, seria a Internet 3D populada por pessoas reais representadas pelos seus avatares.

Desta forma, ainda que o conceito esteja largamente difundido e o termo venha sendo empregado exaustivamente, na prática o metaverso ainda não existe. Deverá ser uma construção coletiva, a exemplo do que aconteceu anteriormente com a Web. No caso do metaverso, entretanto, são consideradas 7 camadas.

A primeira camada é a **infraestrutura**, a base técnica que dará suporte para todo o projeto, incluindo as tecnologias 5G (e 6G), já que velocidade, processamento e armazenamento em nuvem são essenciais. A segunda camada é a **interface** necessária para acessar o metaverso e inclui os óculos de realidade virtual e aumentada, smartphones e toda a tecnologia necessária para conectar os avatares digitais à experiência sensorial física dos usuários. O terceiro nível é a **descentralização** e envolve a tecnologia necessária para garantir a liberdade para que todas as pessoas naveguem de um ambiente a outro. É esperado que *blockchain*, NFTs e inteligência artificial tenham um papel importante neste nível.

A quarta camada está sendo chamada de **computação espacial** e visa garantir a integração do mundo virtual com o mundo real. Ela faz uso de realidade virtual, realidade aumentada, sensores, dispositivos e técnicas de interação 3D para fazer o mapeamento entre os dois mundos. A quinta camada é chamada de **economia de criação** e envolve uma infinidade de ferramentas de *design*, fluxos de trabalho personalizados, mercados de ativos, *assets*, etc. Existe a expectativa de que essa camada envolva a colaboração entre as empresas da área.

A sexta camada, **descoberta** permitirá que as empresas monetizem com os usuários através de publicidade, reviews, lojas, etc. Finalmente, a sétima e última camada é

denominada **experiência**. É nesta camada que o público é atraído e cativado através de um espaço de entretenimento e gamificação (ver Figura 5.1). A camada de experiência envolve o desenvolvimento e a disponibilização de conteúdo através das diversas aplicações. Estima-se que educação, entretenimento, trabalho colaborativo, e jogos são áreas que mais se beneficiarão do metaverso.

Figura 5.1. Espaço do INF-UFRGS no metaverso, criado com a ferramenta Mozilla Hubs.

Para que o metaverso seja uma realidade, pesquisadores, empresas de tecnologia e desenvolvedores independentes estão explorando ativamente esse conceito e trabalhando separadamente nas sete camadas identificadas (ver Figura 5.2).

Nas próximas seções, apresentaremos duas áreas de trabalho que estão sendo exploradas em realidade virtual e que, no futuro, potencialmente se enquadrarão na camada sete do metaverso.

5.3. Simuladores Imersivos

Sistemas de realidade virtual têm sido amplamente utilizados para treinar profissionais em áreas tão diversas como medicina, indústria e combate a incêndios, bem como educação e saúde. Essas aplicações são apresentadas como jogos com um propósito sério, simplesmente *jogos sérios (SGs)* ou ainda *jogos aplicados*. Sawyer [Sawyer 2007] os define como "qualquer uso significativo de recursos de jogos informatizados ou da indústria de jogos cuja missão principal não seja o entretenimento". Zyda [Zyda 2005] descreve como sendo "um desafio mental, jogado com um computador de acordo com regras espe-

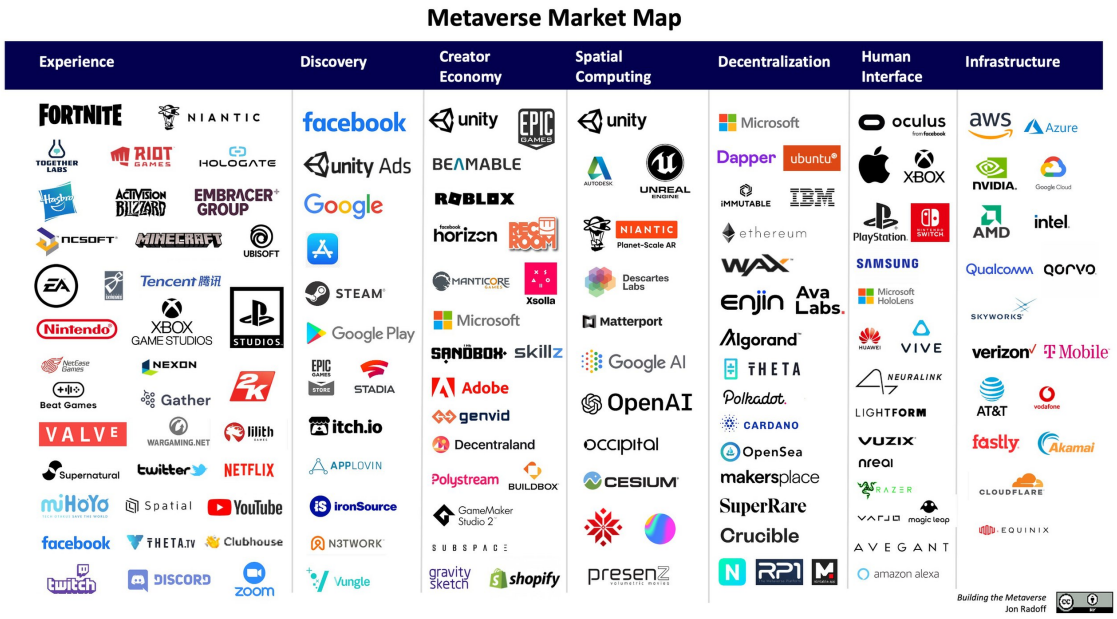


Figura 5.2. As sete camadas do metaverso e as empresas-chave que atuam em cada uma. Fonte: [Wikipedia 2023].

cíficas, que usa o entretenimento para promover objetivos de treinamento governamentais ou corporativos, educação, saúde, políticas públicas e comunicação estratégica". Finalmente, de acordo com Michael e Chen [Michael and Chen 2005], são "jogos que não têm entretenimento ou diversão como seu principal objetivo".

Assim como os jogos de vídeo, os SGs envolvem em jogabilidade, desafio, interação e objetivo, enquanto as aplicações *gamificadas* incorporam apenas elementos de jogos [Deterding et al. 2011]. Segundo Deterding et al. [Deterding et al. 2011], gamificação é definida como "o uso de elementos de design de jogos em contextos não relacionados a jogos". Esses elementos podem estar relacionados a:

- componentes de *design* de interação e soluções de *design* para um problema conhecido em um contexto, como crachá, classificação e nível;
- componentes de jogabilidade, como limite de tempo, recursos limitados e rodadas;
- diretrizes avaliativas, como um jogo duradouro e objetivos claros;
- modelos conceituais de componentes de jogos, como desafio, fantasia e curiosidade; e
- práticas e processos, como testes de jogo e *design* centrado no jogo.

Além disso, os SGs são construídos sobre estruturas pedagógicas e educacionais, que definem a relação entre aprendizado e mecanismos de jogo, garantindo uma combinação bem sucedida desses fatores para alcançar seu propósito sério. O *framework* apresentado por Ibáñez et al. [Ibanez et al. 2011] abrange seis facetas do desenvolvimento de SGs:

- objetivos de aprendizado, que definem um quadro de referência do domínio a ser ensinado;
- simulação de domínio, que define um modelo formal estabelecendo as bases da simulação;
- interação com a simulação de domínio, que é o coração da metáfora;
- problemas e progressão, que definem o nível de dificuldade e a progressão de habilidades;
- decoração, que descreve como entreter e envolver o jogador; e
- implantação, que descreve as condições de uso para preservar as qualidades de aprendizado do jogo.

Técnicas de *design* de jogos podem ser úteis para criar aplicativos de RV mais envolventes e que sejam, ao mesmo tempo, mais atrativos para o público. Essas aplicações, combinadas com a crescente disponibilidade de óculos de realidade virtual (HMDs) para consumidores, podem levar a um futuro em que as pessoas treinem e aprendam usando a RV imersiva. Níveis mais elevados de imersão têm apresentado efeitos positivos em julgamentos espaciais em pequena escala e memorização. A combinação desses fatores leva à memorização de procedimentos complexos, o que permite que o ambiente virtual produza resultados de treinamento e aprendizado. Bowman et al. [Bowman et al. 2009] fornecem evidências empíricas de que um alto nível de imersão também pode produzir uma melhoria mensurável no desempenho de uma atividade mental abstrata. Chalmers et al. [Chalmers and Debattista 2009] afirmam que um alto nível de realismo é necessário para garantir que o treinamento e a aprendizagem em ambientes virtuais sejam equivalentes ao mundo real. Eles também devem ser capazes de simular todos os sentidos humanos simultaneamente.

A RV imersiva combinada com técnicas usadas em jogos leva ao desenvolvimento de simulações imersivas com aspectos de jogo que permitem o envolvimento em atividades de aprendizado que de outra forma seriam caras ou muito perigosas, difíceis ou impraticáveis de implementar em sala de aula. Seu uso ajuda a mudar o relacionamento das pessoas com as informações, incentivando a visualização, experimentação e criatividade. Elas são flexíveis e complexas o suficiente para atender a diferentes estilos de aprendizado e ampliar a exposição de diferentes pessoas e perspectivas, incentivando a colaboração e apoiando discussões significativas após o jogo.

5.3.1. Simulação para Análise de Risco

O Ministério do Trabalho e Previdência do Brasil registra oficialmente mais de 700.000 acidentes de trabalho a cada ano. Cerca de 2.800 trabalhadores morrem como consequência desses acidentes, enquanto 15.000 ficam permanentemente incapacitados. Esses números não contabilizam muitos outros casos que não são oficialmente reportados. Além do custo intangível em vidas, o governo e grandes empresas gastam mais de 30 bilhões de dólares por ano com as consequências desse tipo de acidente. Nesse contexto, o setor industrial e as empresas de serviços públicos têm investido em projetos inovadores para

melhorar a segurança no trabalho. O uso de ambientes virtuais para treinar procedimentos seguros é uma prática cada vez mais difundida. Em muitas áreas, no entanto, mais do que procedimentos bem projetados e uma equipe bem treinada usando equipamentos de proteção individual (EPI), uma série de fatores humanos é crucial para um comportamento seguro.

Em termos de fatores humanos, a capacidade de perceber riscos está relacionada com cada indivíduo, suas crenças, motivações e relações com outras pessoas. Em outras palavras, as pessoas reagem de acordo com seu modelo mental de uma situação potencialmente arriscada, em vez do risco real em si [Asnar and Zannone 2008]. Por exemplo, uma lixeira caída em um corredor ou calçada representa um risco para algumas pessoas. No entanto, muitos nem notariam a sua presença até tropeçarem nela ou simplesmente andarem ao seu redor sem guardar nenhuma memória do fato.

Os simuladores de realidade virtual (RV) têm sido usados em muitas áreas para treinar habilidades especializadas, incluindo a habilidade de perceber situações perigosas. Exemplos típicos são simuladores de voo, cirurgia e direção. Ambientes virtuais imersivos, como esses, são criados para reproduzir o mais precisamente possível os ambientes reais. Essa precisão colabora para impor ao sujeito imerso a sensação de presença, que, por sua vez, está relacionada a um comportamento fiel ou pelo menos plausível em relação ao comportamento da mesma pessoa em um ambiente real.

Entendendo a gravidade das consequências dos acidentes de trabalho causados por falta de treinamento ou mesmo imprudência, propusemos o design e uso de simuladores imersivos para avaliar continuamente a capacidade de comportamento seguro entre os trabalhadores. Propusemos um framework para o desenvolvimento de simuladores para avaliação e treinamento de percepção de riscos. Nosso framework foi inicialmente utilizado, como descrito em trabalhos anteriores [Jorge et al. 2013] para construir um simulador para a análise de percepção de riscos. Este primeiro simulador foi projetado para avaliar periodicamente a capacidade dos trabalhadores de perceber riscos em diferentes cenários. A Figura 5.3 ilustra alguns dos cenários simulados. Em uma sessão típica de simulação, o usuário utiliza um óculos de realidade virtual (HMD) e se move usando um controle de jogo. Quando ele ou ela detecta elementos de risco potencial no cenário, eles devem ser selecionados com o controle ou um gesto. No final da sessão, é emitido um relatório que indica os objetos selecionados, aponta os riscos não detectados e exibe o percurso percorrido. Este simulador também permitiu o desenvolvimento de novas métricas para percepção de riscos e desempenho da tarefa, contribuindo tanto para a segurança quanto para a eficiência das atividades.

Foram realizadas várias simulações experimentais com diversos grupos de participantes para aprimorar as ferramentas e métodos. Isso ajudou a generalizar o simulador na forma de um framework que serve de base para o desenvolvimento de uma pluralidade de outros simuladores focados em outras questões relacionadas ao risco.

O framework é baseado em vários equipamentos de interface alternativos, como HMDs, sensores de gestos, gamepads, entre outros. Neste trabalho, o interesse era em desenvolver um sistema que fosse totalmente imersivo, portátil e de fácil integração. Com essas premissas em mente, foram escolhidos dispositivos e ambientes de programação que pudessem fornecer essas características, minimizando a necessidade de recursos de

Figura 5.3. Três imagens dos cenários usados no simulador de análise de risco: escritórios, substituição de pára-raios, planta de uma subestação de energia [Nedel et al. 2016].

desenvolvimento mais elaborados. Um desses ambientes é um motor de jogo. Os motores de jogo oferecem muitos recursos para simplificar o desenvolvimento de Ambientes Virtuais. Além disso, os objetivos de um trabalhador no seu local de trabalho frequentemente podem ser gamificados (traduzidos em objetivos de jogo). Adicionalmente, as imagens necessárias para criar vários efeitos visuais são facilmente implementadas usando um motor de jogo, que tem a vantagem de mapear eventos (por exemplo, os eventos gerados por dispositivos de entrada de RV), de forma transparente. Neste projeto, o framework foi implementado tanto na Unreal Development Kit quanto na Unity3D.

5.3.2. Explorando Percepção e Aquisição de Conhecimento em Simuladores

Embora ambientes virtuais imersivos venham sendo usados por anos para fins de treinamento e aprendizado (por exemplo, simuladores de voo e de cirurgias), os efeitos do uso de dispositivos de realidade virtual em sessões de simulação ainda não foram completamente compreendidos e, em parte, isso se deve a baixa maturidade destes dispositivos.

Em função disso, neste trabalho foram explorados os efeitos de diferentes dispositivos de realidade virtual em simuladores desenvolvidos para treinamento, com foco nos aspectos de percepção e ganho de conhecimento. Foram realizados dois estudos com usuário para investigar a influência desses dispositivos na carga de trabalho dos usuários, no enjoo causado pelo movimento (*cybersickness*) e no desempenho no domínio do treinamento em segurança no trabalho.

A percepção e o aprendizado estão intrinsecamente ligados: em tarefas de treinamento, não é possível aprender um novo procedimento sem perceber o ambiente ao seu redor. Portanto, este estudo se concentra em investigar os efeitos de diferentes tecnologias de realidade virtual (RV) nos aspectos de percepção e aprendizado das simulações imersivas. Para reduzir a carga cognitiva possivelmente gerada pela combinação de todas essas tecnologias, dividimos o estudo em dois experimentos com usuários: (i) percepção do usuário, comparando três dispositivos de exibição, e (ii) aquisição de conhecimento, comparando quatro combinações de técnicas de interação e locomoção seminaturais e não naturais.

Em primeiro lugar, o objetivo de uma simulação de treinamento é preparar o usuário para uma situação do mundo real, que naturalmente possui alta complexidade visual. Portanto, foram construídos vários simuladores com base em cenários realistas, que suportam a transferência do que foi aprendido durante a simulação para a situação do mundo

real. Foi utilizado um simulador para avaliação de percepção de riscos (Figura 5.3a), cujo objetivo é treinar os trabalhadores a detectar elementos de risco em um ambiente de trabalho normal [Nedel et al. 2016], e um simulador para substituição de pára-raios (Figura 5.3b), que visa treinar profissionais em procedimentos básicos de segurança para instalações elétricas em postes de utilidade pública.

O simulador de avaliação de percepção de riscos reforça a percepção dos usuários por meio de sua capacidade de ver, ouvir ou se conscientizar de algo por meio dos sentidos para apreender seu ambiente, detectando e evitando riscos. Assim, o simulador treina os usuários por meio de aprendizado perceptual, que compreende a capacidade de detectar informações (ou seja, eventos, características distintivas e *affordances*) oferecidas pelo ambiente [Adolph and Kretch 2015]. Ragan et al. [Ragan et al. 2015] já haviam avaliado os efeitos de diferentes níveis de campo de visão (em um dispositivo HMD) em uma tarefa de digitalização usando cenários realistas, que não apresentaram efeito significativo na detecção de alvos ou no uso de estratégias de avaliação. Para ampliar a compreensão do impacto da RV na aprendizagem perceptual, investigamos como diferentes dispositivos de exibição que fornecem diferentes campos de visão e diferentes configurações de uso (mais ou menos confortáveis para o usuário) podem afetar a experiência e o desempenho do usuário.

Além de adquirir informações, o usuário deve ser capaz de reter as informações obtidas por meio das simulações de treinamento para transferi-las para a situação do mundo real. Lembrar informações também é conhecido como conhecimento e pode ser classificado em diferentes categorias, como factual, conceitual, procedural e metacognitivo [Krathwohl 2002]. Ambos os simuladores requerem que o usuário reconheça detalhes ou elementos específicos (ou seja, conhecimento factual). Com o simulador para substituição de pára-raios, foi explorada a retenção de conhecimento, sendo investigados os efeitos de diferentes técnicas de interação e locomoção para agarrar e manipular, que são fundamentais em inúmeras simulações de treinamento (por exemplo, combate a incêndios, treinamento militar, etc). Em termos de interação, foi rastreado o movimento do usuário para mapear a retenção de conhecimento para o mundo real, fornecendo experiências de alta fidelidade ao usuário.

Em relação à locomoção, foi considerado o uso do rastreamento de movimento, mas as soluções encontradas para rastrear os grandes espaços físicos necessários para ambas as simulações (Figura 5.3) eram muito caras [31]. Técnicas de navegação hipernaturais, como "Seven League Boots"[32], poderiam melhorar o desempenho em termos de velocidade, mas podem ser difíceis de controlar quando é necessária precisão [20]. Portanto, os esforços foram concentrados em técnicas de navegação menos naturais, como "walking-in-place"(WIP, andar no lugar) e navegação por joystick. Especificamente, a placa de equilíbrio do Wii tem sido amplamente usada por pesquisadores para fornecer técnicas de WIP de baixo custo, mostrando efeitos positivos no desempenho do usuário em espaços humanos e orientação espacial, ao mesmo tempo em que proporciona uma alta sensação de presença [33].

O primeiro experimento envolveu 61 participantes e procurou entender se e como displays de RV com diferentes campos de visão afetam a capacidade dos usuários de identificar riscos em um ambiente virtual semelhante a um escritório (ou seja, foco na

percepção do usuário).

Posteriormente, num segundo experimento com 46 participantes foi avaliado se e como técnicas de interação que oferecem diferentes graus de liberdade influenciam na capacidade dos usuários de aprender tarefas procedimentais (ou seja, foco no ganho de conhecimento).

A partir dos resultados obtidos, foi encontrado que o conhecimento dos usuários sobre o tópico da simulação (ou seja, segurança do trabalho) e a experiência em jogos desempenham um papel importante em simulações imersivas, e que sintomas de *cybersickness*, como desorientação, provavelmente são causados pela falta de consciência do ambiente real e não pelo conteúdo exibido no ambiente imersivo. Mais detalhes sobre este estudo podem ser encontrados em [Menin et al. 2021].

5.3.3. Treinamento Comportamental na Área da Saúde

O ensino em áreas da saúde objetiva a formação geral do estudante, capacitando-o na prestação de assistência aos problemas mais prevalentes, encaminhamento adequado a níveis mais complexos quando indicado, tomada de decisão e preservação da vida em situações de urgência e emergência. Para estas atividades, alunos e profissionais da saúde necessitam de treinamentos de habilidades técnicas e não-técnicas, conhecidas como *soft skills*.

Um quadro de ensino adequado aos dias atuais deve incluir: "Saiba, Veja, Pratique, Prove, Faça e Mantenha" [Sawyer et al. 2015]. Isso começa com o aluno adquirindo conhecimento cognitivo (Saiba) e observando o procedimento (Veja). Após, progride para a fase de aquisição de habilidades psicomotoras e prática livre do procedimento em um simulador (Pratique). A simulação também é utilizada para permitir que o aluno prove sua competência antes de realizar o procedimento em um paciente real (Prove). Uma vez que a competência é demonstrada, o aluno teria permissão para realizar o procedimento em pacientes com supervisão direta, até que possa ser liberado para realizar o procedimento de forma independente (Faça). A manutenção da habilidade é garantida através da prática clínica contínua, agregada a simulação conforme necessário (Mantenha).

Há carência de sistemas que ofereçam treinamento e parâmetros avaliativos para procedimentos básicos de saúde nas fases iniciais dos cursos universitários. Nesta etapa a formação dos profissionais de saúde está voltada para a relação com o paciente, como a condução de consultas, procedimentos clínicos, realização dos exames físicos necessários, investigação, detecção de patologias e diagnóstico.

Em grande parte dos casos, quando há um hospital escola vinculado a instituição de ensino, os estudantes das áreas da saúde começam apenas observando os procedimentos e ações. Em seguida, passam para o contato direto com o paciente real, sendo observado por um professor preceptor que verifica o aprendizado, o que pode gerar um desconforto para o paciente e insegurança para o aluno. Além disso, muitas instituições de ensino não têm hospitais escola vinculados, restringindo as atividades práticas apenas aos anos finais dos cursos, nos internatos, o que impede a prática dos alunos nos semestres iniciais, importante na garantia da confiança e dos conhecimentos do profissional.

Nesse contexto, está em desenvolvimento um sistema denominado MetaHealth

que concretiza um modelo de ensino, atualização e acompanhamento das habilidades de profissionais de saúde e alunos em ambientes hospitalares [Okuda et al. 2009] utilizando realidade virtual e explorando o conceito de metaverso. O modelo proposto envolve um conjunto de simuladores imersivos que utilizam a metodologia OSCE (Objective Structured Clinical Examination) [Par 2020] para avaliação de *soft skills* e um hub integrador ao qual estes simuladores são conectados.

O hub MetaHealth é dividido em duas partes. Através de um plataforma web, professores e preceptores podem preparar o treinamento de seus estudantes criando estações OSCE (cenários para treinamento e avaliação), indicando sequências de estações a serem realizadas e acompanhando o desempenho dos estudantes após a execução do treinamento. Os estudantes utilizam a plataforma Web para identificar suas sessões de treinamento e aprendizagem e acompanhar seu desempenho. A segunda parte do hub MetaHealth é um portal em RV que serve para ambientar o usuário no cenário 3D imersivo, aferir sua aptidão perceptuo-motora, adaptar os simuladores às condições do usuário e conduzir o estudante ao seu treinamento imersivo, através do uso de simuladores específicos.

O MetaHealth abriga diferentes simuladores imersivos em ambientes clínicos que podem ser administrados e gerenciados conforme a necessidade do componente curricular do curso de saúde, como cenários para Semiologia, Urgência e Emergência, etc. Nesses cenários o usuário pode interagir com pacientes virtuais através de perguntas e utilizar os equipamentos e ferramentas disponíveis no ambiente, com retornos autênticos das interações realizadas pelo usuários, de acordo com os parâmetros definidos pelo professor na criação dos casos.

Um desses casos, criado num simulador no MetaHealth, corresponde a uma estação OSCE envolvendo uma clínica pediátrica simulada. Apresentaram-se tarefas normalmente exigidas durante uma consulta médica, o que ajuda os alunos a treinar habilidades básicas antes do contato com pacientes reais. Tais tarefas incluiram procedimentos de biossegurança e anamnese por meio da coleta de detalhes sobre a doença, sinais e sintomas e histórico médico do paciente. O estudante também devia ser capaz de realizar exames físicos, interpretar os achados e compilar todas as informações para determinar o diagnóstico correto e o tratamento adequado.

Para esse estudo, propusemos uma tarefa de exame médico onde o usuário deve atingir quatro metas durante a consulta. No princípio, o usuário deve investigar corretamente o histórico médico do paciente, os sintomas que ele apresenta e a queixa principal através das questões da anamnese. Em seguida, com base na anamnese, deverá realizar os exames necessários para confirmar os achados. As duas últimas metas são o diagnóstico correto da doença da criança e o tratamento recomendado. As informações do caso e do paciente apresentadas em nosso cenário de estudo foram desenvolvidas e revisadas por um pediatra. Assim, ao chegar no consultório, o usuário segue o protocolo médico e conduz os procedimentos como na vida real, coletando informações do paciente por meio de exames e anamnese, sempre tendo em mente as preocupações de biossegurança. O ambiente dispõe de instrumentos para exame, como otoscópio, estetoscópio e abaixador de língua, uma pia e luvas que podem ser utilizadas. Há também *tablets* e monitores virtuais onde o usuário indica as questões que quer fazer para o paciente, confirma diagnósticos e

Figura 5.4. Paciente durante a anamnese (esquerda) e visualização da pia e dos instrumentos médicos disponíveis (direita).

Figura 5.5. Usuário conduzindo um exame. Vista do exame da garganta. Vista do exame do ouvido.

tratamentos.

No consultório médico, o usuário encontra um paciente pediátrico e sua mãe (Fig. 5.4-esquerda). O usuário deve fazer perguntas sobre a saúde e a situação da criança, escolhidas a partir de uma lista de perguntas pré-definidas exibidas em um tablet virtual. As perguntas pré-definidas são embaralhadas, contendo multiplos tipos de questionamentos diferentes, para que não fique óbvio qual é o caminho a se seguir durante a entrevista. Além da anamnese e histórico médico, o usuário também pode realizar três exames físicos no paciente: a ausculta, tanto cardíaca quanto pulmonar, oroscopia e otoscopia, conforme mostra a figura 5.4-direita. Na ausculta, o usuário deve posicionar o estetoscópio nos pontos corretos e ouvir os sons do coração e dos pulmões. Na oroscopia (Fig. 5.5), o usuário deve segurar o abaixador de língua próximo à boca do paciente para abri-la possibilitando a visualização da garganta. A otoscopia é semelhante; o usuário deve segurar o otoscópio próximo ao ouvido do paciente para visualizar o canal auditivo.

Uma avaliação preliminar do MetaHealth foi realizada com três usuários de perfis diferentes, seguindo o protocolo *think aloud*. Um deles, a quem chamamos de EXPERT, é um pediatra experiente que realiza essa atividade diariamente. O outro, a quem chamamos de RESIDENTE, é médico fazendo residência em pediatria com 5 meses de conhecimento. O último perfil de usuário, que chamamos de ESTUDANTE de uma faculdade

de medicina, acaba de começar a aprender os fundamentos do atendimento médico.

A análise dos dados coletados durante o experimento mostrou que os usuários geralmente se sentiram confiantes em seu desempenho na tarefa de consulta médica. Eles não acharam difícil interagir com os instrumentos médicos e manipular os objetos nas salas virtuais. Afirmaram também que a realização dos exames físicos do paciente foi fácil. Porém, em relação à navegação, os usuários sentiram que o movimento por meio do joystick e da navegação direta era muito rápido. O usuário EXPERT demonstrou sinais de tontura e enjoo, o que obrigou o participante a utilizar apenas a técnica de teletransporte para navegação durante a sessão experimental. Considerando as respostas referentes à dificuldade de navegação pelos três métodos, o movimento livre baseado em joystick foi considerado mais difícil que a técnica de teletransporte e o movimento no mundo real. Esse resultado já era esperado. Agora, com essa evidência, podemos enfatizar os outros métodos de navegação.

A principal lição aprendida com este experimento é como a aplicação em RV pode auxiliar no ensino de habilidades clínicas básicas e na condução de anamnese. Todos os participantes relataram que a visualização do ambiente e a fidelidade do cenário proposto são adequadas às suas atividades. Os resultados de tempo mostraram que os usuários concluíram a consulta médica, realizando todos os exames e etapas em tempo razoável, equivalente a uma consulta comum na unidade básica de saúde na vida real. Além disso, as ações exigidas no cenário, como exame físico e anamnese que todos os usuários realizaram com sucesso, aliadas à confiança relatada e menor dificuldade na manipulação dos instrumentos, sugerem que a investigação da doença e o exame do paciente neste ambiente virtual acompanhe os estudos práticos ministrados em cursos de Medicina. Maiores detalhes sobre o MetaHealth podem ser obtidos em [Negrão et al. 2023].

5.4. Visualização Imersiva

Recentemente técnicas interativas de visualização dados se tornaram populares em função do uso de infográficos interativos pelos meios de comunicação em massa. Entretanto, desde a década de 80, elas vem sendo desenvolvidas com o objetivo de auxiliar o processo de análise e compreensão de dados em diversos domínios de aplicação. Em um relatório seminal, McCormick et al. [McCormick et al. 1987] afirmaram que o objetivo da visualização é potencializar os métodos analíticos existentes, fornecendo novas perspectivas por meio de métodos visuais. Como a maioria das aplicações de visualização, à época, eram científicas ou de engenharia, envolvendo simulações físicas num domínio espaço-temporal, logo ficaram evidentes as vantagens de RV na visualização desses processos.

Assim, por décadas, infraestruturas de RV têm sido utilizadas para auxiliar cientistas e engenheiros na análise e compreensão de seus conjuntos de dados complexos (Marai et al. 2019). A título de exemplo dessa longevidade de utilização citamos o túnel de vento virtual [Bryson and Levit 1992] (Fig. 5.6), que permitia explorar numericamente campos vetoriais representando um fluxo simulado no qual o engenheiro inseria partículas com uma luva e observava o comportamento das mesmas em ambiente imersivo.

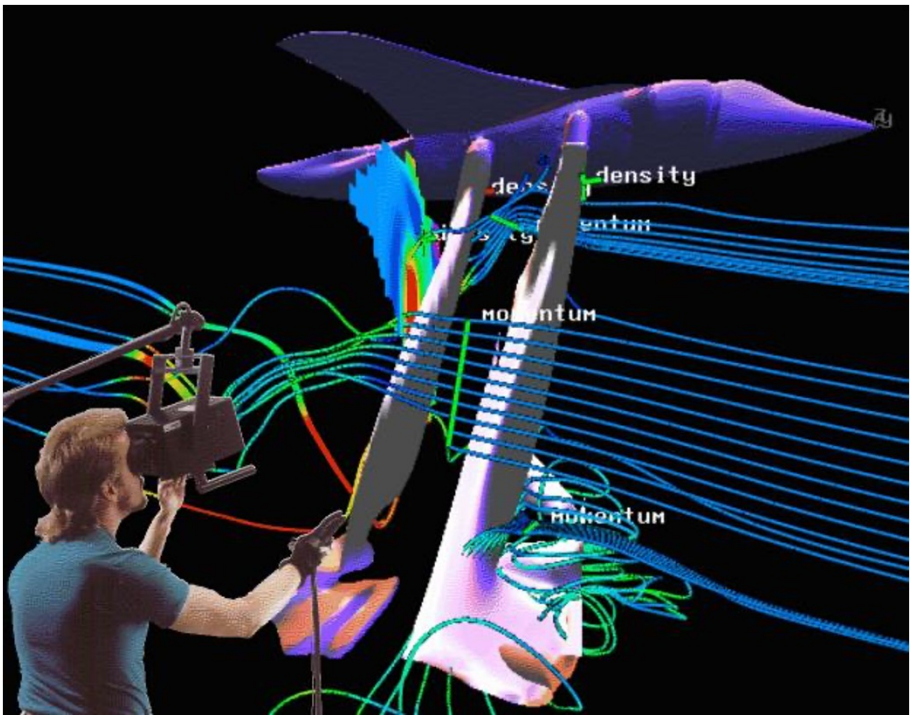


Figura 5.6. Túnel de vento virtual [Bryson and Levit 1992]. Foto obtida de https://www.nas.nasa.gov/Software/VWT/vwt.html.

5.4.1. Aspectos Essenciais para Visualização Imersiva

O desenvolvimento de aplicações de visualização imersiva pressupõe cinco aspectos principais [Marriott et al. 2018a], que constituem um framework estendido daquele proposto por Brehmer e Munzner [Brehmer and Munzner 2013] para aplicações de visualização de informações. O framework original de Brehmer e Munzner pressupõe a consideração de três aspectos: (i) ***what***, ou seja, quais **dados** serão visualizados; (ii) ***why***, porque serão visualizados, ou seja, que tarefas de usuário serão suportadas); e (iii) ***how***, como efetivamente serão as representações visuais interativas implementadas.

Já o framework estendido para visualização imersiva acrescenta dois novos aspectos (***who*** e ***where***), ficando como segue:

- *What*: Diz respeito aos dados que serão visualizados e como estão organizados. No framework original [Brehmer and Munzner 2013], estão identificados cinco tipos de *datasets*: (i) tabelas (dados multidimensionais ou multivariados), (ii) redes (grafos e árvores); (iii) dados espaciais; (iv) geometria; e (v) qualquer coleção possível de itens, como agrupamentos, conjuntos e listas. Ainda, os dados podem ser estáticos ou dinâmicos se os dados estiverem disponíveis na forma de um fluxo contínuo. Quanto aos dados em si, esses podem ser itens de dados, nodos, relações entre nodos ou posições espaciais, cada um desses tendo atributos.

- *Why*: Corresponde à motivação para visualização, ou seja, quais as tarefas que o usuário vai realizar sobre os dados, com o suporte da visualização interativa. Qualquer tarefa do usuário envolve ações e alvos (os dados) dessas ações. O framework

divide as ações em três níveis. Ações de alto nível correspondem ao que entendemos como a principal tarefa do usuário ao visualizar dados: analisá-los. Ações de nível intermediário correspondem às buscas que o usuário pode realizar sobre os dados enquanto ações de baixo nível envolvem consultas específicas, comparações ou sumarização dos dados resultantes das buscas. Quanto aos alvos das ações do usuário, estes podem ser todo o conjunto de dados ou atributos específicos.

- *Who*: Relacionado a quem são as pessoas ou times de pessoas que vão usar o sistema, suas características e necessidades, assim como os diferentes tipos de colaboração. A aplicação vai ser usada por um único analista, um grupo de analistas, ou ela é projetada para comunicar dados para uma comunidade?

- *Where*: Relacionado às diferentes capacidades de interação e display, como diferentes graus de imersão ou conhecimento do mundo, e às características do ambiente físico onde a aplicação será usada. Onde o sistema será usado, incluindo em que tipo de plataforma? O sistema será usado em um ambiente controlado ou em campo?

- *How*: Corresponde às escolhas de *design* para implementar uma técnica de visualização. Elas podem ser divididas em quatro classes principais: *encode* (codificação), *manipulate* (manipulação), *facet* (segmentação) e *reduce* (redução). Codificação abrange a disposição (codificação espacial) das marcas gráficas que representam os itens de dados e o mapeamento (codificação visual) dos valores dos atributos dos itens de dados para canais visuais como cor, tamanho, ângulo, curvatura, forma e movimento dessas marcas. Esse aspecto envolve também a fidelidade da representação gráfica. A segmentação corresponde às diferentes maneiras de dividir os dados em múltiplas visualizações (*views*) e, no caso de visualização imersiva, como posicionar essas *views* no ambiente 3D. . Já a manipulação se refere à alteração de uma visualização ao longo do tempo e é fundamental para apoiar as tarefas dos usuários. Refere-se a qualquer modificação na visualização atual do conjunto de dados e pode ser baseada na alteração (i) da codificação visual ou espacial, (ii) da disposição dos itens na visualização e (iii) do número de itens ou atributos mostrados na visualização. Finalmente, a redução de itens e atributos pode ser obtida por filtragem ou agregação, geralmente com base em alguns atributos. No framework estendido para visualização imersiva, o componente *How* incorpora ainda o uso de modelos computacionais para capturar melhor todos aspectos de *human-in-the-loop analytics* e modelos de aprendizado de máquina e suporte à decisão baseado em otimização em que a visualização interativa é usada para entender e refinar modelos computacionais.

Em maior ou menor grau, esses cinco aspectos aparecem nos dois exemplos de visualização de dados imersiva descritos a seguir.

5.4.2. Exploração de Múltiplas Visualizações 3D Coordenadas

Ao longo dos anos, aplicações de visualização de dados têm adotado múltiplas visualizações coordenadas seja para exibir diferentes visões do mesmo conjunto de dados ou comparar diferentes conjuntos de dados de acordo com alguma característica que tenham

em comum. Ao mesmo tempo, as técnicas de interação para exploração de visualizações 3D também têm sido largamente estudadas.

Entretanto, em relação a múltiplas visualizações, estudos anteriores mostraram que a interação com múltiplas visualizações 3D em telas 2D (seja desktop ou display walls) não atende aos critérios de usabilidade. Essa falta de usabilidade poderia ser superada se a exploração ocorresse em ambientes imersivos, onde o usuário tem um grau extra de liberdade para interagir com visualizações 3D. Além disso, a consciência espacial humana e as capacidades organizacionais podem auxiliar o processo analítico realizado interativamente com as visualizações. Abordagens de análise imersiva têm se aproveitado dessas características.

Com o objetivo de melhorar a interação com várias visualizações coordenadas em ambientes imersivos, desenvolvemos a abordagem *Spaces* (Espaços), representada por um cubo virtual para manipular visualizações tridimensionais. A abordagem corresponde a uma versão 3D das interfaces gráficas do usuário WIMP (janelas, ícones, menus, apontadores). Nessa abordagem, há dois modos de interação, ilustrados na figura 5.7. Os Espaços podem ser agarrados e sobrepostos para facilitar a comparação dos dados representados dentro de cada um (modo macro). As duas mãos virtuais são independentes entre si: o usuário pode ”agarrar” um Espaço com uma mão e explorar suas informações com a outra (modo micro).

Figura 5.7. Modos de interação macro (esquerda) e micro (centro) para apoiar a exploração de múltiplos *Spaces* coordenados. A abordagem permite a exploração de múltiplos Espaços coordenados (direita).

Para avaliar nossa abordagem, formulamos hipóteses inspiradas nos problemas descritos em estudos de múltiplas visões coordenadas relatados na literatura, e projetamos uma versão de desktop semelhante a uma versão de RV e decidimos focar em um primeiro estudo na seguinte pergunta de pesquisa: *A nossa abordagem Spaces melhora a manipulação de múltiplas visualizações 3D coordenadas quando são exploradas em um ambiente virtual imersivo? Como a abordagem difere de uma versão de desktop 3D convencional?*.

Para avaliar a abordagem *Spaces* nessas duas versões, conduzimos um estudo com usuários com 19 participantes (Fig. 5.8). O caso de uso para testar as hipóteses foi a exploração de um conhecido conjunto de dados de músicas. porque não demandaria muito esforço de aprendizado dos participantes. As visualizações implementadas são gráficos de dispersão 3D de faixas musicais, artistas e gêneros, obtidos a partir de uma técnica de projeção multidimensional, e gráficos de barras que mostram o número de faixas por ano, artista e gênero. A visualização principal é um gráfico de dispersão que mostra faixas

Figura 5.8. O ambiente de desktop usado no experimento (esquerda) e em RV (direita): a sala virtual em ambos os casos contém uma TV que exibia as tarefas a serem realizadas pelos participantes.

musicais, e as outras visualizações funcionam como filtros. Cada visualização resultante é exibida em um Espaço. Foram propostas três tarefas envolvendo seleção. Os ambientes de avaliação são vistos na figura 5.8. Na versão desktop, os usuários interagiram com as visualizações usando teclado + mouse, enquanto no ambiente de RV eles usaram controladores como mãos virtuais. Em ambos os casos, os participantes começaram a exploração no centro do ambiente, e as visualizações eram exibidas ao redor deles.

Os resultados mostraram que a versão *desktop* não é significativamente melhor do que a versão imersiva em termos de tempo e precisão, apesar de usar a abordagem FPS padrão com teclado e mouse.

Múltiplas visualizações 3D de dados não são normalmente usadas em versões *desktop*, e esse pode ser o motivo dos resultados não significativos. Entretanto, os resultados subjetivos mostraram que nossa abordagem imersiva é significativamente melhor do que a versão *desktop*. Como conclusão, inferimos que os participantes não são capazes de explorar múltiplas visualizações 3D com dispositivos de interação comuns em *desktop*. Maiores detalhes desse estudo podem ser encontrados em [Quijano-Chavez et al. 2021].

Uma vez que, nesse primeiro estudo, obtivemos resultados positivos, melhoramos os recursos interativos incluindo interação próxima e distante (com raio virtual) e navegação virtual, e os utilizamos para avaliar três técnicas de visualização diferentes em um ambiente totalmente imersivo.

Este segundo estudo avaliou a eficácia de três variantes em tarefas de análise de tendências usando RV e interação 3D. As técnicas de visualização utilizadas nesse estudo foram *Small Multiples*, *Overlaid Trails* (versões estáticas) e *Animation* (versão animada). A abordagem *Spaces* foi melhorada para incluir interações adicionais com essas técnicas. A pergunta de pesquisa foi: *As variantes de gráficos de dispersão 3D, como Small Multiples, Overlaid Trails (versões estáticas) e Animation (versão animada) levam à detecção de tendências quando são exploradas em um ambiente imersivo? Como elas diferem?*

Desse segundo estudo participaram 18 usuários e foi possível comparar a execução de tarefas específicas com cada técnica de visualização em relação ao tempo, precisão e preferências subjetivas. Além disso, incluímos uma cena com todas as três técnicas de

visualização como a última fase do experimento para analisar as escolhas e preferências do usuário. Foram empregados dois conjuntos de dados: o primeiro foi adotado do trabalho de Brehmer et al. [Brehmer et al. 2019], contendo indicadores econômicos e de saúde pública para 16 nações ao longo de 26 anos (de 1975 a 2000). Foram definidas 9 tarefas replicando os comportamentos das tarefas de trabalho anterior [Brehmer et al. 2019] para comparar os resultados, e mais duas tarefas adicionais para avaliar o uso da terceira dimensão, resultando em 11 tarefas formais de análise de tendências.

Os participantes eram posicionados no centro de uma sala virtual (altura = 3m, largura = 4m, profundidade = 4m). Para cada tarefa, uma única visualização foi apresentada ao usuário no ambiente virtual, que também contém uma TV que exibe instruções. Para a última tarefa, uma *cena mista* que inclui as três visualizações foi apresentada, e os usuários podiam escolher livremente qual delas preferiam usar para realizar a tarefa. Nesse caso, a técnica de Animação é exibida no lado esquerdo, *Small Multiples* na parte frontal e *Overlaid Trails* no lado direito (Figura 5.9). O raio e a mão virtual são usados como modos de interação de longa distância e curta distância, respectivamente (detalhes nos círculos em ambos os lados da figura). As informações sobre um item de dados são exibidas quando o usuário alcança o ponto correspondente em qualquer modo.

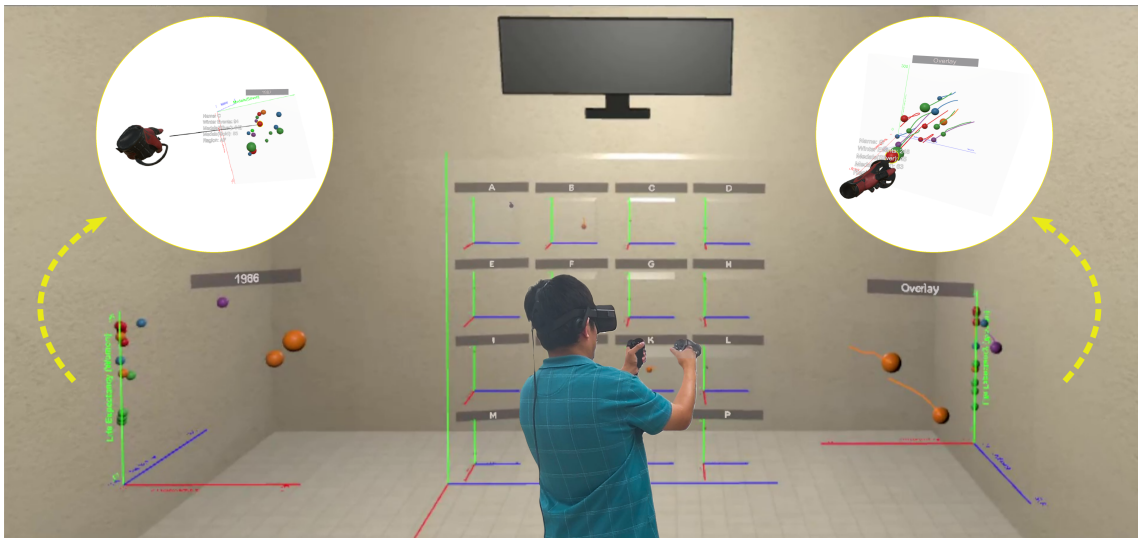


Figura 5.9. Tarefas de análise de tendências usando a abordagem *Spaces* com três variantes interativas de gráficos de dispersão 3D em ambiente imersivo.

Os resultados mostraram que *Overlaid Trails* apresentam o melhor desempenho geral. No entanto, a precisão depende da tarefa e quando a tarefa requer análise de tendências usando as três dimensões, a precisão é inferior. Nossos resultados também mostraram o valor da interação devido aos *insights* proporcionados pela interação nas decisões dos usuários. Maiores detalhes desse segundo estudo podem ser obtidos em [Quijano-Chavez et al. 2023].

5.4.3. Desafios da Exploração de Dados em Ambientes Imersivos

A exploração e análise de dados em ambientes imersivos apresenta uma série de desafios para que sistemas de IA alcançarem seu pleno potencial no que diz respeito à visualização situada, interação, análise colaborativa e avaliação [Ens et al. 2021]. A tabela 5.1

apresenta esses desafios. Da mesma forma, outros autores [Kraus et al. 2021] também refletiram sobre quando e como a imersão pode ser apropriada para análise de dados, apresentando cenários similares aos de [Ens et al. 2021].

Tabela 5.1. Desafios para a análise de dados em ambientes imersivos. Fonte: [Ens et al. 2021]

.

Tópicos		Desafios
VISUALIZAÇÃO DE DADOS ESPACIALMENTE SITUADA	C1	Posicionando Visualizações com Precisão no Espaço
	C2	Extraindo e Representando Conhecimento Semântico
	C3	Definindo Diretrizes para Visualização Espacialmente Situada
	C4	Compreendendo os Sentidos Humanos e a Cognição em Contextos Situados
	C5	Aplicando Visualização Espacial de Forma Ética
INTERAÇÃO COM SISTEMAS DE ANÁLISE IMERSIVA	C6	Explorando os Sentidos Humanos para Análise Imersiva Interativa
	C7	Permitindo Feedback Multissensorial para Análise Imersiva
	C8	Apoiando Transições em Ambientes Imersivos
	C9	Lidando com a Complexidade da Interação na Análise Imersiva
ANÁLISE COLABORATIVA	C10	Apoiando o Comportamento com Colaboradores
	C11	Superando Restrições da Realidade
	C12	Apoiando a Colaboração entre Plataformas
	C13	Integrando a Prática de Colaboração Atual
	C14	Avaliando o Trabalho Colaborativo
CENÁRIOS DE USO E AVALIAÇÃO	C15	Definindo Cenários de Aplicação para Análise Imersiva
	C16	Compreendendo Usuários e Contextos para Avaliação da Análise Imersiva
	C17	Estabelecendo um Framework de Avaliação para Análise Imersiva

Nos dois estudos que apresentamos, num esforço para abordar alguns dos desafios relatados na Tabela 5.1, desenvolvemos e avaliamos a abordagem *Spaces* para interagir com múltiplas visualizações coordenadas que exibem visualizações 3D em ambientes imersivos. Primeiro, exploramos múltiplas visualizações tridimensionais coordenadas, avaliando o desempenho durante tarefas compostas, usabilidade, técnicas de interação e modos de interação [Quijano-Chavez et al. 2021]). Durante essa fase, projetamos a ideia principal de nossa abordagem, na qual o usuário pode agarrar, mover e clonar contêineres de Espaços com visualizações dentro deles, permitindo padrões compostos. Em segundo lugar, aplicamos o conhecimento obtido na primeira fase para aprimorar nossa abordagem e avaliar a eficácia de três variantes de gráfico de dispersão 3D (*Animation*, *Overlaid Trails*

e *Samll Multiples* para análise de tendências em ambientes imersivos [Quijano-Chavez et al. 2023].

O desenvolvimento da abordagem *Spaces* exigiu que abordássemos vários aspectos, com base em estudos anteriores:

1. Desenvolver técnicas para usar múltiplas visualizações em RV é um desafio, pois, segundo Knudsen e Carpendale [Knudsen and Carpendale 2017], elas requerem um controle mais complexo das técnicas de interação .
2. Há uma necessidade de métodos de interação capazes de alcançar as funcionalidades WIMP (janelas, ícones, menus, ponteiro) usadas de forma predominante para tarefas de análise visual [Lee et al. 2012].
3. Alguns experimentos realizados com o FiberClay [Hurter et al. 2019] para explorar trajetórias permitiram que os autores relatassem sugestões para melhorar a experiência do usuário em ambientes de RV com múltiplas visualizações, como: evitar componentes de interface gráfica 2D, limitar o número de modos de interação, facilitar a navegação e o uso preferencial de uma visualização principal.
4. Outros estudos, como o de Yang et al. [Yang et al. 2021], sugeriram a implementação de diversos métodos de navegação para se adequarem a diferentes tamanhos de salas, permitindo uma experimentação suave remotamente.
5. Finalmente, Wagner et al. [Wagner et al. 2021] mostraram que a integração de diferentes modos de interação (longa e curta distância) não é apenas útil, mas necessária para a IA superar as limitações de métodos de entrada específicos.

Finalmente, podemos afirmar que ambos os estudos confirmaram que a abordagem *Spaces* apresentou bons resultados em relação a (1) conforto e interação do usuário em comparação com a versão *desktop* correspondente e (2) utilidade para tarefas comparativas usando técnicas de visualização tridimensional em um ambiente de RV.

5.4.4. VirtualDesk: Alternativa de Interface para Exploração de Dados em Ambiente Imersivo

Os estudos anteriores com usuários, tanto os relatados na literatura, como os descritos na seção anterior deste capítulo, sugerem que abordagens imersivas podem efetivamente auxiliar na exploração de dados, mas que novas avaliações e diretrizes ainda são necessárias.

Muitas abordagens de navegação propostas, por exemplo, são impraticáveis para uso real. Metáforas de voo, em particular, são demoradas e frequentemente resultam em enjoos durante as simulações. Outras abordagens, como caminhar no ambiente real, também são desnecessariamente ineficientes, tanto em termos de tempo quanto de espaço necessário. Além disso, outro problema conhecido é como exibir conteúdo e textos inerentemente 2D em ambientes virtuais, assim como menus de comandos.

Com o objetivo de contornar esses problemas, propusemos e implementamos uma abordagem alternativa de exploração de dados, projetada para ser mais adequada para uso real. A metáfora, chamada *VirtualDesk*, combina características de diferentes origens:

1. A representação visual do conjunto de dados é exibida em escala menor ao alcance do braço, para aproveitar melhor a propriocepção, interação mais precisa em relação ao corpo e maior estereopsia e paralaxe de movimento da cabeça [Mine et al. 1997].

2. A manipulação natural de dados incorporada está em conformidade com o conceito recente de coordenação espaço-dados, ou seja, uma correspondência de um para um entre ações físicas e virtuais, visando reduzir a carga cognitiva do usuário [Cordeil et al. 2017].

3. Uma mesa virtual é representada, sincronizada com a mesa real do usuário. Isso permite a interação tangível com controles e visualizações coordenadas em 2D, colocados na superfície da mesa [Zielasko et al. 2017].

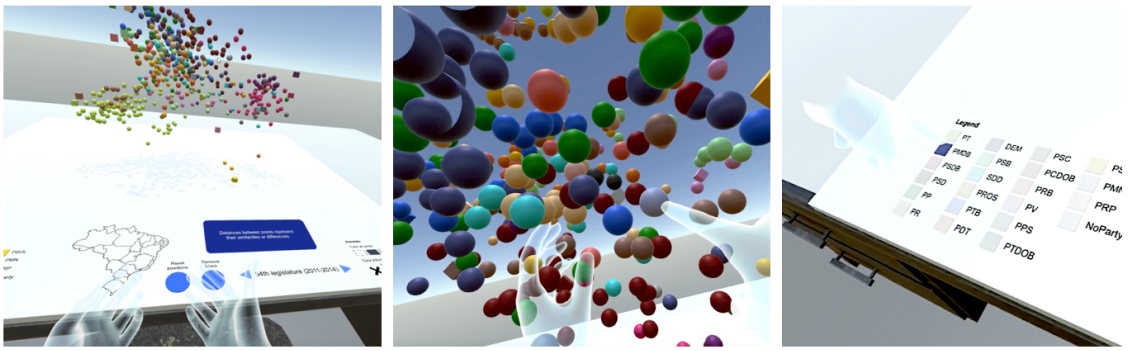


Figura 5.10. Na metáfora da VirtualDesk, os dados são exibidos ao alcance do braço e manipulados apenas por gestos naturais diretos no ar.

Realizamos dois estudos para avaliar como o protótipo *VirtualDesk* se sairia em comparação com as abordagens convencionalmente usadas em desktops e a abordagem imersiva utilizada com navegação por voo.

Escolhemos como caso de uso a visualização dos dados obtidos das votações nominais da Câmara dos Deputados do Brasil. Este conjunto de dados é particularmente interessante porque a visualização resultante é um gráfico de dispersão com os 513 deputados de diferentes partidos políticos, com fronteiras ideológicas muito difusas. Também consideramos este domínio muito apropriado para nossos objetivos devido à alta dimensionalidade de seus conjuntos de dados (cada votação nominal é uma dimensão) e à fácil definição de tarefas analíticas semanticamente significativas. Na Figura 5.10, cada esfera é um deputado e a proximidade entre eles denota a similaridade de seus votos ao longo de um período.

A título de *baseline*, sem a abordagem *VirtualDesk*, um primeiro estudo foi conduzido com 30 participantes recrutados no campus para comparar o desempenho de tarefas de exploração desse conjunto de dados nas condições de visualização baseada em desktop (2D e 3D) e numa condição de visualização 3D baseada em HMD. Neste estudo, uma abordagem convencional de navegação em voo direcionada pelo olhar foi implementada. Essa metáfora foi projetada para ser simples de aprender e permitir uma visão egocêntrica, colocando o usuário dentro dos dados.

Surpreendentemente, no entanto, não foram observadas diferenças perceptuais, e erros igualmente baixos em todas as condições resultaram em melhorias com a adição da terceira dimensão com ou sem imersão quando o conjunto de dados permitia. Mesmo assim, ao examinar os resultados subjetivos, descobriu-se que a condição baseada em HMD exigiu menos esforço para encontrar informações e menos navegação, além de oferecer uma percepção subjetiva muito maior de precisão e envolvimento. Suas principais limitações, por outro lado, foram a alta incidência de enjoo de simulador, com cerca de 40% dos participantes relatando níveis significativos de desconforto, e tempos de conclusão da tarefa prolongados. Maiores detalhes sobre este estudo podem ser encontrados em [Wagner Filho et al. 2018b].

Num segundo estudo, para avaliar a abordagem *VirtualDesk*, desenvolvemos uma versão 3D desktop similar a um ambiente imersivo, com as mesmas funcionalidades, mas seguindo abordagens típicas de interação com mouse e teclado.

Empregando o mesmo caso de uso de exploração dos dados multidimensionais de votações projetados em três dimensões, recrutamos 24 participantes que realizaram um conjunto estendido de tarefas de percepção e interação, inspiradas na literatura e no estudo anterior.

Os resultados mostraram que o *VirtualDesk* se saiu igualmente bem ou melhor em termos de taxas de erro em todas as tarefas analíticas, tanto em comparação com uma interface de *desktop* quanto com a implementação imersiva anterior com navegação por voo. O tempo adicional em relação ao *desktop* foi significativo apenas em tarefas com maiores requisitos para interação na mesa (que exigiram mudança de ponto de vista e também impuseram certas dificuldades para alguns usuários) e foi geralmente de apenas alguns segundos. Isso ocorreu apesar do fato de que a exploração de dados em termos de rotação do conjunto de dados foi 5,8 vezes maior. Considerando que a observação de diferentes pontos de vista é fundamental para a compreensão de uma nuvem de pontos 3D, isso explica parcialmente a vantagem da abordagem *VirtualDesk* em tarefas de percepção.

O ambiente imersivo também contribuiu para percepções subjetivas mais elevadas de eficiência e engajamento, enquanto incorreu em um tempo adicional mínimo e gerou quase nenhum sintoma de enjoo. Apesar do tempo de exposição à RV muito semelhante em ambos os estudos, a pontuação média de enjoo no VirtualDesk foi 7 vezes menor do que na versão com navegação artificial. Além disso, enquanto nesse estudo 40% dos usuários experimentaram níveis de desconforto muito significativos, ou seja, com pontuações superiores a 20, agora a pontuação individual máxima foi 18,3 e 83% dos usuários perceberam apenas sintomas negligenciáveis ou mínimos. Este estudo de caso é descrito em detalhes em [Wagner Filho et al. 2018a].

5.5. Comentários Finais

As experiências aqui relatadas contemplam uma pequena parcela dos desafios que devem ser vencidos para tornar soluções utilizando realidade virtual úteis e viáveis num conjunto mais amplo de aplicações. Aos desafios listados na Tabela 5.1, acrescentamos desafios relacionados a aspectos tecnológicos e outros inerentes à adoção e uso frequente ou contínuo de realidade virtual ou aumentada.

Do ponto de vista tecnológico, apesar da evolução dos HMDs e dispositivos de interação que notamos ao desenvolver e experimentar técnicas ao longo dos anos com diferentes dispositivos, há espaço para melhorias (i)no conforto desses dispositivos, (ii) na precisão de rastreamento *indoor* e *outdoor* assim como (iii) no rastreamento e reconhecimento de gestos sem marcadores. A constante evolução que se espera dos dispositivos traz um desafio de evolução para as aplicações pois as questões de compatibilidade com novos dispositivos e sistemas tendem a ser mais complexas do que as que surgem em aplicações desktop. A interoperabilidade entre ambientes é desafiadora. Por exemplo, como compartilhar dados entre diferentes ambientes imersivos e entre imersivo e desktop, para garantir que aplicações criadas para um ambiente possam ser utilizadas em outro. Isso promove a acessibilidade e uma base de usuários mais ampla.

Do ponto de vista de adoção, estima-se que nem todas as aplicações são de natureza tal que possam se beneficiar de ambientes de realidade virtual. Descobrir quando, onde e porque utilizar realidade virtual deverá estar no horizonte dos projetistas. Nós, humanos, nos beneficiamos do fato de estarmos imersos num espaço tridimensional, no qual utilizamos nossos sentidos em todas as suas potencialidades. Por isso, aplicações que possam se beneficiar das vantagens oferecidas pelas técnicas de realidade virtual e interação 3D ao utilizarem outros sentidos além da visão, ou que incorporem as características de sucesso dos jogos, poderão perdurar além do fator "novidade"que parece mover a oferta de muitos produtos. Por outro lado, as questões de acessibilidade que tem boas soluções em aplicações desktop constituem um outro desafio a ser vencido em ambientes imersivos.

Há outros aspectos tão ou mais importantes a serem investigados e avaliados no que diz respeito ao uso da realidade virtual. Enquanto os riscos de uso continuado, excessivo ou incorreto de outros dispositivos e aplicações têm sido estudados ao longo dos anos e evidências têm sido colhidas, pouco sabemos sobre os riscos e consequências de uso de RV, além dos medidos em experimentos pontuais através de questionários padronizados. Há outras questões críticas a serem exploradas como a segurança dos usuários e a privacidade das informações nesses ambientes.

A realidade virtual até pouco tempo atrás restrita aos laboratórios de pesquisa e a algumas aplicações de treinamento em tarefas complexas e de risco, como na indústria e na saúde, está se tornando disponível a uma gama mais ampla de usuários em áreas como entretenimento, comércio, arquitetura e educação em geral, onde o engajamento é um dos elementos de sucesso. Por isso, as oportunidades de pesquisa são inúmeras quando consideramos os desafios mencionados anteriormente. São particularmente interessantes as oportunidades proporcionadas pelo desenvolvimento de aplicações onde o aprendizado envolve consciência corporal ou obtenção de habilidades em procedimentos espaciais, e as aplicações de análise de dados explorando a transposição para 3D da diversidade de técnicas de visualização existentes.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Os projetos de pesquisa relatados neste capítulo têm o financiamento das agências CNPq, CAPES

(Código de Financiamento 001) e FAPERGS e da RNP.

Referências

[Par 2020] (2020). *Manual para o OSCE*. Sanar, Salvador.

[Adolph and Kretch 2015] Adolph, K. E. and Kretch, K. S. (2015). Gibson's theory of perceptual learning. In Wright, J. D., editor, *International Encyclopedia of the Social & Behavioral Sciences (Second Edition)*, pages 127–134. Elsevier, Oxford, second edition edition.

[Asnar and Zannone 2008] Asnar, Y. and Zannone, N. (2008). Perceived risk assessment. In *Proceedings of the 4th ACM Workshop on Quality of Protection*, pages 59–64. ACM.

[Bowman et al. 2009] Bowman, D. A. et al. (2009). Higher levels of immersion improve procedure memorization performance. In *Proceeding JVRC'09 Proceedings of the 15th Joint Virtual Reality Eurographics Conference on Virtual Environments*.

[Brehmer et al. 2019] Brehmer, M., Lee, B., Isenberg, P., and Choe, E. K. (2019). A comparative evaluation of animation and small multiples for trend visualization on mobile phones. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):364–374.

[Brehmer and Munzner 2013] Brehmer, M. and Munzner, T. (2013). A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385.

[Bryson and Levit 1992] Bryson, S. and Levit, C. (1992). The virtual wind tunnel. *IEEE Computer Graphics and Applications*, 12(4):25–34.

[Chalmers and Debattista 2009] Chalmers, A. and Debattista, K. (2009). Level of realism for serious games. In *Games and Virtual Worlds for Serious Applications, 2009. VS-GAMES '09. Conference in*.

[Cordeil et al. 2017] Cordeil, M., Bach, B., Li, Y., Wilson, E., and Dwyer, T. (2017). A design space for spatio-data coordination: Tangible interaction devices for immersive information visualisation. In *Proceedings of IEEE Pacific Visualization Symposium (Pacific Vis)*.

[Deterding et al. 2011] Deterding, S. et al. (2011). From game design elements to gamefulness: Defining gamification. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*.

[Ens et al. 2021] Ens, B., Bach, B., Cordeil, M., Engelke, U., Serrano, M., Willett, W., Prouzeau, A., Anthes, C., Büschel, W., Dunne, C., Dwyer, T., Grubert, J., Haga, J. H., Kirshenbaum, N., Kobayashi, D., Lin, T., Olaosebikan, M., Pointecker, F., Saffo, D., Saquib, N., Schmalstieg, D., Szafir, D. A., Whitlock, M., and Yang, Y. (2021). *Grand Challenges in Immersive Analytics*. Association for Computing Machinery, New York, NY, USA.

[Hurter et al. 2019] Hurter, C., Riche, N. H., Drucker, S. M., Cordeil, M., Alligier, R., and Vuillemot, R. (2019). Fiberclay: Sculpting three dimensional trajectories to reveal structural insights. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):704–714.

[Ibanez et al. 2011] Ibanez, B., Marne, B., and Labat, J. (2011). Conceptual and technical frameworks for serious games. In *Proceedings of the 5th European Conference on Games Based Learning*, pages 81–87.

[Jorge et al. 2013] Jorge, V. A. M., Sarmiento, W. J., Maciel, A., Nedel, L., Collazos, C. A., Faria, F., and Oliveira, J. (2013). Interacting with danger in an immersive environment: Issues on cognitive load and risk perception. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*, pages 83–92, New York, NY, USA. ACM.

[Knudsen and Carpendale 2017] Knudsen, S. and Carpendale, S. (2017). Multiple views in immersive analytics. In *IEEE VIS 2017 Workshop on Immersive Analytics*.

[Krathwohl 2002] Krathwohl, D. R. (2002). A revision of bloom's taxonomy: An overview. *Theory Into Practice*, 41(4):212–218.

[Kraus et al. 2021] Kraus, M., Klein, K., Fuchs, J., Keim, D. A., Schreiber, F., and Sedlmair, M. (2021). The value of immersive visualization. *IEEE Computer Graphics and Applications*, 41(4):125–132.

[Lee et al. 2012] Lee, B., Isenberg, P., Riche, N. H., and Carpendale, S. (2012). Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2689–2698.

[Marriott et al. 2018a] Marriott, K., Chen, J., Hlawatsch, M., Itoh, T., Nacenta, M. A., Reina, G., and Stuerzlinger, W. (2018a). Just 5 questions: Toward a design framework for immersive analytics. In Marriott, K., Schreiber, F., Dwyer, T., Klein, K., Riche, N. H., Itoh, T., Stuerzlinger, W., and Thomas, B. H., editors, *Immersive Analytics*, volume 11190, chapter 9, pages 119–133. Springer.

[Marriott et al. 2018b] Marriott, K., Schreiber, F., Dwyer, T., Klein, K., Riche, N. H., Itoh, T., Stuerzlinger, W., and Thomas, B. H. (2018b). *Immersive Analytics*, volume 11190. Springer, Cham.

[McCormick et al. 1987] McCormick, B. H., DeFanti, T. A., and Brown, M. (1987). Visualization in scientific computing. *ACM Computing Graphics*, 21(6).

[Menin et al. 2021] Menin, A., Torchelsen, R., and Nedel, L. (2021). The effects of VR in training simulators: Exploring perception and knowledge gain. *Computers and Graphics*, 102:402–412.

[Michael and Chen 2005] Michael, D. and Chen, S. (2005). *Serious Games: Games that Educate, Train, and Inform*. Muska & Lipman/Premier-Trade.

[Milgram and Kishino 1994] Milgram, P. and Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, E77-D(12):1321–1329.

[Mine et al. 1997] Mine, M. R., Brooks Jr, F. P., and Sequin, C. H. (1997). Moving objects in space: exploiting proprioception in virtual-environment interaction. In *SIGGRAPH*, pages 19–26. ACM Press/Addison-Wesley Publishing Co.

[Munzner 2014] Munzner, T. (2014). *Visualization Analysis and Design*. CRC Press.

[Nedel et al. 2016] Nedel, L., de Souza, V. C., Menin, A., Sebben, L., Oliveira, J., Faria, F., and Maciel, A. (2016). Using immersive virtual reality to reduce work accidents in developing countries. *IEEE Computer Graphics and Applications*, 36(2):36–46.

[Negrão et al. 2023] Negrão, M., Ferreira, W., Bohrer, B., Freitas, C., Maciel, A., and Nedel, L. (2023). Design and think-aloud study of an immersive interface for training health professionals in clinical skills. In *Proceedings of the Symposium on Virtual and Augmented Reality (SVR)*, Rio Grande, RS, Brazil.

[Okuda et al. 2009] Okuda, Y., Bryson, E. O., DeMaria Jr, S., Jacobson, L., Quinones, J., Shen, B., and Levine, A. I. (2009). The utility of simulation in medical education: what is the evidence? *Mount Sinai Journal of Medicine: A Journal of Translational and Personalized Medicine: A Journal of Translational and Personalized Medicine*, 76(4):330–343.

[Quijano-Chavez et al. 2021] Quijano-Chavez, C., Nedel, L., and Freitas, C. M. (2021). An immersive approach based on two levels of interaction for exploring multiple coordinated 3d views. In *Human-Computer Interaction – INTERACT 2021*, pages 493–513, Cham. Springer International Publishing.

[Quijano-Chavez et al. 2023] Quijano-Chavez, C., Nedel, L., and Freitas, C. M. D. S. (2023). Comparing scatterplot variants for temporal trends visualization in immersive virtual environments. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 669–679.

[Ragan et al. 2015] Ragan, E. D., Bowman, D. A., Kopper, R., Stinson, C., Scerbo, S., and McMahan, R. P. (2015). Effects of field of view and visual complexity on virtual reality training effectiveness for a visual scanning task. *IEEE Transactions on Visualization and Computer Graphics*, 21(7):794–807.

[Sawyer 2007] Sawyer, B. (2007). The "serious games"landscape. In *Instructional & Research Technology Symposium for Arts, Humanities, and Social Sciences*.

[Sawyer et al. 2015] Sawyer, T., White, M., Zaveri, P., Chang, T., Ades, A., French, H., Anderson, J., Auerbach, M., Johnston, L., and Kessler, D. (2015). Learn, See, Practice, Prove, Do, Maintain: An Evidence-Based Pedagogical Framework for Procedural Skill Training in Medicine. *Academic Medicine*, 90(8):1025–1033.

[Souza et al. 2021] Souza, V., Maciel, A., Nedel, L., and Kopper, R. (2021). Measuring presence in virtual environments: A survey. *ACM Comput. Surv.*, 54(8).

[Sutherland 1965] Sutherland, I. E. (1965). The ultimate display. In *Proceedings of the Congress of the International Federation of Information Processing (IFIP)*, volume volume 2, pages 506–508.

[Thomas and Cook 2005] Thomas, J. and Cook, K. (2005). *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Centre, Richland, WA.

[Wagner et al. 2021] Wagner, J., Stuerzlinger, W., and Nedel, L. (2021). Comparing and combining virtual hand and virtual ray pointer interactions for data manipulation in immersive analytics. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2513–2523.

[Wagner Filho et al. 2018a] Wagner Filho, J. A., Freitas, C., and Nedel, L. (2018a). Virtualdesk: A comfortable and efficient immersive information visualization approach. *Computer Graphics Forum*, 37(3):415–426.

[Wagner Filho et al. 2018b] Wagner Filho, J. A., Rey, M. F., Freitas, C. M. D. S., and Nedel, L. (2018b). Immersive visualization of abstract information: An evaluation on dimensionally-reduced data scatterplots. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 483–490.

[Wikipedia 2023] Wikipedia (2023). Metaverso. Accessed: September 2023.

[Yang et al. 2021] Yang, Y., Cordeil, M., Beyer, J., Dwyer, T., Marriott, K., and Pfister, H. (2021). Embodied navigation in immersive abstract data visualization: Is overview+detail or zooming better for 3d scatterplots? *IEEE Transactions on Visualization and Computer Graphics*, 27(02):1214–1224.

[Zielasko et al. 2017] Zielasko, D., Weyers, B., Bellgardt, M., Pick, S., Meibner, A., Vierjahn, T., and Kuhlen, T. W. (2017). Remain seated: towards fully-immersive desktop VR. In *2017 IEEE 3rd Workshop on Everyday Virtual Reality (WEVR)*, pages 1–6. IEEE.

[Zyda 2005] Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer*, 38(9):25–32.

Capítulo 6

Computação Visual e Detecção Precoce de Doenças em Escala Global: Oportunidades e Desafios

Manuel M. Oliveira[1], Giovani A. Meneguel[1], Maikel M. Rönnau[1], Pantelis V. Rados[2]

[1] Programa de Pós-Graduação em Computação (PPGC) - UFRGS
[2] Programa de Pós-Graduação em Odontologia (PPGODO) - UFRGS

Abstract

Combining recent advances in machine learning with the Internet infrastructure and the computing capabilities of smartphones allows us to develop computational solutions with the potential to positively impact people's lives on a global scale. Such potential is particularly promising for the early detection of diseases, such as cancer. This work presents a software architecture for addressing these and other critical challenges in public health. It discusses how to develop solutions that benefit the population, leading to feedback to improve the quality of the provided services, thus creating a virtuous cycle.

Resumo

Avanços recentes em técnicas de aprendizagem de máquina combinados à infraestrutura da Internet e aos recursos disponíveis em smartphones nos permitem desenvolver soluções computacionais capazes de impactar positivamente a qualidade de vida das pessoas em escala planetária. Tal potencial é particularmente importante para a detecção precoce de doenças como, por exemplo, o câncer. Este trabalho apresenta uma arquitetura de software para o desenvolvimento de sistemas que visam enfrentar estes e outros importantes desafios em saúde pública. Ele discute como desenvolver soluções que beneficiem a população, que por sua vez provê realimentação para o aprimoramento dos serviços disponibilizados, criando um ciclo virtuoso.

Vídeo com a apresentação do capítulo: `https://youtu.be/1l4VMarSOsg`

6.1. Introdução

O uso de tecnologias da informação e comunicação (TICs) tem permitido ampliar o acesso da população a serviços de saúde, bem como racionalizar os custos associados à disponibilização desses serviços. Os termos *telemedicina* e *telessaúde* são utilizados para se referir a estas iniciativas que, segundo a Organização Mundial da Saúde (OMS), consistem na "*disponibilização de serviços de saúde, onde a distância é um fator crítico, e nos quais profissionais de saúde utilizam tecnologias da informação e comunicação para troca de informações visando o diagnóstico, tratamento e prevenção de doenças, pesquisa e avaliação, bem como para o processo de educação continuada de profissionais de saúde, objetivando a melhoria da saúde de indivíduos e de suas comunidades*" [WHO 1998]. Embora o termo telemedicina seja por vezes utilizado para referir-se especificamente a serviços prestados por médicos enquanto o termo telessaúde é utilizado para designar serviços prestados por profissionais de saúde em geral, neste trabalho os dois termos são usados como sinônimos.

Os primeiros registros de atividades de telemedicina remontam ao início do século 20 quando dados de eletrocardiograma foram transmitidos através de linhas telefônicas [Le 1906]. Atualmente, serviços como teleradiologia, teledermatologia, telepatologia, telepsiquiatria e mesmo telecirurgias estão se tornando comuns em países desenvolvidos [WHO 2010], e a recente pandemia do coronavírus (SARS-COV-2) contribuiu para uma rápida adoção de serviços de teleconsulta em diversos países. O Brasil, caracterizado por sua vastidão geográfica, por regiões de difícil acesso, e por uma distribuição irregular de serviços médicos especializados e de qualidade, possui um grande potencial para ampliação do acesso a serviços de saúde à sua população via telemedicina. Visando explorar este potencial, o Ministério da Saúde em articulação com as universidades públicas implementou núcleos técnico-científicos e criou o Programa Nacional Telessaúde Brasil Redes (PNTBR) [Maldonado et al. 2016].

Entretanto, a implementação de um amplo e efetivo programa de telessaúde requer a solução de diversos **desafios**, entre os quais:

1. Como garantir a disponibilização de serviços com ampla cobertura geográfica;
2. Como garantir a qualidade destes serviços e o seu constante aprimoramento;
3. Como garantir a escalabilidade dos serviços para um grande número de usuários;
4. Como prover serviços em diversas especialidades e para múltiplas doenças;
5. Como garantir a disponibilidade dos serviços 24 horas por dia, 7 dias por semana;
6. Como auxiliar a detecção precoce de doenças, visando a melhoria da qualidade de vida dos pacientes e a redução dos custos dos tratamentos;
7. Como implementar os requisitos acima com baixo custo.

Sem esquecer que os profissionais de saúde são os principais atores de um serviço de telessaúde, propomos uma estratégia para solução dos desafios técnicos listados acima. Tal

estratégia permitirá a disponibilização de sistemas de telessaúde mais efetivos, podendo ser diretamente incorporada ao PNTBR.

A busca por sistemas de saúde mais inclusivos, de qualidade, e com efetividade de custo guiou o desenvolvimento de sistemas de telessaúde na última década. Neste contexto, a Internet e os smartphones tiveram um papel fundamental. Recentemente, observamos o desenvolvimento acelerado de mais uma tecnologia disruptiva caracterizada por técnicas de aprendizagem de máquina, notadamente ligadas a áreas como visão computacional e modelos de linguagens. No caso específico de visão computacional, modelos baseados em redes neurais convolucionais (CNNs) têm demonstrado bons resultados na predição de diversas doenças como câncer de pele [Dildar et al. 2021], de mama [Mambou et al. 2018] e de pulmão [Bhatia et al. 2019], Covid-19 [Ozturk et al. 2020, Ismael and Şengür 2021], retinopatia diabética [Gargeya and Leng 2017], glaucoma [Abbas 2017], e Alzheimer [Ebrahimighahnavieh et al. 2020], para citar apenas alguns exemplos. O surgimento de técnicas automáticas com acurácia semelhante a de especialistas tem o potencial de transformar a telemedicina, ampliando significativamente a disponibi-lidade e o alcance dos serviços oferecidos. Tais avanços em técnicas de aprendizagem de máquina combinados à infraestrutura da Internet e aos recursos disponíveis em smartphones nos oferecem a **oportunidade** *de desenvolver soluções computacionais escaláveis para telessaúde capazes de impactar positivamente a qualidade de vida das pessoas em escala planetária*. Este potencial é particularmente importante na identificação precoce de doenças.

Este trabalho apresenta uma arquitetura de software para suporte à detecção precoce de doenças partir de imagens (Seção 6.2). Ele discute o desenvolvimento de soluções escaláveis, de baixo custo e com potencial de alcance mundial que beneficiem a população, que por sua vez provê realimentação para o aprimoramento dos serviços, criando um ciclo virtuoso. Para tal, a arquitetura proposta disponibiliza serviços automatizados de telessaúde (*e.g.*, predição de doenças) utilizando infraestrutura existente: a Internet, que fornece o meio de comunicação entre os provedores de solução e seus usuários; smartphones, que provêm os recursos computacionais para os usuários; e lojas de aplicativos (*e.g., Google Play* e *App Store*), que oferecem canais de distribuição de apps (para acesso aos serviços) com capilaridade global. O uso desta infraestrutura permite que estes sistemas sejam disponibilizados com um baixo investimento financeiro. Para ilustrar o uso da arquitetura proposta, apresentaremos um sistema para suporte à detecção precoce de câncer de boca que se encontra em desenvolvimento no Programa de Pós-Graduação em Computação (PPGC) da UFRGS em parceria com a Faculdade de Odontologia (Seção 6.3).

6.2. Uma Arquitetura para Detecção Precoce de Doenças a partir de Imagens

A Figura 6.1 ilustra a arquitetura proposta, na qual profissionais de saúde podem submeter imagens de exames (*e.g.*, radiografias, retinografias, etc.) ou fotos de lesões suspeitas em seus pacientes (*e.g.*, manchas de pele, lesões na língua ou na mucosa bucal) para avaliação por um modelo de aprendizagem de máquina especializado no tipo de doença considerada (desafios 1, e 3 a 5). Tal modelo, treinado a partir de uma base de dados de imagens fornecidas e anotadas por especialistas, analisa cada imagem submetida e retorna sua predição sobre a probabilidade de ocorrência da doença específica, indicando as áreas suspeitas na imagem. Quando apropriado, o sistema recomenda ao profissional de saúde a realização

de biópsia para confirmação de casos suspeitos (*e.g.*, suspeita de câncer). No caso de realização de biópsia, espera-se que o profissional de saúde submeta o laudo ao sistema. De posse do laudo (positivo ou negativo) e do conjunto de imagens associadas fornecidas pelo profissional de saúde, um profissional responsável pela curadoria dos dados acrescentará as imagens correspondentes com as respectivas anotações ao banco de imagens. O banco de imagens então atualizado será utilizado para gerar (treinar) uma nova versão do modelo de predição. A disponibilização de novas imagens com seus respectivos laudos gera um ciclo virtuoso que deve levar a uma melhoria da acurácia do modelo de predição (desafio 2).

O sistema pode ser hospedado em um servidor institucional ou em algum serviço em nuvem. Completam este ecossistema os desenvolvedores de aplicativos (apps) para dispositivos móveis. Os apps são disponibilizados através das lojas de aplicativos, tornando-os acessíveis em todo o mundo (desafio 1). A interação do profissional de saúde com o modelo de predição é realizada por meio destes aplicativos ou por meio de uma aplicação web acessível via computadores pessoais (Figura 6.1). Utilizando estes mecanismos, os profissionais de saúde podem enviar imagens para predição, enviar laudos de biópsias, e acessar os resultados das predições para seus pacientes. As imagens enviadas são submetidas ao modelo de predição, sendo armazenadas juntamente com os resultados das avaliações nos registros dos respectivos pacientes no banco de dados.

A arquitetura mostrada na Figura 6.1 pode ser especializada para predição de diferentes tipos de doenças, diferindo apenas com relação ao modelo de predição e ao conjunto de imagens e respectivas anotações utilizadas para o treinamento do modelo (desafio 4). Assim, por exemplo, o modelo de inferência pode ser instanciado para detecção de doenças como câncer de pele [Dildar et al. 2021], câncer de mama [Mambou et al. 2018], câncer de pulmão [Bhatia et al. 2019], câncer de boca, para detecção de Covid-19 a partir de raio-X [Ozturk et al. 2020, Ismael and Şengür 2021], retinopatia diabética [Gargeya and Leng 2017], glaucoma a partir de imagens de fundo de olho [Abbas 2017], e Alzheimer a partir de neuroimagens [Ebrahimighahnavieh et al. 2020]. Por ser um serviço automatizado, disponível 24 horas por dia, apresentar um custo relativamente baixo, e ser aplicável a vários tipos de doenças, acreditamos que esta estratégia terá grande importância na democratização de serviços de saúde nos próximos anos. Note que um único servidor pode hospedar vários modelos, cada um treinado para detecção (precoce) de um tipo específico de doença. Isto contribui para uma redução ainda maior dos custos de disponibilização destes serviços (desafios 6 e 7).

Os smartphones constituem uma interface bastante conveniente de interação com esta arquitetura, permitindo, por exemplo, a captura e envio de imagens para avaliação pelo sistema, bem como o acompanhamento dos resultados. A Seção 6.3 descreve um sistema para detecção de câncer de boca baseado na arquitetura apresentada.

6.3. Sistema para Detecção Precoce de Câncer de Boca

O câncer é a segunda maior causa de mortes em todo o mundo [Roser and Ritchie 2015] e estima-se que em 2020 a doença tenha vitimado 10 milhões de pessoas [Sung et al. 2021]. O câncer de boca, por sua vez, é o tipo mais prevalente na região da cabeça e pescoço, com estimativas de 657.000 novos casos e 300.000 mortes anualmente, sendo

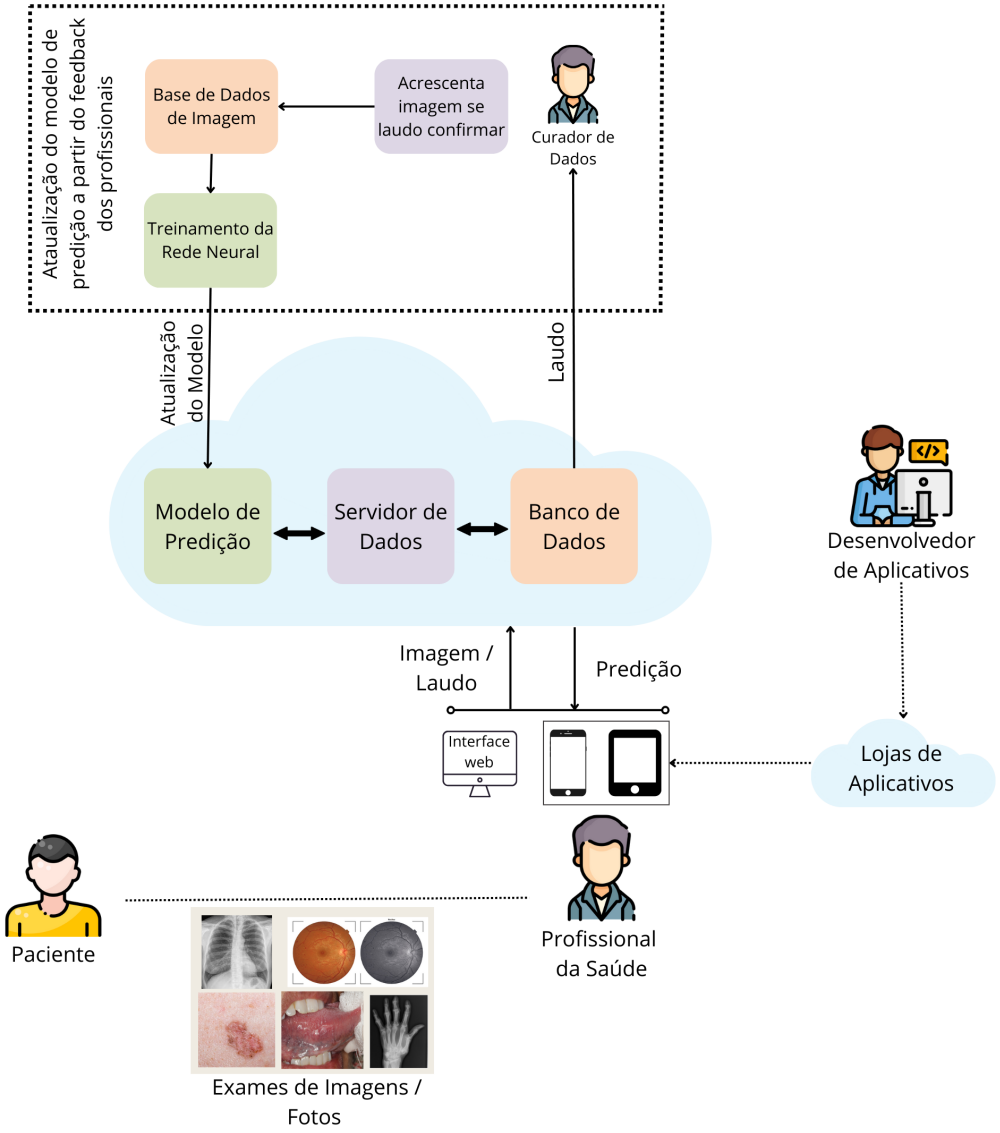


Figura 6.1: Arquitetura geral de sistema proposto para detecção precoce de doenças a partir de imagens.

que por mais de uma década o número de diagnósticos vem subindo [Foundation 2023]. Atualmente, a forma mais efetiva de reduzir o número de vítimas fatais de todas as formas de câncer é a detecção precoce.

O diagnóstico do câncer de boca ainda depende da análise clínica e de realização de biopsia. Geralmente, isso ocorre apenas em estágios avançados da doença, quando o paciente sofre com o desconforto causado pelo tumor e busca ajuda médica. Apesar dos recentes avanços em tratamentos e procedimentos cirúrgicos, a taxa de sobrevivência ainda é inferior a 60% em um período de cinco anos [Ries et al. 1998]. Além disso, devido ao diagnóstico tardio e à agressividade do tumor (e do tratamento), pacientes são frequentemente submetidos a remoção de tecidos que além de causar deformidades faciais, impactam a capacidade de falar, engolir e mastigar [Foundation 2023]. Isso resulta não apenas em sequelas cosméticas e funcionais, mas também deixa marcas emocionais.

Por outro lado, se diagnosticado precocemente, as chances de sobrevivência sobem para 80-90% [Foundation 2023].

Objetivando contribuir para a detecção precoce do câncer de boca, encontra-se em desenvolvimento no PPGC-UFRGS em parceria com a Faculdade de Odontologia um sistema (*Oral Cancer Screening – OCS*) baseado na arquitetura apresentada na Figura 6.1. O desenvolvimento deste projeto foi autorizado pelo Comitê de Ética em Pesquisa da UFRGS (parecer CAAE - 39212420.9.0000.5347). Um aplicativo para smartphones permite que dentistas, ao perceberem algo incomum durante o exame de um paciente, fotografem as regiões suspeitas e enviem as fotos para avaliação em nosso servidor (Figura 6.2). Conforme apresentado na Seção 6.2, os resultados das avaliações são então disponibilizados aos dentistas, indicando as regiões das fotografias que contém elementos suspeitos. Quando apropriado, a resposta do sistema também inclui uma recomendação para realização de biópsia. Fotos de lesões acompanhadas por laudos de biópsias são utilizadas para aprimorar o treinamento do modelo, contribuindo para melhorar sua acurácia.

A comunicação entre os componentes do sistema permite a integração de diferentes tecnologias que complementam suas funcionalidades. A solução é constituída por três módulos principais: *interface com usuário* (*front end*), *servidor de dados* (*back end*) e *modelo de predição*. O sistema possibilita o armazenamento dos dados de usuários (dentistas), pacientes, imagens submetidas, resultados das predições, recomendações, e laudos submetidos, além de possuir uma interface para acessos a estes dados. A seguir, é apresentado um detalhamento dos três módulos principais, os quais encontram-se representados na Figura 6.3.

6.3.1. Interface com o Usuário

A interação dos usuários com o sistema ocorre prioritariamente via smartphone por meio de app distribuído através das lojas de aplicativos (veja Figura 6.1) e disponível para as plataformas Android e iOS. O sistema também pode ser acessado através de uma interface web por meio de uma aplicação desktop. O aplicativo móvel foi desenvolvido utilizando o *framework React Native* enquanto a aplicação desktop foi desenvolvida utilizando o *framework ReactJS*. O aplicativo móvel permite a captura de fotos de lesões de boca observadas pelo dentista utilizando a câmera do smartphone e o seu envio para avaliação pelo modelo de predição. Ele também oferece acesso a todas as demais funcionalidades do sistema, incluindo o envio de laudos referentes a exames e o acesso a dados de pacientes (*e.g.,* imagens e laudos submetidos e resultados de predições). Por se tratar de um sistema multiplataforma, todas essas funcionalidades, exceto a captura de imagens, também está disponível na versão desktop, a qual é disponibilizada para conveniência dos usuários. Embora o aplicativo seja disponibilizado livremente, o acesso aos serviços descritos requer cadastramento, o que pode ser feito via interface com o usuário. Um curador analisa as solicitações de cadastramento, verifica os dados e certifica o usuário (dentista), como forma de garantir que o sistema seja utilizado por profissionais habilitados. A partir da certificação, é possível fazer *login* e acessar todas as funcionalidades do sistema.

A Figura 6.2 ilustra, de modo simplificado, os passos envolvidos no processo de suporte à detecção precoce de câncer de boca. Neste exemplo, o aplicativo é utilizado pelo dentista para fotografar uma lesão observada na língua de um paciente e enviar a

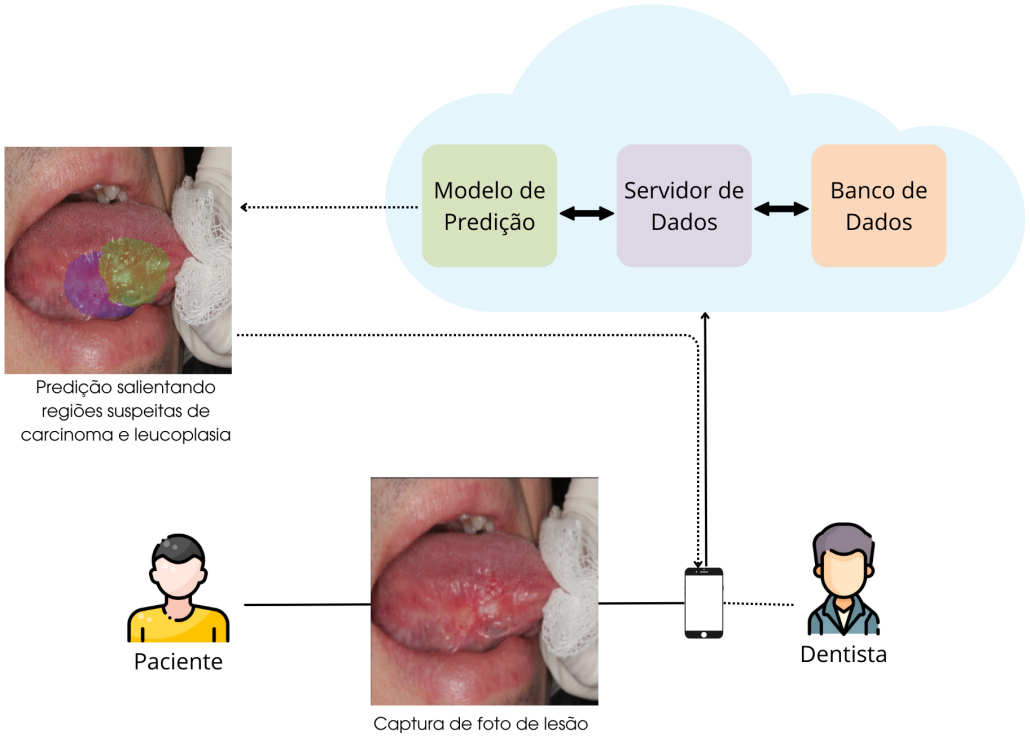


Figura 6.2: Visão simplificada do uso do sistema OCS. Dentista utiliza o smartphone para capturar foto de lesão na boca (língua) de paciente e enviá-la para avaliação pelo sistema. O resultado da predição é disponibilizado para o dentista, salientando regiões suspeitas.

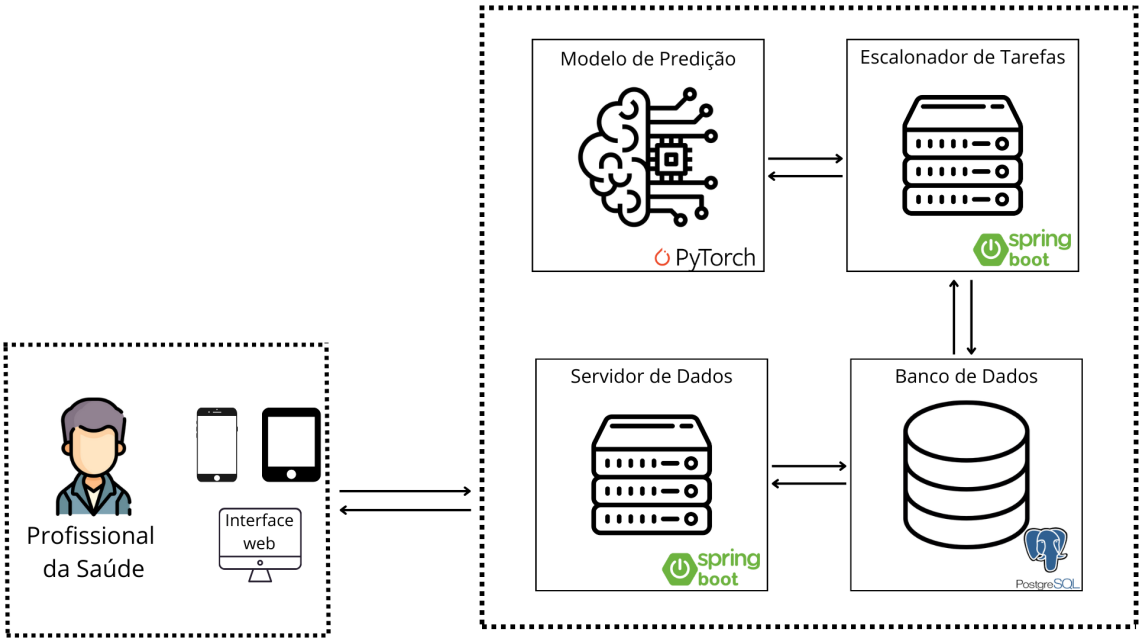


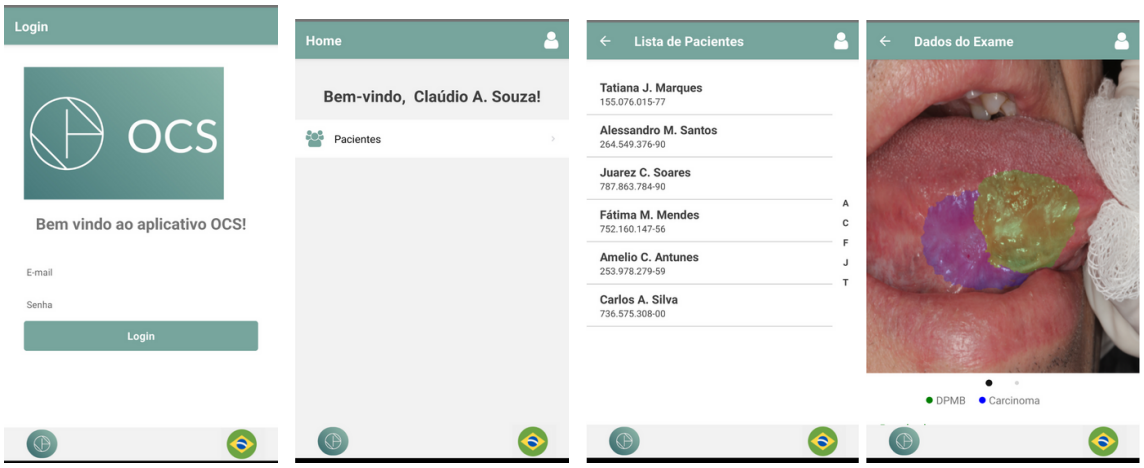
Figura 6.3: Componentes do Sistema OCS. Interface com o usuário (esquerda). Servidor de dados, modelo de predição, escalonador de tarefas e banco de dados (direita).

imagem para análise pelo sistema. O resultado da análise (predição) é disponibilizado para o dentista salientando regiões suspeitas de ocorrência de câncer e outras lesões ou

desordens potencialmente malignas de boca, que podem evoluir para câncer (Figura 6.2). Neste exemplo, o sistema recomenda ao dentista a realização de biópsia para confirmar as suspeitas. Espera-se que o dentista forneça o laudo da biópsia, para que um curador de dados possa acrescentar a imagem ao banco de imagens utilizado para treinamento do modelo (veja Figura 6.1), realimentando o processo com o objetivo de melhorar a acurácia do sistema. Note que tal retorno é importante independentemente do resultado da biópsia: em caso de existência de lesão maligna, a imagem reforçará a confiança do sistema na predição; caso contrário, a imagem contribuirá para ajustar a predição por meio de contraexemplo.

A Figura 6.4 apresenta algumas telas do aplicativo OCS. A Figura 6.4 (a) mostra a tela inicial (*login*). Após a verificação das credenciais (e-mail e senha), o usuário pode acessar os dados dos seus pacientes cadastrados (b), os quais são apresentados em ordem alfabética (c). A Figura 6.4 (d) mostra o resultado da predição realizada pelo modelo para uma foto de um dos pacientes.

Conforme mencionado na Seção 6.2, as lojas de aplicativos têm papel fundamental como canal de distribuição para tornar o sistema acessível a pessoas independente de suas localizações geográficas. Contribuindo para isso, o aplicativo e a aplicação web são disponibilizadas com versões em Português e em Inglês, podendo ser facilmente customizados para outros idiomas. Com o aumento do número de usuários, espera-se um aumento no número de realimentações (laudos) por parte dos usuários. Isto deverá contribuir para um incremento na quantidade e diversidade das imagens utilizadas para treinamento do modelo, com consequente impacto na acurácia do sistema.



(a) (b) (c) (d)

Figura 6.4: Exemplos de telas do aplicativo OCS para dispositivos móveis. (a) Tela inicial (login) solicitando dados de usuário e senha. (b) Tela mostrada após a verificação das credenciais do usuário. (c) Lista de pacientes (de um dentista). (d) Resultado da predição para uma foto de paciente, salientando regiões e indicando os tipos de lesões suspeitas.

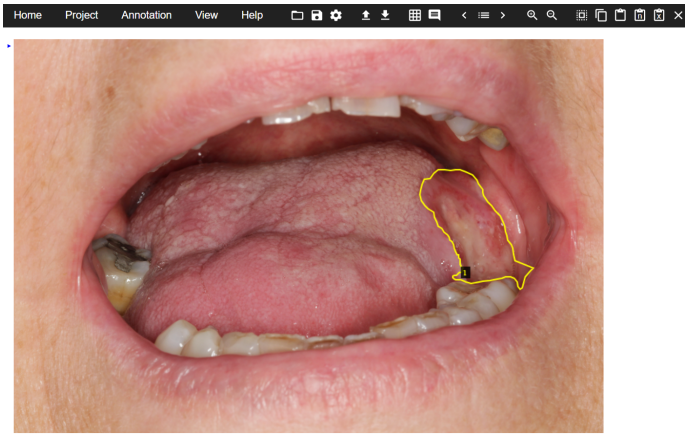


Figura 6.5: Exemplo de imagem anotada por especialista utilizando o software VIA [Dutta et al. 2016]. A região contendo uma lesão é delimitada por linha poligonal fechada simples e associada a um tipo de específico de lesão. Neste exemplo, trata-se de carcinoma espinocelular.

6.3.2. Servidor de Dados

Um banco de dados armazena as imagens, os resultados de predições e os laudos associados a cada paciente, os quais podem ser consultados pelos usuários através de um servidor de dados. Um escalonador de tarefas é responsável por verificar periodicamente a existência de solicitações de avaliação ainda não processadas e encaminhá-las, em ordem de chegada, para o modelo de predição (Figura 6.3). Os resultados das predições são coletados e armazenados pelo servidor no banco de dados.

O banco de dados foi implementado utilizando PostgreSQL, um gerenciador de bancos de dados relacionais disponibilizado como software livre e que apresenta características de robustez, segurança e extensibilidade. O servidor de dados foi desenvolvido utilizando o framework *Spring Boot* da linguagem Java. Este servidor possui um conjunto de funções (API) responsáveis por disponibilizar dados para os demais processos utilizando uma arquitetura cliente/servidor e protocolo de comunicação HTTP.

6.3.3. Modelo de Predição

O modelo de predição corresponde a uma rede neural convolucional treinada para detectar lesões de boca (Figura 6.4d). O treinamento está sendo realizado utilizando um conjunto de imagens (contendo lesões) fornecidas e anotadas por especialistas da Faculdade de Odontologia da UFRGS. Nestas imagens, cada lesão é delimitada por uma linha poligonal fechada e contém um código que identifica o tipo de lesão (Figura 6.5). A partir destas anotações, são geradas automaticamente máscaras que definem os pixels correspondentes a cada lesão. Estas máscaras são utilizadas juntamente com as imagens para o treinamento supervisionando do modelo. A versão atual do módulo de predição busca identificar duas classes de lesões: *carcinoma espinocelular* (câncer) e *leucoplasia* (uma forma de lesão que pode eventualmente evoluir para câncer). Em uma etapa posterior, pretende-se estender a classificação para incluir outras formas de desordens potencialmente malignas, neoplasias benignas, e outros tipos de lesões não suspeitas de malignidade.

6.3.4. Avaliação de Imagens de Microscopia

A arquitetura proposta também pode ser utilizada com imagens de microscopia para auxílio na detecção precoce de câncer de boca. Para tanto, basta realizar a substituição do modelo de predição.

A citopatologia pode ajudar a detectar os primeiros sinais de desenvolvimento de câncer de boca. O número de Regiões Organizadoras Nucleolares Argirófilas (AgNORs) encontradas no núcleo das células indica o quão rapidamente estas células estão se replicando e serve como um indicador de lesões com potencial maligno [Jajodia et al. 2017]. Dada seu menor custo em relação a outras técnicas, a coloração de AgNORs é uma opção atraente, especialmente para países em desenvolvimento. No entanto, a contagem manual, ainda utilizada atualmente, envolve o trabalho de um especialista (citopatologista), sendo um processo demorado, cansativo e sujeito a erros. Visando eliminar estas limitações, desenvolvemos um método automático que utiliza uma CNN para segmentar e contar o número de núcleos e de AgNORs em cada núcleo em imagens de lâminas de microscopia. O modelo resultante apresenta desempenho similar ao de citopatologistas, sendo, entretanto, significativamente mais rápido. A Figura 6.6 compara os resultados da segmentação automática de núcleos e AgNORs realizados pela técnica desenvolvida (Resultado) contra segmentações de referência (Referência) realizadas manualmente por especialistas para um conjunto de imagens de lâminas citológicas (Entrada). Os núcleos, AgNORs e o fundo são mostrados nas cores laranja, azul e cinza, respectivamente. Note o alto grau de concordância das segmentações. A variabilidade de cores, contraste, e nível de ruído nas imagens de entrada atesta a capacidade de generalização do modelo. Uma descrição detalhada desta técnica e de seus resultados podem ser encontradas em [Rönnau et al. 2023].

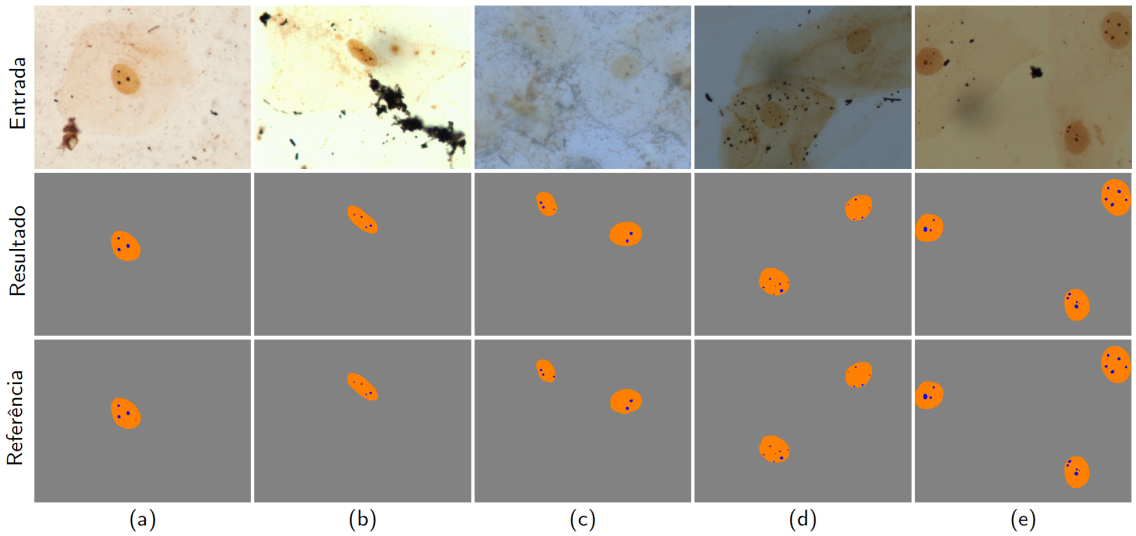


Figura 6.6: Exemplos de segmentação automática de núcleos e AgNORs em células da mucosa oral usando nosso modelo para AgNORs. Entrada: Imagens de lâminas citológicas coradas com AgNOR. Resultado: Segmentação automática produzida pelo nosso modelo. Referência: Segmentação manual realizada por especialistas (padrão ouro). Note o alto grau de concordância entre as segmentações.

Além da contagem de AgNORs, também investigamos a utilização de outros características celulares que possam, potencialmente, indicar a ocorrência de sinais precursores de câncer de boca. A existência de núcleos com maior volume ou de aglomerados pode ser um indicador de malignidade. Para tentar detectar estes sinais, desenvolvemos mais um modelo preditivo baseado em CNN para classificação de núcleos e citoplasmas de células da mucosa bucal coradas pela técnica de Papanicolaou. A Figura 6.7 ilustra alguns resultados obtidos através de segmentação automática produzida pelo nosso modelo, comparando-os com a segmentação manual realizada por citopatologistas para um conjunto de imagens contendo características variadas. Observe-se, também neste caso, o alto grau de concordância entre as segmentações. Nos exemplos da Figura 6.7, a cor laranja corresponde ao citoplasma de uma célula isolada, ao passo que a cor azul (mais escuro) representa o citoplasma de aglomerados celulares (grupos de células em contato direto). Os núcleos representados em vermelho indicam células superficiais, ao passo que os núcleos em ciano estão associados a células intermediárias. O sistema também é capaz de detectar escamas (células anucleadas), bem como células suspeitas de malignidade (com núcleos com maior volume).

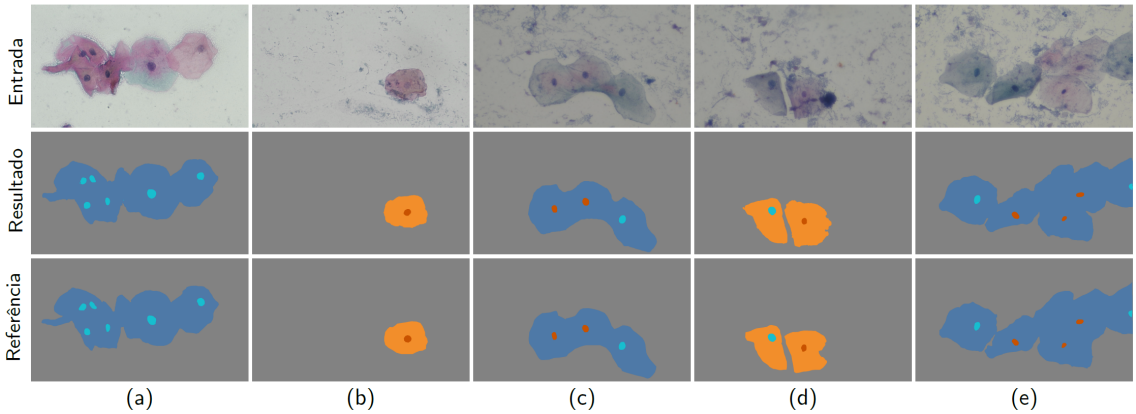


Figura 6.7: Exemplos de segmentação e classificação de células da mucosa bucal coradas pela técnica de Papanicolaou. Resultado: Segmentação e classificação automática produzida pelo nosso modelo. Referência: Segmentação e classificação manual produzidas por especialistas (padrão ouro). Mais uma vez, observe-se o elevado grau de concordância entre os dois resultados.

O processamento de imagens de células coradas pelas técnicas de AgNOR e Papanicolaou fornecem informações relevantes para a tomada de decisão de citopatologistas. A automatização desses processos permite a liberação dos profissionais para realização de outras tarefas. Também permite que os respectivos testes sejam realizados com maior rapidez e de maneira escalável, atendendo um maior número de pacientes.

6.4. Iniciativas Relacionadas

Esta seção discute algumas iniciativas relacionadas à arquitetura de sistema aqui descrito.

Haron et al. (2020) [Haron et al. 2020] desenvolveram um aplicativo para dispositivos móveis chamado de MeMoSA (*Mobile Mouth Screening Anywhere*) cujo objetivo é contribuir para a identificação de câncer de boca. O aplicativo permite a captura de fotos

da boca do paciente, as quais são enviadas para análise por especialistas que emitem um diagnóstico. O aplicativo não detecta regiões suspeitas. A análise de imagens é realizada por especialistas humanos, o que limita a escalabilidade desta solução. Além disso, os diagnósticos são emitidos a partir de análise de imagens sem a realização de biópsias.

Welikala et al. (2020) [Welikala et al. 2020] utilizaram a base de imagens capturadas com o aplicativo MeMoSA para treinar uma rede neural para detecção de câncer de boca. As imagens capturadas pelo aplicativo foram anotadas por especialistas e separadas em um conjunto de treinamento e um conjunto de teste. Não há registros de que a rede resultante tenha sido incorporada pelo projeto do MeMoSA ou disponibilizada como um serviço online.

Warin et al. (2021) [Warin et al. 2021] utilizaram uma CNN para classificação de existência ou não de câncer de boca. A CNN foi treinada com 700 imagens anotadas por especialistas e divididas igualmente entre imagens com e sem câncer de boca. Em um trabalho subsequente dos mesmos autores [Warin et al. 2022], as imagens são classificadas entre contendo desordens potencialmente malignas, contendo carcinoma, ou sem identificação de patologias. Huang et al. (2023) [Huang et al. 2023] treinaram uma CNN para classificação de imagens com relação à presença de câncer de boca utilizando 130 imagens disponíveis na plataforma Kaggle [Dataset 2023]. Lin et al. (2021) [Lin et al. 2021] utilizaram uma CNN (HRNet [Wu et al. 2021]), treinada a partir de imagens capturadas com diferentes modelos de smartphones e anotadas por especialistas, para classificar lesões de boca. A CNN classifica as imagens entre contendo: carcinoma, úlcera aftosa, mucosa normal, e lesões potencialmente malignas de alto e de baixo risco. Todos estes trabalhos descrevem o treinamento de CNNs que foram avaliadas de modo isolado utilizando apenas seus respectivos conjuntos de testes. Não há registros de que essas redes tenham sido utilizadas em serviços online.

Considerando outras áreas em telessaúde, Hacisoftaoglu et al. (2020) [Hacisoftaoglu et al. 2020] utilizaram uma CNN para a identificação de retinopatia diabética em imagens de fundo de olho. As imagens para treinamento foram capturadas usando diferentes modelos de câmeras do tipo *digital single lens reflex* (DSLR). Porém, o objetivo é fornecer como entrada para a CNN imagens capturadas por smartphones, o que é feito com o auxílio de diferentes equipamentos oftalmológicos. A classificação feita pela CNN é binária, ou seja, a imagem é classificada como apresentando ou não retinopatia diabética.

Archibong et al. (2017) [Archibong et al. 2017] utilizaram dispositivos móveis e técnicas convencionais de processamento de imagens (*i.e.*, sem o uso de aprendizagem de máquina) para a identificação de hemólise (processo de dissolução ou destruição de glóbulos vermelhos do sangue). Este processo é caracterizado pela elevada presença de enzimas hepáticas e baixa contagem de plaquetas. As imagens foram capturadas por um smartphone acoplado a um dispositivo que contém a amostra e processadas pelo smartphone. O resultado é exibido na tela do aparelho. O processo requer calibração da câmera, que é feita através de uma curva de calibração específica para cada modelo de smartphone.

Várias das iniciativas descritas acima utilizam CNNs para classificação de imagens e algumas utilizam smartphones para captura de imagens. Entretanto, nenhuma de-

las fornece uma solução integrada e escalável para detecção automática de doenças como mostrado na Figura 6.1.

6.5. Conclusão

Este trabalho apresentou uma proposta de arquitetura para construção de sistemas de telessaúde escaláveis que podem ser customizados para detecção de diversos tipos de doenças. A solução proposta combina três importantes componentes: (i) modelos de aprendizagem de máquina, os quais podem ser treinados e customizados para identificar uma grande variedade de doenças; (ii) a Internet, que disponibiliza a infraestrutura de comunicação; e (iii) os smartphones e aplicativos, que oferecem uma interface versátil entre os usuários e os serviços disponibilizados. Esta combinação, ilustrada na Figura 6.1, permite a oferta de serviços para suporte à detecção de diversos tipos de doenças de forma automatizada, ininterrupta, e com baixo custo.

A arquitetura proposta se beneficia da realimentação provida por seus usuários, permitindo que o sistema evolua, melhorando assim a acurácia de suas predições. Sua implementação envolve custos relativamente modestos. Estes incluem: (i) a disponibilização de um servidor para hospedagem do banco de dados, do servidor de dados e do modelo de predição; (ii) a construção de um ou mais bancos de imagens anotadas por especialistas para doenças específicas; (iii) a especificação e treinamento de um ou mais modelos de predição a partir dos dados anotados; e (iv) o desenvolvimento do servidor de dados, de um ou mais aplicativos para dispositivos móveis, e, opcionalmente, de uma aplicação web para desktop.

Um sistema para suporte à detecção precoce de câncer de boca baseado na arquitetura proposta está sendo desenvolvido no PPGC em parceria com a Faculdade de Odontologia da UFRGS. Os componentes desenvolvidos para este sistema (aplicativo para dispositivo móvel, aplicação web para desktop, servidor de dados, e CNN para predição) podem ser utilizados como referência para o desenvolvimento de novos sistemas.

A possibilidade de treinar modelos de aprendizagem de máquina para detecção precoce de doenças como o câncer pode contribuir para evitar a ocorrência de mortes prematuras e para uma melhoria da qualidade de vida de pacientes. A estratégia aqui descrita representa uma evolução natural e necessária aos serviços de telessaúde. Dada a sua aplicabilidade a vários cenários, baixo custo, facilidade de implementação e potencial de alcance em escala global, acreditamos que ela pode desempenhar um papel relevante na ampliação e democratização de serviços de saúde nos próximos anos.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. O presente trabalho foi realizado com apoio do CNPq [305474/2022-7], FAPERGS [51956.616.19834].

Referências

[Abbas 2017] Abbas, Q. (2017). Glaucoma-deep: detection of glaucoma eye disease on retinal fundus images using deep learning. *International Journal of Advanced Compu-*

ter Science and Applications, 8(6).

[Archibong et al. 2017] Archibong, E., Konnaiyan, K. R., Kaplan, H., and Pyayt, A. (2017). A mobile phone-based approach to detection of hemolysis. *Biosensors and bioelectronics*, 88:204–209.

[Bhatia et al. 2019] Bhatia, S., Sinha, Y., and Goel, L. (2019). Lung cancer detection: a deep learning approach. In *Soft Computing for Problem Solving: SocProS 2017, Volume 2*, pages 699–705. Springer.

[Dataset 2023] Dataset, K. (2023). Mouth cancer images. https://www.kaggle.com/datasets/edward112/mouth-cancer-images. Setembro de 2023.

[Dildar et al. 2021] Dildar, M., Akram, S., Irfan, M., Khan, H. U., Ramzan, M., Mahmood, A. R., Alsaiari, S. A., Saeed, A. H. M., Alraddadi, M. O., and Mahnashi, M. H. (2021). Skin cancer detection: a review using deep learning techniques. *International journal of environmental research and public health*, 18(10):5479.

[Dutta et al. 2016] Dutta, A., Gupta, A., and Zissermann, A. (2016). VGG image annotator (VIA). http://www.robots.ox.ac.uk/ vgg/software/via/. Version: 2.0.12, Accessed: 2023.

[Ebrahimighahnavieh et al. 2020] Ebrahimighahnavieh, M. A., Luo, S., and Chiong, R. (2020). Deep learning to detect alzheimer's disease from neuroimaging: A systematic literature review. *Computer methods and programs in biomedicine*, 187:105242.

[Foundation 2023] Foundation, T. O. C. (2023). Cancer screening protocols. https://oralcancerfoundation.org/discovery-diagnosis/cancer-screening-protocols/. Julho de 2023.

[Gargeya and Leng 2017] Gargeya, R. and Leng, T. (2017). Automated identification of diabetic retinopathy using deep learning. *Ophthalmology*, 124(7):962–969.

[Hacisoftaoglu et al. 2020] Hacisoftaoglu, R. E., Karakaya, M., and Sallam, A. B. (2020). Deep learning frameworks for diabetic retinopathy detection with smartphone-based retinal imaging systems. *Pattern recognition letters*, 135:409–417.

[Haron et al. 2020] Haron, N., Zain, R. B., Ramanathan, A., Abraham, M. T., Liew, C. S., Ng, K. G., Cheng, L. C., Husin, R. B., Chong, S. M. Y., Thangavalu, L. A., et al. (2020). m-health for early detection of oral cancer in low-and middle-income countries. *Telemedicine and e-Health*, 26(3):278–285.

[Huang et al. 2023] Huang, Q., Ding, H., and Razmjooy, N. (2023). Optimal deep learning neural network using issa for diagnosing the oral cancer. *Biomedical Signal Processing and Control*, 84:104749.

[Ismael and Şengür 2021] Ismael, A. M. and Şengür, A. (2021). Deep learning approaches for covid-19 detection based on chest x-ray images. *Expert Systems with Applications*, 164:114054.

[Jajodia et al. 2017] Jajodia, E., Raphael, V., Shunyu, N. B., Ralte, S., Pala, S., and Jitani, A. K. (2017). Brush cytology and agnor in the diagnosis of oral squamous cell carcinoma. *Acta cytologica*, 61(1):62–70.

[Le 1906] Le, E. (1906). Le télécardiogramme [the telecardiogram]. *Arch. Int. Physiol.*, 4:132–164.

[Lin et al. 2021] Lin, H., Chen, H., Weng, L., Shao, J., and Lin, J. (2021). Automatic detection of oral cancer in smartphone-based images using deep learning for early diagnosis. *Journal of Biomedical Optics*, 26(8):086007–086007.

[Maldonado et al. 2016] Maldonado, J. M. S. d. V., Marques, A. B., and Cruz, A. (2016). Telemedicine: challenges to dissemination in brazil. *Cadernos de saude publica*, 32:e00155615.

[Mambou et al. 2018] Mambou, S. J., Maresova, P., Krejcar, O., Selamat, A., and Kuca, K. (2018). Breast cancer detection using infrared thermal imaging and a deep learning model. *Sensors*, 18(9):2799.

[Ozturk et al. 2020] Ozturk, T., Talo, M., Yildirim, E. A., Baloglu, U. B., Yildirim, O., and Rajendra Acharya, U. (2020). Automated detection of covid-19 cases using deep neural networks with x-ray images. *Computers in Biology and Medicine*, 121:103792.

[Ries et al. 1998] Ries, L. A. G., Kosary, C., Hankey, B., Miller, B., and Edwards, B. (1998). Seer cancer statistics review, 1973-1995. *Bethesda, MD: National Cancer Institute*.

[Rönnau et al. 2023] Rönnau, M. M., Lepper, T. W., Amaral, L. N., Rados, P. V., and Oliveira, M. M. (2023). A cnn-based approach for joint segmentation and quantification of nuclei and nors in agnor-stained images. *Computer Methods and Programs in Biomedicine*, 242:107788.

[Roser and Ritchie 2015] Roser, M. and Ritchie, H. (2015). Cancer. *Our World in Data*. https://ourworldindata.org/cancer.

[Sung et al. 2021] Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., and Bray, F. (2021). Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 71(3):209–249.

[Warin et al. 2021] Warin, K., Limprasert, W., Suebnukarn, S., Jinaporntham, S., and Jantana, P. (2021). Automatic classification and detection of oral cancer in photographic images using deep learning algorithms. *Journal of Oral Pathology & Medicine*, 50(9):911–918.

[Warin et al. 2022] Warin, K., Limprasert, W., Suebnukarn, S., Jinaporntham, S., Jantana, P., and Vicharueang, S. (2022). Ai-based analysis of oral lesions using novel deep convolutional neural networks for early detection of oral cancer. *Plos one*, 17(8):e0273508.

[Welikala et al. 2020] Welikala, R. A., Remagnino, P., Lim, J. H., Chan, C. S., Rajendran, S., Kallarakkal, T. G., Zain, R. B., Jayasinghe, R. D., Rimal, J., Kerr, A. R., et al. (2020). Automated detection and classification of oral lesions using deep learning for early detection of oral cancer. *IEEE Access*, 8:132677–132693.

[WHO 1998] WHO (1998). *A Health Telematics Policy in Support of WHO'S Health-For-All Strategy for Global Development: Report of the WHO Group Consultation on Health Telematics*. World Health Organization.

[WHO 2010] WHO (2010). *Telemedicine: opportunities and developments in member states. Report on the second global survey on eHealth.* World Health Organization.

[Wu et al. 2021] Wu, H., Liang, C., Liu, M., and Wen, Z. (2021). Optimized hrnet for image semantic segmentation. *Expert Systems with Applications*, 174:114532.

Capítulo 7

Segurança Cibernética 2030: Experiências, Desafios e Oportunidades

Alberto Egon Schaeffer-Filho, Jéferson Campos Nobre, Juliano Araújo Wickboldt, Lisandro Zambenedetti Granville, Luciano Paschoal Gaspary, Weverton Luis da Costa Cordeiro

Programa de Pós-Graduação em Computação (PPGC) - UFRGS

Abstract

Cybersecurity has assumed an increasingly critical role as a fundamental pillar of a digital society, deeply interconnected and increasingly dependent on services provisioned via consolidated (e.g., 4G/5G) and emerging (such as artificial intelligence) technologies and concepts. In this context, expectations have increased that computing can contribute to solving emerging challenges in cybersecurity, especially those challenges intrinsically influenced by the particularities of Brazilian society. In this sense, there is great expectation about how computational solutions can support cybersecurity professionals and researchers in solving the challenges that plague our digital society, such as fake news, cyber scams, identity theft, data theft, privacy violations, etc. This chapter will address the cybersecurity research landscape, highlighting the opportunities and challenges that are relevant for the next decade: People-centric security, Artificial intelligence and security and Security in the era of programmable networks.

Resumo

A cibersegurança tem assumido um papel cada vez mais crítico como pilar fundamental de uma sociedade digital, profundamente interconectada e cada vez mais dependente de serviços provisionados via tecnologias e conceitos consolidados (por ex., 4G/5G) e emergentes (como inteligência artificial). Neste contexto, aumentaram as expectativas de que a computação possa contribuir na solução dos desafios emergentes em segurança cibernética, em especial àqueles desafios intrinsecamente influenciados pelas

Vídeo com a apresentação do capítulo: `https://youtu.be/uEyOldWYauI`

particularidades da sociedade brasileira. Nesse sentido, há grande expectativa sobre como as soluções computacionais poderão apoiar profissionais e pesquisadores em cibersegurança a resolverem os desafios que afligem nossa sociedade digital, tais como fake news, golpes cibernéticos, usurpação de identidade, roubo de dados, violações de privacidade, etc. O presente capítulo abordará o panorama de pesquisa em cibersegurança, destacando as oportunidades e desafios que se impõem como relevantes para a próxima década: Segurança Centrada nas Pessoas, Inteligência artificial e segurança e Segurança na era de redes programáveis.

7.1. Introdução

A importância da cibersegurança tem crescido significativamente à medida que nossa sociedade se torna cada vez mais interligada e dependente de tecnologias estabelecidas e emergentes. Evidentemente, tal crescimento traz um grande número de oportunidades assim como de riscos. O ciberespaço constitui um cenário promissor pela prática de toda sorte de ações ilícicas, as quais não respeitam fronteiras geopolíticas tradicionais. Dessa forma, ataques cibernéticos exploram as vulnerabilidades das estruturas de Tecnologia da Informação e Comunicação. Como exemplos desses ataques, podem ser citados fake news, golpes cibernéticos, usurpação de identidade, roubo de dados, violações de privacidade, etc.

Os desafios em cibersegurança oferecem uma oportunidade para "repensar"o papel da computação na própria sociedade. Nesse sentido, há grande expectativa sobre como a computação como área desempenhará a abordagem dos desafios emergentes em cibersegurança, produzindo soluções que poderão apoiar profissionais e pesquisadores em cibersegurança. Mesmo sendo na sua maioria globais, é necessária uma atenção especial em aspectos são influenciados diretamente por características específicas da sociedade brasileira.

Considerando o panorama de pesquisa em cibersegurança, alguns pontos podem ser destacados. Primeiro, a emergência da Segurança Centrada nas Pessoas. Assim, vislumbra-se a pesquisa e o projeto de mecanismos de segurança que consideram de forma central aspectos humanos e sociais. Segundo, as relações entre Inteligência Artificial e Cibersegurança. Neste contexto, a IA impacta substancialmente a cibersegurança, tanto de forma positiva quanto negativa. Finalmente, a segurança na era de redes programáveis. Os avanços recentes na internet ampliaram nossa capacidade de modificá-la, sendo necessária a verificação e garantia de propriedades de segurança. Os pontos destacados merecem ser discutidos a fim de se buscar uma compreensão da evolução da cibersegurança nos próximos anos.

O presente capítulo está organizado da seguinte forma. No Capítulo 7.2, será discutida a Segurança Centrada nas Pessoas. No Capítulo 7.4, serão apresentados aspectos sobre a relação entre Inteligência Artificial e Cibersegurança. No Capítulo 7.3, a Segurança na era de redes programáveis será abordada. Finalmente, as oportunidades e desafios que se impõem como relevantes para a próxima década na cibersegurança são discutidos no Capítulo 7.5.

7.2. Segurança Centrada nas Pessoas

A Segurança Centrada nas Pessoas é a cibersegurança projetada com as pessoas em mente. Tradicionalmente, os mecanismos e controles de segurança costumam ser projetados sob suposições ingênuas a respeito dos humanos: que os mesmos sempre agem de forma lógica, racional e fazendo o melhor para si. Infelizmente, isso não é o caso em uma grande quantidade de eventos. Em ambientes organizacionais, as principais motivações de colaboradores são metas, necessidades dos clientes, etc. Assim, se houver medidas de cibersegurança que se oponham em relação a tais motivações, soluções alternativas serão buscadas. Dessa forma, é necessário que a cibersegurança seja desenvolvida considerando que sua abordagem seja funcional e adaptada às pessoas, e não o contrário.

O desenvolvimento de mecanismos de cibersegurança centrados nas pessoas vislumbra a pesquisa e o projeto de técnicas de segurança personalizadas, considerando aspectos humanos e sociais. Isto é necessário porque se a cibersegurança não está implicada para as pessoas, muito riscos estarão associados. Por exemplo, fatores que afetam o comportamento humano e aumentam a suscetibilidade das pessoas à manipulação precisam ser considerados na produção de sistemas computacionais. Excluindo-se tais fatores, a possibilidade de explorar o comportamento humano para obtenção de dados e informações relevantes de potenciais alvos é aumentada, além facilitar a realização de ações pelos colaboradores que colocam as organizações em risco.

Políticas de segurança da informação frequentemente são elaboradas sem um entendimento de como as pessoas realmente trabalham. As pessoas moldam a cibersegurança ao criar soluções alternativas, porque seguir a política muitas vezes dificulta a realização de suas tarefas laborais. Na prática, o fator humano é o elo mais fraco na cadeia de cibersegurança [Mitnick and Simon 2003]. No entanto, a Segurança Centrada nas Pessoas busca inverter esta lógica e colocar o humano como centro dos processos de cibersegurança. Tal inversão frequentemente promove a conscientização e o envolvimento das pessoas de forma colaborativa.

A Segurança Centrada nas Pessoas ajuda os profissionais de segurança a projetarem sistemas e políticas que funcionem considerando as características dos humanos. Como as pessoas não são normalmente motivadas pela segurança, são necessárias estratégias de convencimento (e.g., campanhas de conscientização), além de políticas de segurança da informação claras. Assim, é necessário que os profissionais de cibersegurança estejam dispostos a produzir essas políticas com ênfase nas pessoas por design.

A presente seção está organizada da seguinte forma. Inicialmente, serão discutidos os ataques focados nas pessoas, especialmente no que tange à Engenharia Social. Em seguida, aspectos relacionados com a automatização da engenharia social e os Grande Modelos de Linguagem serão apresentados. Finamente, estratégias para prevenção e mitigação de ataques focados nas pessoas serão comentadas.

7.2.1. Ataques Focados nas Pessoas: uma Introdução à Engenharia Social

Engenharia Social é caracterizada como a prática de aproveitar aspectos humanos com o intuito de obter acesso a dados e informações de possíveis alvos em sistemas de informação, independentemente do uso de tecnologia. Trata-se de uma abordagem de ataque

que se baseia na exploração do comportamento humano, utilizando persuasão e manipulação psicológica. Os ataques de Engenharia Social colocam o atacante em uma posição favorecida no fluxo de informações, tirando proveito de uma relação de confiança. O desenvolvimento de uma relação de confiança faz uso da manipulação psicológica induzindo as pessoas realizarem ações específicas.

O processo de proteção de dados e informações, visando garantir sua confidencialidade, integridade e disponibilidade, está intrinsecamente relacionado à Cibersegurança. A relação entre a Engenharia Social e a Cibersegurança é ainda mais reforçada pelo contínuo desenvolvimento tecnológico, que tem possibilitado a automação e escalabilidade dos ataques de Engenharia Social, tornando-os desafios cada vez mais difíceis de combater [Beal 2005]. Em situações reais, é importante reconhecer que o elemento humano frequentemente representa o ponto mais vulnerável na cadeia de segurança cibernética. [Mitnick and Simon 2003] [Klimburg-Witjes and Wentland 2021].

A ampla disponibilidade de diversos meios de comunicação de grande alcance cria um ambiente propício para os ataques de Engenharia Social. O avanço da tecnologia tem facilitado a automação e escalabilidade desses ataques, permitindo que os invasores alcancem um grande número de possíveis vítimas em um curto espaço de tempo [Pinheiro 2020]. Portanto, compreender e se proteger contra os ataques de Engenharia Social tornase essencial para garantir a segurança de dados e sistemas em ambientes conectados e dependentes da tecnologia.

Os engenheiros sociais podem utilizar a automação para desenvolver ferramentas pré-programadas para realizar tarefas sem a intervenção humana, possibilitando a escalabilidade dos ataques. Tais técnicas podem, por exemplo, considerar o uso de chatbots, tanto como ferramenta para ataques, como para auxílio dos profissionais. Além disso, devem ser considerados aspectos da interação das ferramentas e processos de segurança com o comportamento humano.

Os ataques automatizados podem ser preparados utilizando informações coletadas ou através da influência sobre indivíduos nas redes sociais. Essas redes representam um espaço virtual que pode ser usado para os atacantes explorarem vulnerabilidades técnicas e a falta de conhecimento e conscientização dos usuários sobre ações de Engenharia Social. Por exemplo, uma das vulnerabilidades que são encontradas em redes sociais é a criação de perfis falsos, os quais constituem um percentual significativo dos usuários dessas redes.

7.2.2. Automatização como uma Evolução da Engenharia Social

A crescente conectividade e automação revolucionaram as infraestruturas econômicas e culturais do mundo, ao mesmo tempo em que introduziram riscos em termos de ataques cibernéticos. A Engenharia Social Automatizada representa uma abordagem que combina técnicas de Engenharia Social com a automação, usando ferramentas e scripts para criar ataques eficazes em grande escala. Os ataques de Engenharia Social tradicionais requerem investimento de tempo e recursos para estabelecer uma relação de confiança entre o atacante e o usuário. Portanto, ao automatizar os aspectos repetitivos e monótonos desse processo, os agressores aproveitam para realizar ataques em larga escala de forma mais eficiente [Guzman and Lewis 2020].

A comunicação humana tem sido a base para o desenvolvimento de interfaces homem-máquina. Neste contexto, as redes sociais facilitam a comunicação, a interação social e o compartilhamento de informações pessoais e informações corporativas, aumentando sua popularidade no ambiente cibernético. As conexões formadas nesses ambientes virtuais de socialização permitem um grande troca de informações, reforçando o papel das redes como estruturas comunicativas para as relações sociais [Castells 2002].

As redes representam um espaço virtual atraente para os invasores explorarem vulnerabilidades técnicas. idades e falta de conhecimento e conscientização dos usuários sobre as ações de Engenharia Social [Al-Charchafchi et al. 2019]. O crescimento das redes sociais tem possibilitado a criação de um grande número de perfis falsos, com o uso de bots automatizados para suportar e dimensionar as atividades maliciosas.

Atacantes têm empregado bots para automatizar as etapas necessárias para estabelecer uma conexão de confiança com os usuários [Shafahi et al. 2016]. Essa construção de confiança envolve técnicas de manipulação psicológica, incentivando os usuários a interagir com ferramentas usadas pelos engenheiros sociais no ambiente digital. A combinação de táticas de manipulação psicológica com tecnologia avançada permite que os atacantes alcancem múltiplos alvos com surpreendente eficácia.

Bots são capazes de simular conversas humanas, sendo conhecidos como "ChatBots", e quando atuam nas redes sociais, são chamados de "SocialBots"[Shafahi et al. 2016]. Os ataques de Engenharia Social Automatizada requerem intervenção humana mínima, como um robô automatizado personificando outro humano para es-estabelecer uma conexão com as vítimas e pode atingir vários alvos simultaneamente devido à sua capacidade de escalabilidade [Mitnick and Simon 2003] [Huber et al. 2009].

Ataque de Engenharia Social Automatizada usando recursos como SocialBots e phishing são cada vez mais comuns, aproveitando o uso crescimento para atividades pessoais e profissionais. Os ataques de Engenharia Social requerem tempo e recursos para estabelecer uma relação de confiança. Os ataques de ES demandam tempo e recursos para estabelecer um relacionamento de confiança. No entanto, o desenvolvimento de uma interface homem-máquina permite que tais relacionamentos sejam automatizados.

7.2.3. Engenharia Social Automatizada e os Grande Modelos de Linguagem

O Processamento de Linguagem Natural (*Natural Language Processing* - NLP) tem recebido recentemente ampla atenção na cibersegurança, particularmente na automação cibernética. NLP é uma área da ciência da computação que permite que computadores interajam com a linguagem humana por meio do uso de software específico. Grandes Modelos de Linguagem (*Large Language Models* - LLMs) tornaram-se amplamente utilizados em aplicativos de NLP (e.g., ChatGPT e Google BERT), incluindo chatbots e assistentes virtuais. No entanto, com a utilização crescente destes modelos surge a necessidade de garantir a privacidade dos dados e a conformidade da segurança, especialmente quando estão envolvidas informações sensíveis.

Os usuários devem ter cautela ao enviar informações pessoais para o aplicações que utilizam LLMs, já que esses modelos podem ser treinados com dados que contêm informações sensíveis. Ao enviar uma pergunta, os usuários devem evitar incluir qualquer

informação que possa ser usada para identificá-los ou a outra pessoa, como, por exemplo, nomes, endereços, endereços de e-mail, etc.

Ferramentas genéricas de NLP não funcionam bem com linguagem específica de domínio, pois cada domínio possui características únicas que uma ferramenta genérica não está treinada para lidar. O domínio da cibersegurança apresenta uma variedade de dificuldades únicas, como a necessidade de compreender termos técnicos em constante evolução. Neste contexto, modelos de linguagem de cibersegurança têm sido criados, sendo os mesmos capaz de capturar conotações de texto em textos relacionados à cibersegurança. Um exemplo de tais modelos é o SecureBERT [Liberato 2022].

7.2.4. Prevenção e Mitigação para Ataques Focados nas Pessoas

Os atacantes investem em focar em pessoas dentro de uma organização e seus relacionamentos, a fim de lançar ataques que perpassam os mecanismos de cibersegurança tradicional. Tais atacantes exploram relacionamentos de confiança entre usuários internos e externos. Dessa forma, são necessárias estratégias para prevenção e mitigação de ataques focados nas pessoas. infelizmente, tais estratégias frequentemente falham em capturar o interesse das pessoas e são percebidas como uma tarefa secundária, um obstáculo ou uma distração de suas responsabilidades principais.

O treinamento de conscientização em cibersegurança pode incluir simulações de ataques, fornecendo aos usuários a oportunidade de vivenciar situações reais e aprender a identificar os sinais de manipulação. Ao aumentar a conscientização os usuários passam a ser defensores dos ativos de informação contra a ataques focados nas pessoas (e.g., Engenharia Social Automatizada), reduzindo o impacto desses ataques e fortalecendo a segurança dos sistemas de informações. Uma solução que tem se mostrado eficaz reside na implementação da gamificação, oferecendo uma alternativa envolvente e interativa às sessões de treinamento obrigatórias [Nijland 2022].

A comunicação e a consistência são fundamentais para facilitar uma cultura de segurança positiva[1]. Não apenas quando um incidente ocorreu, mas em qualquer situação em que seja necessário entender exatamente o que está acontecendo. Tal cultura dá aos usuários a confiança de que não apenas podem falar abertamente, mas que quaisquer ações ou decisões serão avaliadas de maneira justa. Isso faz com que se aumente o engajamento nos processos de cibersegurança, permitindo que os usuários concentrem no que é melhor para a organização, em vez de se preocuparem em se proteger.

7.3. Segurança na Era de Redes Programáveis

Os avanços recentes em Redes Definidas por Software (*Software Defined Networking*, SDN) expandiram nossa capacidade de programar a rede em direção ao plano de dados. Através de linguagens específicas de domínio como o P4, os operadores de rede podem rapidamente implementar novos protocolos em dispositivos de encaminhamento, personalizar suas funcionalidades e desenvolver serviços inovadores. Essa flexibilidade vem, no entanto, com um custo: as propriedades de segurança e de corretude em toda a rede (e.g., isolamento e acessibilidade) tornam-se muito mais difíceis de garantir, porque o

[1]A positive security culture - https://www.ncsc.gov.uk/collection/you-shape-security/a-positive-security-culture

comportamento da rede agora é determinado por uma combinação da configuração mantida pelo plano de controle e os programas do plano de dados que residem nos dispositivos de encaminhamento. Neste contexto, as ferramentas existentes para análise de segurança de redes, as quais dependem de um modelo fixo e invariante do plano de dados, são inadequadas para planos de dados programáveis.

Ao mesmo tempo, a capacidade de programar o plano de dados significa que é possível não apenas remodelar o comportamento da rede, mas também tornar a rede mais segura e confiável, melhorando sua confiabilidade, disponibilidade e integridade [Avizienis et al. 2004]. Isso pode ser feito por meio de um fluxo de serviços de segurança e confiabilidade, desenvolvidos a partir de blocos de construção provisionados diretamente nos dispositivos. Exemplos de blocos de construção incluem monitoramento e classificação de fluxo, bem como recursos de plano de dados de aplicação de políticas. Essa abordagem de provisionamento de serviços pode trazer várias vantagens exclusivas. Por exemplo, conformidade com a política pode ser garantida mesmo se o plano de controle e/ou um subconjunto de dispositivos de encaminhamento estiverem com defeito/comprometidos. Sendo assim, a medição da rede e a detecção de anomalias podem ocorrer de maneira verdadeiramente distribuída, com os dispositivos de encaminhamento de dados (*switches*) acionando prontamente ações de contramedidas, se for necessário.

Além dos requisitos de desempenho, as redes modernas podem ter políticas de segurança (explícitas ou implícitas) que definem o fluxo de informação entre *hosts*. Em uma rede *multi-tenant*, por exemplo, o operador pode querer garantir que os *tenants* estejam completamente isolados uns dos outros ou que um *tenant* não possa negar ao outro acesso à rede. Várias classes de propriedades foram consideradas pela comunidade de pesquisa: independentes de contexto (propriedades agnósticas de sessões de fluxo), dependente de contexto (referem-se aos fluxos de dados, por exemplo, iniciação da sessão), quantitativas (que são asseguradas com base em contadores, por exemplo, largura de banda garantida) e híbridas. À medida que os planos de controle e de dados se tornam mais complexos, torna-se mais difícil garantir que eles funcionem sempre corretamente. Para garantir que certas propriedades críticas sejam sempre satisfeitas, é vantajoso ter um mecanismo separado que seja apenas responsável por garantir que essas propriedades sejam respeitadas.

Esta seção visa fomentar discussão sobre a segurança de redes na era de planos de dados programáveis, ao apresentar (i) como o conceito de programabilidade do plano de dados pode ser usado para tornar as redes de computadores mais seguras, e (ii) quais os principais desafios de segurança que emergem juntamente com o conceito.

7.3.1. Modelagem e Análise de Políticas de Segurança

Uma maneira de expressar os requisitos que um sistema em rede deve atingir ou satisfazer é por meio de políticas de rede. A literatura é rica em soluções para especificação de políticas, verificação e aplicação. Boubata e Aib [Boutaba and Aib 2007], apresentam uma perspectiva histórica sobre a gestão de rede baseada em políticas. Os requisitos muitas vezes confiam em protocolos padrão para definir o que pode ser observado e executado (como endereços IP, portas TCP/UDP e outros campos de cabeçalho de protocolos padrão). A agenda de pesquisa de modelagem e análise de políticas para planos de dados programáveis deve se concentrar em três grandes questões: 1) como modelar e expressar

políticas, 2) como traduzir/refinar políticas e 3) como lidar com conflitos entre elas.

7.3.1.1. Propriedade baseadas em políticas específicas

As soluções baseadas em políticas para o plano de dados programável deve considerar classes de propriedades para expressar requisitos de nível superior/inferior que um sistema necessita satisfazer. A questão é, quais são essas classes e propriedades? Trabalhos anteriores consideraram isolamento, acessibilidade e equivalência em SDN, mas sem fornecer uma discussão conceitual de nível superior [Khurshid et al. 2013, Lopes et al. 2015].

Uma propriedade é dita independente de contexto se for agnóstica de fluxo de sessões, ou seja, pode ser definida por pacote, sem recorrer ao estado das informações. Exemplos incluem isolamento e conectividade. Por outro lado, uma propriedade é dita dependente do contexto se aborda o fluxo de pacotes fluxos dependendo de sua semântica na rede. Um exemplo é o início da sessão, que expressa em que direção as conexões podem ser iniciadas na rede (por exemplo, um *host* pode enviar uma consulta de resolução de nomes, mas não receber um). Neste caso, diz-se que algum *host* tem permissão para iniciar uma sessão com outro. Outra classe agrega propriedades quantificáveis. Os exemplos incluem largura de banda garantida, limite de largura de banda e k-redundância. A primeira expressa uma taxa mínima que um *host* tem garantido para enviar pacotes para outro. O segundo expressa uma taxa máxima permitida para o fluxo de informações entre esses *hosts*. A terceira propriedade, k-redundância (k interpretada como uma métrica de redundância), é definida para um determinado *link* lógico e especifica a existência de k outros *links* lógicos conectando o mesmo conjunto de *hosts*. Esta propriedade pode ser útil para expressar canais de *backup* e/ou melhorar a robustez contra Ataques de negação de serviço distribuído (DDoS).

Por fim, as propriedades híbridas apresentam aquelas com características de mais de uma das classes acima. Um exemplo é o *link* equivalência, que expressa que os *links* lógicos conectando quaisquer duas entidades têm o mesmo isolamento, conectividade, largura de banda, configurações, etc. Uma noção estendida da propriedade de equivalência é a redundância k-equivalente. Um *link* é dito k-equivalente redundante se houver k outros *links* conectando o mesmo conjunto de *hosts* e com propriedades equivalentes. A oportunidade de pesquisa envolve a proposta de linguagens políticas expressivas que apoiem o nível de especificação de políticas, e que simultaneamente se aproximem mutuamente de metas conflitantes. Por exemplo, essas linguagens devem ser agnósticas do formato do cabeçalho do pacote ou da semântica de análise, mas também permitem a expressão de políticas de uma maneira que corresponda ao atual comportamento do *switch*.

7.3.1.2. Tradução de políticas de nível superior para nível inferior

Como o *hardware* de rede é personalizado sob demanda e sua semântica de análise de pacotes muda com o tempo, as soluções de especificação de políticas de segurança precisam ter uma dinâmica de revisão e atualização [Udupi et al. 2007, Craven et al. 2011]. Essas políticas em um contexto de planos de dados programáveis despertam oportunida-

des de pesquisas. Por exemplo: 1) Como garantir a consistência entre políticas de nível superior e inferior [Verma 2002, Westerinen et al. 2001] à medida que o comportamento do *switch* muda?; 2) Como pode-se expressar políticas baseadas em propriedades genéricas de segurança e confiabilidade, de uma forma que as torne verificáveis e aplicáveis em qualquer configuração de plano de dados? Neste contexto, é importante definir quais classes de propriedade são de interesse, bem como entender as suas implicações no projeto de mecanismos de tradução de políticas de segurança.

Em uma rede definida por *software*, cabe ao controlador garantir que as políticas de nível superior sejam mantidas [Kreutz et al. 2013]. No entanto, à medida que os aplicativos do plano de controle e os programas de comutação do plano de dados evoluem de forma independente e se tornam mais complexos, torna-se mais difícil garantir a consistência das políticas intra e internível. Esse cenário dinâmico exige soluções que vão além da tradução de políticas e também verificam inconsistências. Um exemplo é uma política declarando que duas redes A e B devem ser isoladas (um cenário de *datacenter* multilocatário) e uma permitindo pacotes do *host* $a_i \in A$ para $b_j \in B$. Outro caso é uma política que expressa que dois *hosts* estão simultaneamente isolados e conectados.

Pesquisas anteriores consideraram casos como conflitos entre diferentes tipos de políticas de nível superior [Lupu and Sloman 1999] e análise de conflito baseada em regras [Hamed and Al-Shaer 2006]. No entanto, eles são limitados, pois consideram linguagens de especificação de políticas de nível mais alto ou são fortemente acoplados a protocolos de rede tradicionais. Sendo assim, a criação de soluções que possam garantir consistência de políticas de nível superior a inferior, considerando a especificação abstrata de programas de comutação, apresenta-se como uma avenida de pesquisa promissora a ser explorada pela comunidade de pesquisa.

7.3.2. Verificação de Políticas de Segurança

A imposição e a verificação são abordagens complementares que podem ser aplicadas como solução para garantir que políticas de segurança sejam respeitadas. Usando a imposição, o plano de dados pode ser monitorado durante a execução para buscar e bloquear ações que resultem em violações das políticas. A verificação (em conjunto com validação) se concentra em encontrar os *bugs* antes que os programas sejam implantados. Ela atua assegurando que o programa atenda às propriedades declaradas por seus requisitos.

Em um mundo onde os gerentes e operadores de rede podem redefinir o comportamento de dispositivos de encaminhamento, escrevendo seus próprios códigos para implementar alguma especificação de protocolo, a verificação e validação adequada (V&V) do código dos dispositivos torna-se crítica para o gerenciamento adequado das operações de rede e, portanto, a continuidade dos negócios. Em 2016, um roteador com defeito forçou a *Southwest Airlines* a cancelar 2.300 voos em quatro dias, resultando em uma perda de US$ 74 milhões [Carey 2017]. Alguns anos depois (julho de 2020), uma configuração de roteador defeituosa na *Cloudflare* causou uma interrupção de rede que durou apenas 27 minutos, mas levou a uma grande interrupção dos serviços de Internet em todo o mundo por mais de uma hora [Winder 2020]. A comunidade de redes tem pesquisado soluções para lidar com defeitos de *software* antes que eles causem tais danos. Abordagens como metadados sintáticos, execução simbólica, asserções e testes funcionais têm sido aplicadas

ao teste de *software* de plano de dados. Nesta seção são abordadas algumas das técnicas utilizadas para verificação e validação para *software* de plano de dados programáveis.

7.3.3. Imposição (*Enforcement*) de Políticas de Segurança

Uma alternativa à verificação é a imposição (enforcement). Em vez de verificar se uma configuração de rede está correta, um *kernel* de segurança logicamente separado evita ações que violem a política de segurança. O *kernel* de segurança deve mediar todas as ações de manipulação de pacotes no plano de dados. Ao contrário do modo de verificação, onde verifica-se as violações da política antes de uma configuração ser enviada para a rede, no modo de imposição, verifica-se as violações da política, uma vez que estão prestes a ocorrer. Tanto a verificação como a imposição têm suas vantagens e desvantagens. Por um lado, a verificação capta problemas precocemente; um verificador pode fornecer informações de diagnóstico detalhadas sobre por que uma configuração viola uma política durante a fase de verificação. No regime de imposição, os problemas são detectados à medida que ocorrem. A imposição pode ser mais atrativa do que a verificação, porque não depende da complexidade do programa de controle ou do plano de dados.

Sendo assim, emergem benefícios para imposição (*enforcement* da política de segurança em plano de dados programáveis. Os planos permitem que os operadores de rede modifiquem o *pipeline* de processamento de pacotes dos dispositivos de rede para implementar novos protocolos, personalizar o comportamento da rede e estabelecer serviços de rede avançados. No que pese a sua simplicidade da programação, os programas P4 demonstraram ser propensos a uma variedade de *bugs* e erros de configuração [Stoenescu et al. 2016, Freire et al. 2018]. Como resultado, os operadores de rede precisam de estruturas para garantir que os programas que produzem tenham um comportamento correto para obter os benefícios de um ecossistema de *software* de plano de dados. Ferramentas de verificação de rede de última geração podem obter um modelo da rede, sua configuração e um conjunto de propriedades específicas usando formalismos tradicionais (por exemplo, lógica temporal ou regras de *Datalog*) e verificar automaticamente se essas propriedades são válidas para qualquer pacote [Beckett et al. 2017, Lopes et al. 2015].

Embora essas ferramentas ajudem os operadores de rede a identificar *bugs* antes que eles se manifestem, deve-se considerar: (i) Primeiro que a maioria dessas ferramentas exige que os programadores modelem manualmente os planos de dados programáveis, atividade complexa e propensa a erros [Lopes et al. 2015]; (ii) Em segundo lugar, essas ferramentas são geralmente restritas em termos de propriedades de acessibilidade para reduzir os tempos de verificação [Lopes et al. 2016]; (iii) Terceiro, ferramentas mais expressivas capazes de verificar múltiplas propriedades frequentemente enfrentam problemas graves de escalabilidade (por exemplo, verificar a conformidade com uma especificação de protocolo pode levar dias, mesmo para um único plano de dados programáveis; e (iv) Por fim, os programadores precisam ter habilidades técnicas formais de verificação para especificar corretamente suas propriedades.

Neves et al. [Neves et al. 2021] apresentam uma nova abordagem baseada na aplicação dinâmica (ou em tempo de execução) em vez de verificação estática. Essa abordagem tem várias vantagens práticas. Já que não é necessário esperar pelo resultado de um longo processo de verificação para enviar uma nova configuração para os *switches*

de rede. Sendo assim, a aplicação do tempo de execução pode intervir prontamente se situações problemáticas realmente ocorrerem, possibilitando: obter informações úteis do código com *bugs* quando ele tem um comportamento correto e reparar problemas sem interferir em qualquer serviço de rede.

Em contraposição com a verificação estática, a aplicação do tempo de execução também permite ao programador expressar a política e o mecanismo usando o mesmo ambiente de programação que o resto do programa. Esse valor deve ser considerado, não só porque facilita a vida do programador, como evita também erros de tradução entre a implementação e as políticas. Sendo assim, para perceber os benefícios de uma aplicação dinâmica, Neves et al. [Neves et al. 2021] desenvolveram o P4box, um sistema para implantação de monitores de tempo de execução em planos de dados programáveis.

Usando P4box os programadores podem anexar monitores antes e depois dos blocos de controle, transições de estado do analisador e chamadas para funções externas de um programa P4. Cada monitor pode modificar a entrada e saída do bloco de código ou função que monitora, permitindo a verificação de pré e pós-condições a serem utilizadas para impor propriedades específicas ou modificar o comportamento do bloco monitorado.

Um monitor de tempo de execução insere-se na interação de um bloco de controle P4 ou analisador com o restante do ambiente de execução, permitindo que o programador do monitor modifique o comportamento do bloco P4 incluso com o restante do ambiente. Um bloco programável P4 faz a *interface* com o restante do ambiente de execução P4 na entrada no bloco, retornar do bloco as chamadas para funções externas fornecidas pela arquitetura. Na programação do modelo P4box, quando um bloco programável é invocado, o controle passa primeiro para um monitor, também escrito em P4, antes de passar para o bloco programável pretendido. Da mesma forma, quando um bloco programável completa o processamento, o controle passa primeiro para o monitor antes de retornar ao dispositivo, permitindo que um monitor modifique o comportamento de blocos programáveis de maneira bem definida.

7.3.4. Explorando Planos de Dados Programáveis para Detectar Ataques DDoS

Ataques de negação de serviço distribuído (DDoS) fazem uso dos limites de capacidade específicos aplicados a todos os recursos da rede. Esses ataques dependem de *botnet* para esgotar recursos computacionais e interromper aplicações na Internet [Hoque et al. 2015]. Buscam encaminhar um grande número de solicitações para o recurso tecnológico invadido, visando exceder a sua capacidade, interrompendo o seu funcionamento.

À medida que os *botnets* aumentam a sua aplicabilidade para explorar os dispositivos IoT (Internet das Coisas) vulneráveis, a frequência, a capacidade e o volume dos ataques DDoS amplia o seu alcance drasticamente. A detecção dessa ameaça é o primeiro passo para minimizar as perdas por meio do desencadeamento das medidas defensivas, no entanto, representa um desafio para a pesquisa em rede [Antonakakis et al. 2017, Anstee et al. 2017, Zargar et al. 2013].

Preferencialmente, a detecção e o bloqueio de ataques DDoS devem ocorrer nas fontes para economizar esforços de deslocamento e processamento sobre o tráfego indesejado [Gil and Poletto 2001, Mirkovic et al. 2002, Peng et al. 2004]. No entanto, isso

é impedido pela disseminação da atividade maliciosa, que é construída a partir da sincronização de solicitações aparentemente legítimas. Além disso, essas fontes normalmente pertencem a diferentes domínios administrativos, nos quais as políticas de segurança são definidas de forma independente. Mais adiante, nas proximidades da vítima, apesar do tráfego de ataque ser mais proeminente para detecção [Kim et al. 2006, Hoque et al. 2015], ele pode já ter saturado recursos *in-path*. A alternativa é implantar medidas defensivas em Provedores de Serviços de Internet (ISPs), que gerenciam a comunicação [Haq et al. 2015, Kang et al. 2016]. Os ISPs se beneficiam de uma visão privilegiada do tráfego e contam com *links* de alta taxa de transferência, permitindo que eles descubram e impeçam as ameaças em tempo hábil.

Ao contrário dos *datacenters*, onde o monitoramento de rede sofisticado pode ser realizado em *hosts* finais [Moshref et al. 2016, Yu et al. 2011], os ISPs dependem de *switch primitive* como amostragem de pacotes [CiscoNetworks 2017, Sflow 2017] e contagem baseada em fluxo [McKeown et al. 2008]. Os dados resultantes são então normalmente montados em servidores fora de banda para inspeção. Enquanto essas primitivas apresentam compensações entre granularidade de visibilidade, utilização de largura de banda, espaço de memória e a comunicação com servidores externos incorre em uma latência adicional para detectar eventos de rede [Moshref et al. 2013]. A fim de manter a utilização razoável da largura de banda e a carga de processamento, a amostragem de pacotes é geralmente empregada em taxas agressivamente baixas [Phaal 2009], apenas transmitindo informações de um conjunto limitado de pacotes. Diferentemente, a contagem baseada em fluxo, como em *switches* OF [McKeown et al. 2008], fornece valores exatos para métricas volumétricas com um alto custo de entradas nas tabelas.

Como alternativa promissora para este problema, o conceito emergente de programabilidade do plano de dados oferece flexibilidade para a execução de algoritmos nos *switches* de rede [Bosshart et al. 2014]. Assumindo um fluxo de pacotes como entrada, esses algoritmos são modelados como um *pipeline* de primitivas elementares, acessos à memória e pesquisas em tabelas. Sendo assim, os operadores podem definir funções de monitoramento e delegá-las a dispositivos de plano de dados em toda rede. Essa arquitetura pode ser explorada para realizar a inspeção em cada pacote sem incorrer em sobrecarga de comunicação. No entanto, buscando executar a taxa de linha com custos razoáveis, o processamento de pacotes é restrito a um pequeno orçamento de tempo e uma quantidade limitada de memória por estágio de *pipeline* [Bosshart et al. 2013].

Lapolli et. al [Lapolli et al. 2019] desenvolveram uma arquitetura de sistema para detecção de DDoS, na qual o plano de dados responsável pela coleta do fluxo das métricas e sua inspeção. Isso é apresentado na forma de uma detecção de ataque DDoS em banda sistema totalmente implementável em uma chave programável através de P4. O trabalho compreende um *pipeline* de processamento para estimar as entropias dos endereços IP de origem e destino. Esses valores são usados para caracterizar o tráfego supostamente legítimo em tempo real. Os resultados desta caracterização servem para calcular a detecção limiares considerando um coeficiente de sensibilidade parametrizável. A fim de respeitar o rigoroso orçamento de tempo e restrições de memória para o cálculo da entropia, a frequência de endereços IP distintos é aproximada por esboços de contagem aprimorados [Charikar et al. 2002]. Outras funções aritméticas de computação intensiva são resolvidas com a ajuda de uma tabela de pesquisa *Longest Match Routing Rule* (LPM) otimizada

para memória.

7.3.5. Depuração e Rastreabilidade de Aplicações em Planos de Dados Programáveis

Planos de dados programáveis permitem que a execução de aplicações cruzem a fronteira entre servidores x86 tradicionais e a rede de computadores, habilitando o descarregamento (ou seja, *offloading*) de partes da computação para PDPs. Esse paradigma tem sido chamado de *in-network computing* [Benson 2019]. À luz desse desenvolvimento, tanto a indústria quanto pesquisadores começaram a investigar ativamente novos projetos para aplicações distribuídas a fim de melhorar o desempenho, a escalabilidade ou a confiabilidade dessas, transferindo parte de sua funcionalidade para a rede. Dessa forma, uma vasta gama de problemas tem explorado essa possibilidade de descarregar parte da computação para a rede: *Caching*: NetCache [Jin et al. 2017] armazena em cache pares de chave-valor em switches, evitando potencialmente longos RTTs para acessar um servidor de armazenamento de chave-valor remoto; *Agregação de Dados*: DAIET [Sapio et al. 2017] realiza agregação de dados na rede para maior escalabilidade; *Machine Learning*: machine learning dentro de switches pode mitigar gargalos existentes durante o treinamento distribuído de modelos [Sanvito et al. 2018, Xiong and Zilberman 2019a]; *Pattern Matching*: a correspondência de padrões eficiente pode ser alcançada através da realização de parte da computação na rede [Jepsen et al. 2019].

À medida que essas abordagens recém-descobertas se aproximam da implantação, surgem preocupações práticas sobre seu gerenciamento em tempo de execução, porque as aplicações distribuídas agora podem executar parcialmente no plano de dados. Especificamente, a incorporação de lógica em PDPs adicionou outra camada de complexidade para rastrear e solucionar problemas dessas aplicações, e esforços tradicionais de rastreabilidade e observabilidade de aplicações em servidores x86 tradicionais não se traduzem diretamente para *in-network computing* [Benson 2019]. Em particular, switches programáveis atuais não fornecem uma abstração rica o suficiente para suportar técnicas de rastreamento tradicionais [Sigelman et al. 2010, Mace and Fonseca 2018, Chow et al. 2014], e essa falta de primitivas de rastreamento força os programadores a criarem suas próprias soluções exclusivas. Isso leva à criação de ferramentas de rastreamento muito específicas e não reutilizáveis para depurar a computação na rede. Mais importante, rastros produzidos por soluções específicas para o PDP provavelmente não serão interoperáveis com estruturas de diagnóstico de rastreamento existentes, por exemplo, Dapper[Sigelman et al. 2010] do Google. Ortogonalmente, as estruturas de rastreamento existentes não fornecem primitivas para gerar ou capturar dados de rastreamento em planos de dados programáveis.

Um desafio de pesquisa atual visa preencher a lacuna entre técnicas tradicionais para telemetria de redes e *frameworks* de rastreamento distribuído. Isso requer abordar execuções que cruzem a fronteira da aplicação distribuída para o plano de dados programável, capturando dados de rastreamento de PDPs e apresentando-os ao plano de aplicação por meio de uma abstração flexível e bem definida. Um dos primeiros esforços nessa direção é o P4-Intel [Castanheira et al. 2019], que (i) aproveita a telemetria de rede para instrumentar PDPs no monitoramento de dados de rastreamento arbitrários definidos pelo usuário e (ii) coordena o armazenamento, coleta e formatação desses dados de rastreamento internamente, fornecendo apenas dados de contexto bem formados para qualquer

ferramenta de depuração do plano de aplicação.

7.4. Inteligência Artificial e Segurança

Para "o bem e para o mal", a área de Inteligência Artificial (IA) vem impactando substancialmente a Segurança Cibernética. Por um lado, *deepfakes* tornam mais fáceis golpes virtuais. Por outro, IA tem potencial para melhorar os processos de segurança, por exemplo, via identificação automatizada de fraudes e ações suspeitas. Além de como usar IA para melhorar a segurança, impõe-se como questão central considerar aspectos-chave como ética, transparência, responsabilidade, explicabilidade e confiabilidade. Considerando-se a experiência do Grupo de Redes de Computadores nesse grande tema, a seguir, aborda-se um específico: a oportunidade de se capitalizar redes programáveis como base para o projeto e o desenvolvimento de mecanismos *in-network* inteligentes voltados à proteção de redes e serviços.

Um marco significativo na evolução de Redes Definidas por Software (SDN) foi o desenvolvimento do OpenFlow como uma implementação real de SDN. No entanto, a operação da rede ainda está limitada ao conjunto de protocolos e cabeçalhos suportados pelo hardware dos dispositivos de encaminhamento (ex: switches e interfaces de rede). Assim, a definição de funções personalizadas para o processamento de pacotes torna-se muito difícil. Recentemente, o conceito de Planos de Dados Programáveis (PDPs) surgiu para superar essas limitações. Os PDPs permitem o controle completo do comportamento da rede, desde as aplicações até o processamento de pacotes dentro dos dispositivos, incluindo a definição e a análise de cabeçalhos personalizados. Tal proporciona uma oportunidade sem precedentes para desenvolver novos recursos nos dispositivos de encaminhamento e revisitar funções existentes para o gerenciamento de redes [Cordeiro et al. 2017]. Atualmente, a linguagem P4 é o padrão de fato para descrever como os pacotes de rede devem ser processados.

Uma das áreas que pode se beneficiar de PDPs/P4 é a de Segurança de Redes. Sistemas de Detecção de Intrusão (IDSs) podem ser aprimorados implementando-os como funções eficientes implantadas no plano de dados, capazes de reagir rapidamente a anomalias de rede que possam representar ameaças. IDSs geralmente dependem da coleta de características de tráfego (*traffic features*), que são posteriormente alimentadas em sistemas sofisticados baseados principalmente em algoritmos de Aprendizado de Máquina (*Machine Learning* – ML). ML tem sido usada com sucesso em segurança de redes devido à sua capacidade de detectar e descobrir padrões e comportamentos não observados anteriormente no tráfego. A maioria das abordagens de segurança desenvolvidas no contexto de SDN e baseadas em ML foram implementadas exclusivamente no plano de controle, apesar dos problemas associados à precisão e à sobrecarga significativa que podem introduzir [Xie et al. 2019].

As funcionalidades introduzidas pelos PDPs tornam possível considerar um novo cenário para soluções de segurança baseadas em ML, aproveitando as capacidades de transferência (*offloading*) de parte dos algoritmos para os dispositivos de encaminhamento. Assim, soluções mais precisas e responsivas podem ser implantadas. A decisão sobre quanto das funções deve ser transferido para o plano de dados não é trivial [Ports and Nelson 2019], pois as capacidades de computação dos dispositivos de rede são limi-

tadas, e o *offloading* excessivo de funcionalidades pode prejudicar a vazão máxima no encaminhamento de pacotes.

A seguir, discute-se alguns desafios enfrentados na interseção entre Planos de Dados Programáveis e IA/ML para detecção de intrusão. Foca-se em como aproveitar as funcionalidades dos PDPs na implementação de IA/ML, especialmente algoritmos de ML. Aborda-se, principalmente, a questão de quanto das operações dos algoritmos é viável ser transferida para dispositivos de encaminhamento. A reflexão tem como base o esforço realizado por Gutiérrez *et al.* [Gutiérrez et al. 2021].

7.4.1. Aprendizado de Máquina em Planos de Dados Programáveis para melhorar a Detecção de Intrusão

O Aprendizado de Máquina tornou-se um marco essencial em vários tipos de soluções de segurança cibernética devido à sua capacidade de extrair anomalias e padrões que podem ser sintomas de ataques internos ou externos contra a infraestrutura. Essas soluções são geralmente integradas por componentes de segurança de rede e *host*, incluindo *firewalls*, antivírus e IDSs [Le and Zincir-Heywood 2020].

Os IDSs estão sendo revisitados para melhor aproveitar as possibilidades habilitadas pelo novo contexto de redes programáveis. A maioria das soluções desenvolvidas para a implementação de IDSs foi implantada como aplicações em execução no plano de controle. No entanto, essa abordagem para a implementação de IDSs tem duas principais desvantagens. Primeiro, o conjunto de características de tráfego derivadas de contadores padrões (por exemplo, aqueles disponíveis nas versões atuais do OpenFlow) ou, em situações extremas, via eventos `PACKET_IN`, é insuficiente para obter precisão razoável nos algoritmos de ML. Segundo, os algoritmos de ML são, normalmente, intensivos em computação. Se não forem adequadamente projetados e implantados, podem introduzir sobrecarga no plano de controle e prejudicar o funcionamento correto da rede [Binbusayyis and Vaiyapuri 2019].

Como introduziu-se anteriormente, o surgimento dos PDPs torna possível o *offloading* de algumas funcionalidades para o plano de dados. A seguir, apresenta-se uma visão geral das etapas de um IDS baseado em ML e delineia-se como os PDPs podem ser aproveitados para melhorar algumas dessas etapas por meio do processamento personalizado de pacotes e do *offloading* de operações específicas [Le and Zincir-Heywood 2020]. A Figura 7.1 apresenta a visão sequencial e as relações entre essas etapas.

Coleta de Dados. Os PDPs estendem as possibilidades de coleta de dados além das estatísticas padrões disponíveis nos dispositivos de encaminhamento. Estatísticas personalizadas podem ser introduzidas, e algum processamento com estado (*stateful*) pode ser incluído nos dispositivos, o que pode se traduzir em indicadores de grande valor para tarefas de detecção de intrusão [Kohler et al. 2018]. Apesar das limitações inerentes ao poder de computação e às primitivas de programação disponíveis em dispositivos de encaminhamento programáveis, duas funcionalidades podem ser aproveitadas para implementar a coleta de dados eficiente para algoritmos de ML: análise personalizada de pacotes e agregação de dados. A análise personalizada de pacotes permite o processamento de cabeçalhos que podem ser usados para calcular estatísticas personalizadas, por

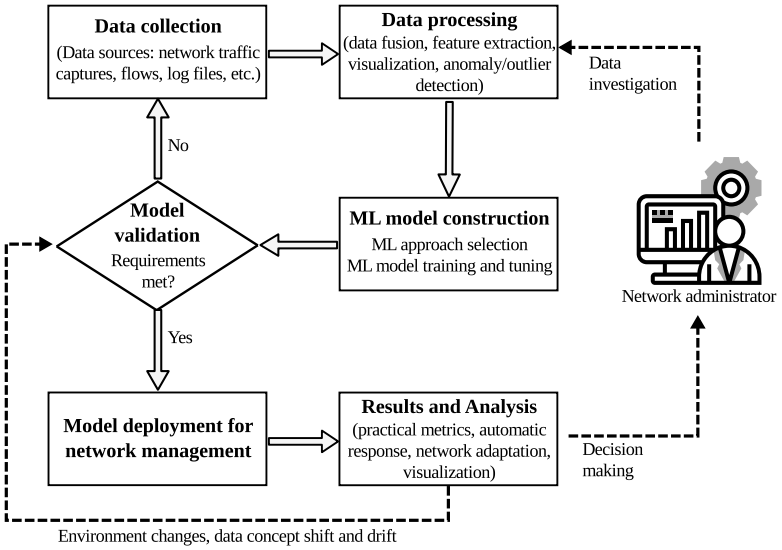


Figura 7.1: Estágios de um sistema baseado em Aprendizado de Máquina aplicado ao domínio de Gerenciamento de Redes (extraído de [Gutiérrez et al. 2021] *apud* [Le and Zincir-Heywood 2020].

exemplo baseadas em campos de novos protocolos [Gupta et al. 2018]. Já a agregação ajuda a reduzir a quantidade de dados que precisa ser transmitida dos dispositivos para o Plano de Controle para a execução de operações complexas [Sapio et al. 2021].

Processamento de Dados. Propostas recentes introduzem o conceito de processamento na rede como um serviço, onde a implementação de operações no plano de dados fica disponível para uso por estruturas de alto nível de ML ou análise de dados [Mustard et al. 2019, Sapio et al. 2017]. Outras propostas introduzem a noção de consultas que acionam a coleta e a análise preliminar de dados para produzir estatísticas que podem ser posteriormente entregues a mecanismos de processamento de alto nível [Gupta et al. 2018]. A ideia central por trás dessas abordagens é que recursos dos PDPs permitem não apenas medições e contagens, mas também a realização de análises em paralelo com a coleta de dados.

Construção do Modelo. A literatura apresenta várias abordagens para aproveitar as funcionalidades disponíveis nos PDPs para implementar diferentes algoritmos, considerando as restrições computacionais dos dispositivos de encaminhamento programáveis [Ports and Nelson 2019]. Essas propostas incluem o uso de registradores de switches (para implementar aritmética e armazenamento de valores), tabelas de correspondência (*match-action*), entre outros construtos, para a implementação de técnicas como Árvores de Decisão, Máquinas de Vetores de Suporte (SVM), classificadores Naive Bayes e Redes Neurais [Qin et al. 2020]. Essa abordagem reduz a quantidade de informações que precisam ser encaminhadas para o plano de controle (por exemplo, eventos `PACKET_IN`), o que contribui para diminuir a sobrecarga do canal de controle [Macías et al. 2020], ao mesmo tempo que aumenta a precisão e a capacidade de resposta [Ports and Nelson

2019, Xiong and Zilberman 2019b]. Além da implementação direta nos dispositivos, outra abordagem a ser seguida é a cooperação na formação de modelos em grande escala por meio da análise de métricas locais. Essa abordagem é chamada de Aprendizado Federado e pode ser usada para treinar modelos complexos, como Redes Neurais Profundas [Qin et al. 2020].

Validação do Modelo. A validação é uma tarefa de alto nível que envolve análise extensa e *feedback* de especialistas humanos. Portanto, os PDPs não têm intervenção direta nas tarefas associadas a essa etapa. No entanto, funcionalidades como Processamento de Eventos Complexos [Kohler et al. 2018] e telemetria baseada em consultas [Gupta et al. 2018] são úteis para fornecer *insights* para depurar situações de baixa precisão e baixo desempenho dos algoritmos de ML.

Implantação. Esta operação deve considerar as particularidades envolvidas no desenvolvimento dos algoritmos. Por exemplo, a disponibilidade, em uma determinada arquitetura de hardware, do tipo de tabelas necessárias ou o número de registradores que podem ser usados para armazenar o estado dos pacotes são aspectos que devem ser validados [Qin et al. 2020]. Para uma discussão detalhada dos problemas associados à implantação de algoritmos de ML em dispositivos de encaminhamento programáveis, consulte [Xiong and Zilberman 2019b].

Análise de Resultados. Funcionalidades como Telemetria de Rede em Banda, que dependem de recursos dos PDPs [Gupta et al. 2018], e Processamento de Eventos Complexos [Kohler et al. 2018] podem fornecer *insights* importantes para essa etapa. Além disso, a definição tanto de limiares para *features* específicas quanto de intervalos de tempo adequados para análise contribuem para avaliar a eficácia dos algoritmos, permitindo algum grau de análise dos dados.

7.4.1.1. BUNGEE-ML: Um Estudo de Caso

BUNGEE-ML é um sistema que combina o processamento rápido do plano de dados e a alta capacidade e inteligência do plano de controle para detecção precisa e mitigação de ataques na rede. Avanço mais recente de toda uma linha de trabalhos [Lapolli et al. 2019, Ilha et al. 2021, González et al. 2021], o sistema implementa uma estratégia de vários níveis [Marnerides et al. 2011] para garantir a operação contínua da rede, promovendo a cooperação vertical e horizontal entre os elementos da rede (Fig. 7.2):

- *Cooperação vertical*: para contornar as limitações de processamento do ASIC (*Application-Specific Integrated Circuit*) dos dispositivos de encaminhamento programáveis, BUNGEE-ML realiza uma análise de tráfego mais sofisticada fora do ASIC, uma abordagem *vertical*, que depende dos recursos da CPU do switch e do controlador SDN. Essa análise *profunda* prioriza a precisão e pode corrigir decisões tomadas no ASIC.

- *Cooperação horizontal*: Aproveitando a topologia programável, BUNGEE-ML "empurra" oportunisticamente o tráfego malicioso o mais longe possível da vítima, uma estratégia de mitigação em *largura* no plano de dados, que permite respostas rápidas a ataques DDoS.

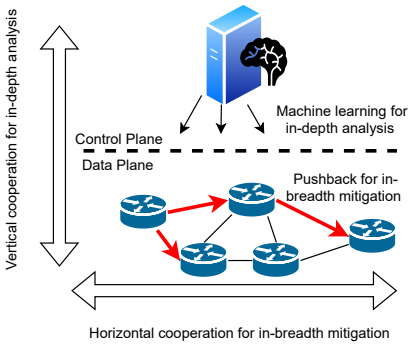


Figura 7.2: Cooperação vertical e horizontal (extraída de [González et al. 2023]).

Essencialmente, BUNGEE-ML permite que ambos os planos cooperem, explorando suas forças individuais. Primeiro, o ASIC do switch executa estratégias leves que possibilitam a detecção precoce de tráfego suspeito na taxa de linha. Isso é combinado com processamento ligeiramente mais sofisticado para comparar estatísticas de tráfego recentes na CPU do switch. No entanto, para realizar uma análise mais profunda e sofisticada das fontes suspeitas, o plano de controle aplica técnicas de Aprendizado de Máquina para decidir se os suspeitos (identificados pelo plano de dados) são atacantes.

Embora as ações de mitigação possam ser implantadas assim que o switch tenha marcado um fluxo como suspeito – por exemplo, o plano de dados pode reduzir seletivamente (*throttling*) o tráfego das fontes suspeitas para lidar rapidamente com o ataque, as contramedidas tornam-se permanentes após o plano de controle confirmar os fluxos de ataque. Nesse caso, o plano de dados implementa uma estratégia de recuo nos suspeitos confirmados, incentivando dispositivos *upstream* a construir uma frente de mitigação colaborativa e parar o ataque o mais longe possível da vítima.

A Fig. 7.3 ilustra o fluxo geral do BUNGEE-ML, mostrando as interações entre seus componentes nos planos de controle e dados:

- A etapa de monitoramento de fluxos (❶) é a base da implementação. O sistema realiza monitoramento contínuo e executa uma estratégia com base na análise de entropia dos pacotes de entrada usando os ASICs nos dispositivos do plano de dados para detectar mudanças no comportamento da rede durante uma "janela de monitoramento".

- No caso de um ataque, o monitoramento de fluxos aciona um alerta para o componente de *Window Inspection* (❷). Nesse componente, os endereços de origem mais recentes da janela de monitoramento são comparados com estatísticas globais para identificar os suspeitos que estão causando a perturbação na rede.

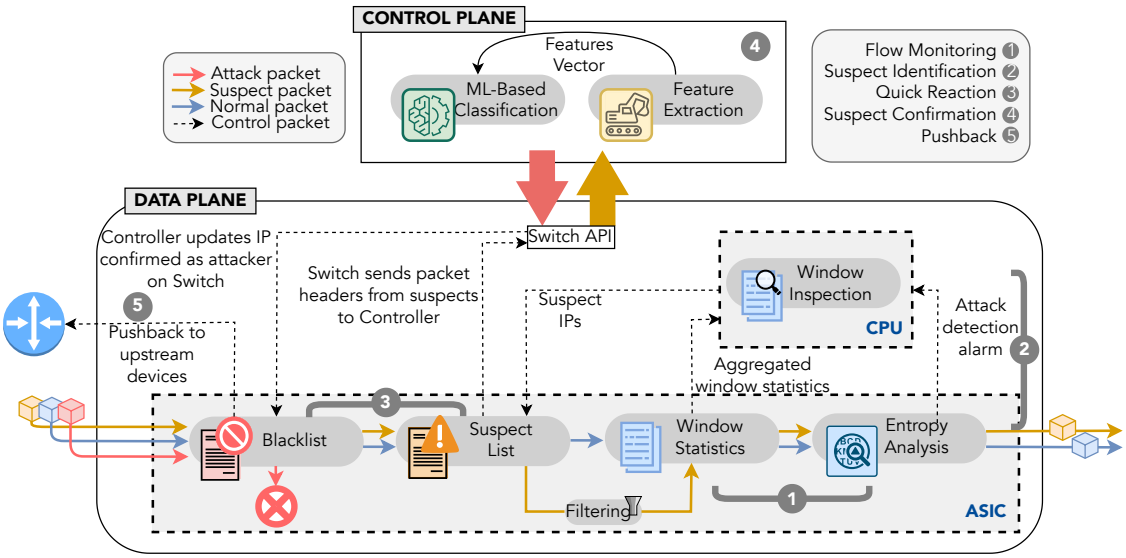


Figura 7.3: Visão geral do BUNGEE-ML (extraída de [González et al. 2023]).

- Uma reação rápida começa depois que as fontes suspeitas são identificadas (❸). Isso inclui a manutenção de uma *Lista de Suspeitos* para que pacotes subsequentes desses suspeitos sejam filtrados. Pacotes suspeitos de entrada são desviados para o plano de controle, onde uma inspeção adicional é realizada para determinar se as fontes são atacantes ou não.

- O plano de controle extrai características dos pacotes para classificar os endereços de origem usando mecanismos de Aprendizado de Máquina (❹). Após a classificação de um endereço, o plano de controle notifica o plano de dados para *(a)* remover os endereços classificados como benignos da *Lista de Suspeitos* ou *(b)* confirmar os endereços classificados como maliciosos, ou seja, incluí-los em uma *Lista Negra*. Pacotes de entrada de fontes incluídas na *Lista Negra* são descartados.

- Por fim, o switch alerta os dispositivos "upstream" sobre o ataque em andamento (❺) enviando a lista de suspeitos confirmados para tomar as medidas de mitigação apropriadas, que se denomina como ação de recuo.

As etapas ❶, ❷, ❸, e ❺ são todas executadas no plano de dados para detectar e mitigar um ataque. Enquanto isso, o plano de controle aprimora a lista de suspeitos formada pelo plano de dados (❹) para melhorar a precisão da classificação e mitigação.

7.5. Considerações Finais

Violações de cibersegurança custam trilhões anualmente às organizações. Assim, são necessários mecanismos que assegurem a proteção dos ativos das ameaças a sua integridade, disponibilidade e confidencialidade. As organizações tradicionalmente implementam esses mecanismos contra acessos não autorizados, alterações indevidas ou sua indisponibilidade. No entanto, os avanços em diversas áreas da computação necessitam ser acompanhados de avanços em cibersegurança.

O presente capítulo discute algumas tendências para cibersegurança nos próximos

anos. Inicialmente, a Segurança Centrada nas Pessoas foi discutida, incluindo Engenharia Social e as evoluções que tem acompanhado a utilização de fatores humanos a cibersegurança. Em seguida, o uso de Inteligência Artificial e Segurança foi relatado, considerando mecanismos de detecção de intrusão e mitigação de ataques. Finalmente, A Segurança na era dos planos de dados programáveis foi apresentada, explorando a relação entre a mecanismos de segurança e a programabilidade do plano de dados.

Apesar da discussão apresentada no capítulo, novos tópicos podem ser trazidos em trabalhos futuros. A ampliação de funcionalidade de computadores quânticos implica em riscos para diversos mecanismos de criptografia usados atualmente. Dessa forma, é necessário o desenvolvimento e a implementação de algoritmos e protocolos de Criptografia Pós-Quântica. Finalmente, a compreensão dos desafios éticos em Computação é fundamental para assegurar uma ambiente digital seguro e protegido. Assim, repercussões filosóficas e sociais precisam ser integrados às discussões técnicas.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Referências

[Al-Charchafchi et al. 2019] Al-Charchafchi, A., Manickam, S., and Alqattan, Z. N. (2019). Threats against information privacy and security in social networks: A review. In *International Conference on Advances in Cyber Security*, pages 358–372. Springer.

[Anstee et al. 2017] Anstee, D., Bussiere, D., Sockrider, G., and Morales, C. (2017). Worldwide infrastructure security report. *Arbor Networks Inc., Westford, MA, USA*.

[Antonakakis et al. 2017] Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M., et al. (2017). Understanding the mirai botnet. In *26th USENIX security symposium (USENIX Security 17)*, pages 1093–1110.

[Avizienis et al. 2004] Avizienis, A., Laprie, J.-C., Randell, B., and Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing*, 1(1):11–33.

[Beal 2005] Beal, A. (2005). Segurança da informação: Princípios e melhores práticas para a proteção dos ativos de informação nas organizações. *Atlas*.

[Beckett et al. 2017] Beckett, R., Gupta, A., Mahajan, R., and Walker, D. (2017). A general approach to network configuration verification. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 155–168.

[Benson 2019] Benson, T. A. (2019). In-network compute: Considered armed and dangerous. In *Proceedings of the Workshop on Hot Topics in Operating Systems*, HotOS '19, pages 216–224, New York, NY, USA. ACM.

[Binbusayyis and Vaiyapuri 2019] Binbusayyis, A. and Vaiyapuri, T. (2019). Identifying and benchmarking key features for cyber intrusion detection: An ensemble approach. *IEEE Access*, 7:106495–106513.

[Bosshart et al. 2014] Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., et al. (2014). P4: Programming protocol-independent packet processors. 44 (3): 87–95, july 2014.

[Bosshart et al. 2013] Bosshart, P., Gibb, G., Kim, H.-S., Varghese, G., McKeown, N., Izzard, M., Mujica, F., and Horowitz, M. (2013). Forwarding metamorphosis: Fast programmable match-action processing in hardware for sdn. *ACM SIGCOMM Computer Communication Review*, 43(4):99–110.

[Boutaba and Aib 2007] Boutaba, R. and Aib, I. (2007). Policy-based management: A historical perspective. *Journal of Network and Systems Management*, 15(4):447–480.

[Carey 2017] Carey, S. (2017). Why a single failed router can ground a thousand flights. *The Wall Street Journal*.

[Castanheira et al. 2019] Castanheira, L., Schaeffer-Filho, A., and Benson, T. A. (2019). P4-intel: Bridging the gap between icf diagnosis and functionality. In *Proceedings of the 1st ACM CoNEXT Workshop on Emerging In-Network Computing Paradigms*, ENCP '19, page 21–26, New York, NY, USA. Association for Computing Machinery.

[Castells 2002] Castells, M. (2002). *A sociedade em rede*. Editora Paz e Terra.

[Charikar et al. 2002] Charikar, M., Chen, K., and Farach-Colton, M. (2002). Finding frequent items in data streams. In *International Colloquium on Automata, Languages, and Programming*, pages 693–703. Springer.

[Chow et al. 2014] Chow, M., Meisner, D., Flinn, J., Peek, D., and Wenisch, T. F. (2014). The mystery machine: End-to-end performance analysis of large-scale internet services. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 217–231, Broomfield, CO. USENIX Association.

[CiscoNetworks 2017] CiscoNetworks (2017). Cisco ios netflow. In *Accessed on June, 29 2017*.

[Cordeiro et al. 2017] Cordeiro, W., Marques, J., and Gaspary, L. (2017). Data plane programmability beyond openflow: Opportunities and challenges for network and service operations and management. *J. Netw. Syst. Manage.*, 25(4):784–818.

[Craven et al. 2011] Craven, R., Lobo, J., Lupu, E., Russo, A., and Sloman, M. (2011). Policy refinement: Decomposition and operationalization for dynamic domains. In *2011 7th International Conference on Network and Service Management*, pages 1–9. IEEE.

[Freire et al. 2018] Freire, L., Neves, M., Leal, L., Levchenko, K., Schaeffer-Filho, A., and Barcellos, M. (2018). Uncovering bugs in p4 programs with assertion-based verification. In *SOSR*, page 4. ACM.

[Gil and Poletto 2001] Gil, T. M. and Poletto, M. (2001). {MULTOPS}: A {Data-Structure} for bandwidth attack detection. In *10th USENIX Security Symposium (USENIX Security 01)*.

[González et al. 2023] González, L. A. Q., Castanheira, L., Marques, J. A., Schaeffer-Filho, A. E., and Gaspary, L. P. (2023). Bungee-ml: A cross-plane approach for a collaborative defense against ddos attacks. *J. Netw. Syst. Manag.*, 31(4):77.

[González et al. 2021] González, L. A. Q., Castanheira, L., Marques, J. A., Schaeffer-Filho, A., and Gaspary, L. P. (2021). Bungee: An adaptive pushback mechanism for ddos detection and mitigation in p4 data planes. In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 393–401.

[Gupta et al. 2018] Gupta, A., Harrison, R., Canini, M., Feamster, N., Rexford, J., and Willinger, W. (2018). Sonata: Query-driven streaming network telemetry. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '18, page 357–371, New York, NY, USA. Association for Computing Machinery.

[Gutiérrez et al. 2021] Gutiérrez, S. A., Branch, J. W., Gaspary, L. P., and Botero, J. F. (2021). Watching smartly from the bottom: Intrusion detection revamped through programmable networks and artificial intelligence. arXiv cs.NI 2106.00239.

[Guzman and Lewis 2020] Guzman, A. L. and Lewis, S. C. (2020). Artificial intelligence and communication: A human–machine communication research agenda. *New Media & Society*, 22(1):70–86.

[Hamed and Al-Shaer 2006] Hamed, H. and Al-Shaer, E. (2006). Taxonomy of conflicts in network security policies. *IEEE Communications Magazine*, 44(3):134–141.

[Haq et al. 2015] Haq, O., Abaid, Z., Bhatti, N., Ahmed, Z., and Syed, A. (2015). Sdn-inspired, real-time botnet detection and flow-blocking at isp and enterprise-level. In *2015 IEEE International Conference on Communications (ICC)*, pages 5278–5283. IEEE.

[Hoque et al. 2015] Hoque, N., Bhattacharyya, D. K., and Kalita, J. K. (2015). Botnet in ddos attacks: trends and challenges. *IEEE Communications Surveys & Tutorials*, 17(4):2242–2270.

[Huber et al. 2009] Huber, M., Kowalski, S., Nohlberg, M., and Tjoa, S. (2009). Towards automating social engineering using social networking sites. In *2009 International Conference on Computational Science and Engineering*, volume 3, pages 117–124. IEEE.

[Ilha et al. 2021] Ilha, A. d. S., Lapolli, A. C., Marques, J. A., and Gaspary, L. P. (2021). Euclid: A fully in-network, p4-based approach for real-time ddos attack detection and mitigation. *IEEE Transactions on Network and Service Management*, 18(3):3121–3139.

[Jepsen et al. 2019] Jepsen, T., Alvarez, D., Foster, N., Kim, C., Lee, J., Moshref, M., and Soulé, R. (2019). Fast string searching on pisa. In *Proceedings of the 2019 ACM Symposium on SDN Research*, SOSR '19, pages 21–28, New York, NY, USA. Association for Computing Machinery.

[Jin et al. 2017] Jin, X., Li, X., Zhang, H., Soulé, R., Lee, J., Foster, N., Kim, C., and Stoica, I. (2017). Netcache: Balancing key-value stores with fast in-network caching. SOSP '17.

[Kang et al. 2016] Kang, M. S., Gligor, V. D., Sekar, V., et al. (2016). Spiffy: Inducing cost-detectability tradeoffs for persistent link-flooding attacks. In *NDSS*, volume 1, pages 53–55.

[Khurshid et al. 2013] Khurshid, A., Zou, X., Zhou, W., Caesar, M., and Godfrey, P. B. (2013). {VeriFlow}: Verifying {Network-Wide} invariants in real time. In *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, pages 15–27.

[Kim et al. 2006] Kim, Y., Lau, W. C., Chuah, M. C., and Chao, H. J. (2006). Packetscore: a statistics-based packet filtering scheme against distributed denial-of-service attacks. *IEEE transactions on dependable and secure computing*, 3(2):141–155.

[Klimburg-Witjes and Wentland 2021] Klimburg-Witjes, N. and Wentland, A. (2021). Hacking humans? social engineering and the construction of the "deficient user" in cybersecurity discourses. *Science, Technology, & Human Values*, 46(6):1316–1339.

[Kohler et al. 2018] Kohler, T., Mayer, R., Dürr, F., Maaß, M., Bhowmik, S., and Rothermel, K. (2018). P4cep: Towards in-network complex event processing. In *Proceedings of the 2018 Morning Workshop on In-Network Computing*, NetCompute '18, page 33–38, New York, NY, USA. Association for Computing Machinery.

[Kreutz et al. 2013] Kreutz, D., Ramos, F. M., and Verissimo, P. (2013). Towards secure and dependable software-defined networks. In *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, HotSDN '13, pages 55–60, New York, NY, USA. ACM.

[Lapolli et al. 2019] Lapolli, Â. C., Marques, J. A., and Gaspary, L. P. (2019). Offloading real-time ddos attack detection to programmable data planes. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 19–27. IEEE.

[Le and Zincir-Heywood 2020] Le, D. C. and Zincir-Heywood, A. N. (2020). A frontier: Dependable, reliable and secure machine learning for network/system management. *J. Netw. Syst. Manag.*, 28(4):827–849.

[Liberato 2022] Liberato, M. (2022). Secbert: Analyzing reports using bert-like models. Master's thesis, University of Twente.

[Lopes et al. 2016] Lopes, N., Bjørner, N., McKeown, N., Rybalchenko, A., Talayco, D., and Varghese, G. (2016). Automatically verifying reachability and well-formedness in p4 networks. *Technical Report, Tech. Rep*.

[Lopes et al. 2015] Lopes, N. P., Bjørner, N., Godefroid, P., Jayaraman, K., and Varghese, G. (2015). Checking beliefs in dynamic networks. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pages 499–512, Oakland, CA. USENIX Association.

[Lupu and Sloman 1999] Lupu, E. C. and Sloman, M. (1999). Conflicts in policy-based distributed systems management. *IEEE Trans. Softw. Eng.*, 25(6):852–869.

[Mace and Fonseca 2018] Mace, J. and Fonseca, R. (2018). Universal context propagation for distributed system instrumentation. In *Proceedings of the Thirteenth EuroSys Conference*, EuroSys '18, pages 8:1–8:18, New York, NY, USA. ACM.

[Macías et al. 2020] Macías, S. G., Gaspary, L. P., and Botero, J. F. (2020). Oracle: Collaboration of data and control planes to detect ddos attacks. arXiv cs.NI 2009.10798.

[Marnerides et al. 2011] Marnerides, A., James, C., Schaeffer-Filho, A., Sait, S., Mauthe, A., and Murthy, H. (2011). Multi-level network resilience: Traffic analysis, anomaly detection and simulation. *ICTACT Journal on Communication Technology, Special Issue on Next Generation Wireless Networks and Applications*, 2:345–356.

[McKeown et al. 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review*, 38(2):69–74.

[Mirkovic et al. 2002] Mirkovic, J., Prier, G., and Reiher, P. (2002). Attacking ddos at the source. In *10th IEEE International Conference on Network Protocols, 2002. Proceedings.*, pages 312–321. IEEE.

[Mitnick and Simon 2003] Mitnick, K. D. and Simon, W. L. (2003). *The art of deception: Controlling the human element of security*. John Wiley & Sons.

[Moshref et al. 2013] Moshref, M., Yu, M., and Govindan, R. (2013). Resource/accuracy tradeoffs in software-defined measurement. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pages 73–78.

[Moshref et al. 2016] Moshref, M., Yu, M., Govindan, R., and Vahdat, A. (2016). Trumpet: Timely and precise triggers in data centers. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 129–143.

[Mustard et al. 2019] Mustard, C., Ruffy, F., Gakhokidze, A., Beschastnikh, I., and Fedorova, A. (2019). Jumpgate: In-Network processing as a service for data analytics. In *11th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 19)*, Renton, WA. USENIX Association.

[Neves et al. 2021] Neves, M., Huffaker, B., Levchenko, K., and Barcellos, M. (2021). Dynamic property enforcement in programmable data planes. *IEEE/ACM Transactions on Networking*, 29(4):1540–1552.

[Nijland 2022] Nijland, J. (2022). Gamification of cyber security awareness training for phishing against university students. B.S. thesis, University of Twente.

[Peng et al. 2004] Peng, T., Leckie, C., and Ramamohanarao, K. (2004). Proactively detecting distributed denial of service attacks using source ip address monitoring. In *International conference on research in networking*, pages 771–782. Springer.

[Phaal 2009] Phaal, P. (2009). sflow: Sampling rates. In *June 2009*.

[Pinheiro 2020] Pinheiro, P. P. (2020). Segurança digital: Proteção de dados nas empresas. *1ª edição. São Paulo, SP: Grupo GEN*.

[Ports and Nelson 2019] Ports, D. R. K. and Nelson, J. (2019). When should the network be the computer? In *Proceedings of the Workshop on Hot Topics in Operating Systems*, HotOS '19, page 209–215, New York, NY, USA. Association for Computing Machinery.

[Qin et al. 2020] Qin, Q., Poularakis, K., Leung, K. K., and Tassiulas, L. (2020). Line-speed and scalable intrusion detection at the network edge via federated learning. In *2020 IFIP Networking Conference (Networking)*, pages 352–360.

[Sanvito et al. 2018] Sanvito, D., Siracusano, G., and Bifulco, R. (2018). Can the network be the ai accelerator? In *Proceedings of the 2018 Morning Workshop on In-Network Computing*, NetCompute '18, pages 20–25, New York, NY, USA. Association for Computing Machinery.

[Sapio et al. 2017] Sapio, A., Abdelaziz, I., Aldilaijan, A., Canini, M., and Kalnis, P. (2017). In-network computation is a dumb idea whose time has come. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, HotNets-XVI, page 150–156, New York, NY, USA. Association for Computing Machinery.

[Sapio et al. 2021] Sapio, A., Canini, M., Ho, C.-Y., Nelson, J., Kalnis, P., Kim, C., Krishnamurthy, A., Moshref, M., Ports, D., and Richtarik, P. (2021). Scaling distributed machine learning with In-Network aggregation. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 785–808. USENIX Association.

[Sflow 2017] Sflow (2017). sflow.org - making the network visible. In *Accessed on June, 29 2017*.

[Shafahi et al. 2016] Shafahi, M., Kempers, L., and Afsarmanesh, H. (2016). Phishing through social bots on twitter. In *2016 IEEE International Conference on Big Data*, pages 3703–3712. IEEE.

[Sigelman et al. 2010] Sigelman, B. H., Barroso, L. A., Burrows, M., Stephenson, P., Plakal, M., Beaver, D., Jaspan, S., and Shanbhag, C. (2010). Dapper, a large-scale distributed systems tracing infrastructure. Technical report, Google, Inc.

[Stoenescu et al. 2016] Stoenescu, R., Popovici, M., Negreanu, L., and Raiciu, C. (2016). Symnet: Scalable symbolic execution for modern networks. In *ACM SIGCOMM 2016*, pages 314–327. ACM.

[Udupi et al. 2007] Udupi, Y. B., Sahai, A., and Singhal, S. (2007). A classification-based approach to policy refinement. In *2007 10th IFIP/IEEE International Symposium on Integrated Network Management*, pages 785–788. IEEE.

[Verma 2002] Verma, D. C. (2002). Simplifying network administration using policy-based management. *IEEE network*, 16(2):20–26.

[Westerinen et al. 2001] Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S., Huynh, A., Carlson, M., Perry, J., and Waldbusser, S. (2001). Terminology for policy-based management. Technical report.

[Winder 2020] Winder, D. (2020). Much of the internet went down yesterday: Here's the reason why. *Forbes*.

[Xie et al. 2019] Xie, J., Yu, F. R., Huang, T., Xie, R., Liu, J., Wang, C., and Liu, Y. (2019). A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges. *IEEE Communications Surveys & Tutorials*, 21(1):393–430.

[Xiong and Zilberman 2019a] Xiong, Z. and Zilberman, N. (2019a). Do switches dream of machine learning? toward in-network classification. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*, HotNets '19, pages 25–33, New York, NY, USA. Association for Computing Machinery.

[Xiong and Zilberman 2019b] Xiong, Z. and Zilberman, N. (2019b). Do switches dream of machine learning? toward in-network classification. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*, HotNets '19, page 25–33, New York, NY, USA. Association for Computing Machinery.

[Yu et al. 2011] Yu, M., Greenberg, A., Maltz, D., Rexford, J., Yuan, L., Kandula, S., and Kim, C. (2011). Profiling network performance for multi-tier data center applications. In *8th USENIX Symposium on Networked Systems Design and Implementation (NSDI 11)*.

[Zargar et al. 2013] Zargar, S. T., Joshi, J., and Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE communications surveys & tutorials*, 15(4):2046–2069.

Capítulo 8

“A Nova Eletricidade”: Aplicações, Riscos e Tendências da IA Moderna

Ana L. C. Bazzan, Anderson R. Tavares, André G. Pereira, Cláudio R. Jung, Jacob Scharcanski, Joel Carbonera, Luis C. Lamb, Mariana Recamonde-Mendoza, Thiago L. T. da Silveira, Viviane Moreira[1]

Programa de Pós-Graduação em Computação (PPGC) - UFRGS

Resumo

A provocativa comparação entre IA e eletricidade, feita pelo cientista da computação e empreendedor Andrew Ng, resume a profunda transformação que os recentes avanços em Inteligência Artificial (IA) têm desencadeado no mundo. Este capítulo apresenta uma visão geral pela paisagem em constante evolução da IA. Sem pretensões de exaurir o assunto, exploramos as aplicações que estão redefinindo setores da economia, impactando a sociedade e a humanidade. Analisamos os riscos que acompanham o rápido progresso tecnológico e as tendências futuras da IA, área que trilha o caminho para se tornar uma tecnologia de propósito geral, assim como a eletricidade, que revolucionou a sociedade dos séculos XIX e XX.

Abstract

The provocative comparison between AI and electricity, made by computer scientist and entrepreneur Andrew Ng, summarizes the deep transformation that recent advances in Artificial Intelligence (AI) have triggered in the world. This chapter provides an overview of the ever-evolving landscape of AI. Without intending to exhaust the subject, we explore the applications that are redefining sectors of the economy, impacting society and humanity. We analyze the risks that accompany rapid technological progress and future trends in AI, an area that is on the path to becoming a general-purpose technology, just like electricity, which revolutionized society in the 19th and 20th centuries

Vídeo com a apresentação do capítulo: `https://youtu.be/_1rtWWHFjdw`

[1]Lista de autores em ordem alfabética.

Parte I: Introdução e Fundamentos

8.1. Introdução

Comparar a Inteligência Artificial (IA) e a eletricidade foi a maneira que o cientista da computação e empreendedor Andrew Ng usou para sintetizar o potencial transformador e também os perigos dessa tecnologia [Lynch 2017]. Assim como a eletricidade moldou a história do século XIX, a IA está esculpindo o cenário do século XXI de maneiras que desafiam as fronteiras do conhecimento e da imaginação, com um farol de possibilidades intrigantes e, ao mesmo tempo, profundas preocupações sobre os rumos que a tecnologia pode tomar, mesmo quando usada para fins não-maliciosos.

Indo além da analogia IA-eletricidade de Andrew Ng, o status da IA como uma força transformadora foi reconhecido pela comunidade científica da Ciência da Computação. O prestigiado Prêmio Turing, também chamado de "Nobel da Computação", foi concedido em 2018 aos cientistas da computação Yoshua Bengio, Geoffrey Hinton e Yann LeCun por suas contribuições pioneiras para o desenvolvimento da aprendizagem profunda, o componente da IA no cerne da disrupção provocada na sociedade. Esse reconhecimento não apenas consagrou a significância da IA na era contemporânea, mas também destacou o papel crucial desses visionários em pavimentar o caminho para avanços que reverberam em todas as facetas da sociedade.

Este capítulo introduz conceitos básicos de IA (Seção 8.2) e, na Parte II, apresenta uma visão geral de suas implicações em diversas áreas: Visão Computacional (Seção 8.3), Processamento de Linguagem Natural (Seção 8.4), Saúde (Seção 8.5), Indústria (Seção 8.6), Finanças (Seção 8.7) e Mobilidade Urbana (Seção 8.8). Usando aspectos da provocativa analogia de Ng, em cada área serão apresentadas algumas aplicações, riscos e tendências. A Parte III apresentam um panorama geral do trabalho, revisitando riscos em comum nas diferentes áreas; discute a IA neuro-simbólica como uma abordagem promissora pra esses riscos; e conclui com um chamado à reflexão sobre os rumos da tecnologia e da humanidade. O capítulo não tem pretensões de esgotar o assunto, nem na listagem das áreas impactadas pela IA, pois praticamente todos os aspectos da vida em sociedade serão afetados, nem nas aplicações específicas de cada área. O conteúdo aqui apresentado é um convite para jornadas mais abrangentes e profundas do leitor, que poderá expandir seus horizontes nas referências apresentadas.

8.2. Fundamentos

O psicólogo e economista Daniel Kahneman, ganhador de um Prêmio Nobel de Economia, propôs que o raciocínio humano é dividido em dois sistemas [Kahneman 2011]. No Sistema 1, a mente trabalha de maneira instintiva, rápida, por reflexos, sujeita a erros e de maneira difícil de descrever (sem transparência). Este sistema é mais ativo em decisões rotineiras e tarefas mundanas, como os movimentos corretos a serem feitos enquanto se dirige um carro. No Sistema 2, a mente trabalha de maneira deliberada, lenta, confiável e transparente. Este sistema é mais ativo em decisões estratégicas e tarefas intelectuais, como a escolha de rotas enquanto se dirige um carro.

O modelo da mente humana dividida em Sistemas 1 e 2 é também útil para mapear

abordagens de inteligência artificial [Geffner 2018]. Abordagens baseadas em aprendizado, mais modernas em IA, estão mais próximas ao Sistema 1: modelos treinados dão respostas rápidas, sujeitas a erros e difíceis de rastrear (baixa transparência). Abordagens de IA simbólica, uma tradição dominante historicamente em IA até o início dos anos 2000, estão associadas ao Sistema 2: trata-se de métodos e algoritmos que enumeram explicitamente possíveis soluções para um problema e as investigam de maneira sistemática, sendo lentos, porém fáceis de rastrear (transparentes), pois é possível verificar o estado de um algoritmo e entender as decisões feitas por ele. Cabe ressaltar, no entanto, que Kahneman declarou que o Sistema 1 e Sistema 2 atuam de forma integrada, em debate sobre tendências em IA durante a conferência AAAI-2020 [2]. Assim, embora exista a distinção entre Sistemas 1 e 2, os mesmos podem ser vistos de forma harmônica, o que nos remete à IA neuro-simbólica [d'Avila Garcez and Lamb 2023], que analisaremos na Seção 8.10.

O restante desta seção apresenta fundamentos de IA simbólica (Seção 8.2.1) e IA baseada em aprendizado (Seção 8.2.2). Essa organização segue a ordem baseada na linha de tempo da IA, onde os métodos simbólicos (análogos ao Sistema 2 da mente humana) eram tradicionalmente predominantes, enquanto os métodos baseados em aprendizado (análogos ao Sistema 1 da mente humana) ganharam proeminência em tempos modernos, sendo responsáveis pela notoriedade atual da IA. Ao leitor interessado em se aprofundar nos conceitos apresentados aqui, o abrangente livro de [Russell and Norvig 2020] é uma excelente referência para os conceitos de IA em geral.

8.2.1. IA simbólica

Representação de conhecimento e raciocínio é uma área da IA preocupada com como o conhecimento pode ser representado de forma simbólica e manipulado de maneira automatizada por programas que representam processos de raciocínio realizados sobre as representações simbólicas [Brachman and Levesque 2004]. Esta abordagem da IA busca estudar e produzir comportamento inteligente sem considerar necessariamente a estrutura biológica subjacente ao processo de raciocínio, mas focando no conhecimento que os agentes possuem. Assim, parte-se do pressuposto que o que permite aos agentes (como humanos) se comportarem de maneira inteligente é que eles sabem muitas coisas sobre o seu ambiente e são capazes de aplicar esse conhecimento conforme necessário para se adaptar às situações que se apresentam para alcançar seus objetivos. Portanto, nesta abordagem de IA, focamos no conhecimento e na representação deste conhecimento. As questões-chave nesta área são o que qualquer agente (humano, animal, artificial, etc) precisaria saber para se comportar de maneira inteligente e que tipos de mecanismos computacionais poderiam permitir que seu conhecimento fosse manipulado para realizar inferências que possibilitem que o agente atinja seus objetivos [Fagin et al. 1995].

Na área de representação de conhecimento e raciocínio, o conhecimento é geralmente visto como uma coleção de proposições, que são formulações abstratas geralmente representadas por sentenças declarativas sobre o mundo (ou alguma parte ou aspecto dele) que podem ser verdadeiras ou falsas. Nesta área da IA, em geral, proposições são representadas por símbolos (como sequências de caracteres com sintaxe bem definida), que, ao

[2] AAAI-2020 Fireside Chat with Daniel Kahneman - com Francesca Rossi, Yoshua Bengio, Geoff Hinton e Yann LeCun. `https://vimeo.com/390814190` Acesso em 25 de setembro de 2023.

contrário das proposições, são entidades concretas e que permitem a manipulação das proposições por meios computacionais [Kowalski 1979]. Na abordagem simbólica, pesquisadores fazem utilização intensiva de formulações precisas, através das lógicas clássicas (proposicional e de predicados de primeira ordem) e lógicas não-clássicas, e.g. modais, temporais, epistêmicas, espaciais, probabilísticas, entre outras [Broda et al. 2004]. Assim, a representação de conhecimento é o campo de estudo focado em estudar o uso de símbolos formais para representar proposições que constituem o conhecimento de um agente. Neste contexto, raciocínio é a manipulação computacional deste símbolos que representam o conhecimento de um agente, visando produzir novos símbolos, que representam um novo conhecimento. Uma coleção de representações simbólicas de conhecimento constitui o que chamamos de uma *base de conhecimento*. Sistemas cujo comportamento inteligente é produzido pela manipulação das bases de conhecimento através de mecanismos computacionais que descrevem um processo de raciocínio são chamados de *sistemas baseados em conhecimento*. Neste tipo de sistema, adota-se uma abordagem declarativa, em que especificamos o que o agente sabe (onde fatos novos podem ser descobertos via percepção do ambiente) e o agente pode chegar a conclusões (que podem ser ações) através de inferências lógicas realizadas sobre sua base de conhecimento. Na abordagem declarativa de construção de sistemas, não especificamos o fluxo de controle da manipulação de dados, como em abordagens procedimentais.

Embora existam diversas abordagens, tipicamente a construção de sistemas ou agentes *baseados em conhecimento* envolve o uso de linguagens formais baseadas em lógica (como a lógica de primeira ordem e lógica modal, entre outras) para representar conhecimento dependente de tarefa e domínio. Com isso, podemos definir mecanismos de raciocínio independentes de tarefa e domínio, que realizam processos computacionais que representam inferências lógicas bem fundamentadas que garantem propriedades lógicas desejáveis, como a validade. Estes mecanismos de raciocínio, por sua vez, são capazes de derivar novo conhecimento a partir dos fatos armazenados na base de conhecimento. Historicamente, a área de IA simbólica teve grande impulso a partir dos anos 1970 com desenvolvimentos em programação em lógica [Kowalski 1979]. Especificamente, o desenvolvimento da linguagem Prolog (Programming in Logic) permitiu que pesquisadores passassem a expressar conhecimentos de domínios específicos de forma declarativa, sob uma fundamentação lógica e desenvolvessem sistemas computacionais a partir de uma abordagem simbólica. Prolog também teve grande impacto nos anos 1980 e 1990, notadamente durante o projeto liderado pelo Japão, denominado de "Fifth Generation Computer Systems Project". A linguagem Prolog foi aplicada com sucesso na prova automática de teoremas, sistemas especialistas, planejamento, bancos de dados, processamento de linguagem natural, aplicações legais entre outras áreas onde a representação de conhecimento e inferência lógica exigem uma representação computacional adequada [Warren et al. 2023].

Planejamento automatizado é uma subárea da IA simbólica que visa produzir solucionadores com comportamento direcionado a objetivos que sejam gerais e eficientes. Esses solucionadores, também chamados de planejadores, aceitam modelos que visam produzir comportamento direcionado a objetivos, sendo planejamento clássico um dos modelos mais pesquisados. Uma tarefa modelada por planejamento clássico possui um estado inicial, uma condição objetivo e um conjunto de operadores. Uma solução para

uma tarefa de planejamento é uma sequência de operadores que satisfazem a condição objetivo quando aplicados ao estado inicial. Um exemplo de tarefa modelada no planejamento clássico é um cenário em um armazém onde o objetivo é encontrar uma sequência de movimentos que os robôs devem realizar para alcançar suas posições-objetivo.

A motivação do planejamento automatizado é criar um planejador que tenha um bom desempenho em qualquer tarefa sem conhecimento prévio [Hoffmann 2011]. Isso torna o planejamento uma abordagem com bom custo-beneficio para desenvolvimento de soluções. Pode-se construir ou selecionar um planejador, descrever qualquer tarefa no modelo do planejador e então resolvê-la usando o planejador. Se a tarefa mudar, basta mudar o modelo da tarefa, mas não o planejador. Assim, utilizar planeadores para encontrar boas soluções para problemas do mundo real pode ajudar a reduzir tempo e custos.

Os planejadores mais bem-sucedidos baseiam-se em busca heurística. A^* é o algoritmo de busca heurística mais conhecido [Hart et al. 1968]. Ele processa nodos de maneira sistemática, buscando o caminho ótimo (de menor custo) entre o estado inicial e o objetivo, empregando uma função heurística para descartar caminhos não-promissores. Heurísticas bem construídas aumentam a eficiência do algoritmo enquanto mantém sua otimalidade.

Funções heurísticas usam o modelo da tarefa para raciocinar automaticamente sobre a mesma, gerando estimativas do custo para satisfazer a condição objetivo. Em geral, este processo de raciocínio utiliza relaxações ou abstrações para calcular as estimativas em tempo polinomial [Helmert and Domshlak 2009]. Por esta razão, os planejadores são chamados de independentes de domínio. Eles podem calcular estimativas diretamente do modelo sem conhecimento prévio sobre o domínio ou a tarefa.

8.2.2. IA baseada em aprendizado

IA baseada em aprendizado é também conhecida como Aprendizado de Máquina. Em aprendizado de máquina, os sistemas computacionais são *treinados* para realizar uma tarefa, ao invés de serem explicitamente programados para isso, como na IA simbólica. O ponto chave em aprendizado de máquina é que os sistemas computacionais consigam realizar uma tarefa e melhorar seu desempenho nela à medida que adquirem mais dados ou experiência [Mitchell 1997]. Esta seção apresenta uma visão geral da área, dividida nas três subáreas mais comuns: aprendizado supervisionado 8.2.2.1, não-supervisionado 8.2.2.3 e por reforço 8.2.2.4.

8.2.2.1. Aprendizado supervisionado

Aprendizado supervisionado é uma área relacionada a tarefas de predição. Exemplos rotineiros incluem: predizer a probabilidade de chuva a partir dos dados climáticos, qual o valor de um ativo a partir de seus dados históricos, quais objetos estão presentes em uma imagem, qual a próxima palavra a se escrever a partir das palavras já escritas, entre outros. Para se treinar um modelo de aprendizado supervisionado, é necessária a existência de um conjunto de dados rotulados, isto é, composto por instâncias definidas como pares de entrada e saída esperada. Através do seu processo de aprendizado, o algoritmo utilizará estes dados para encontrar padrões que mapeiam cada entrada para sua saída esperada.

Trata-se, portanto, de uma tarefa com *feedback* instrutivo, onde, para cada entrada, há a instrução de qual é a saída ou resposta correta.

Quando a saída esperada no conjunto de dados faz parte de um conjunto finito de possibilidades, a tarefa de aprendizado supervisionado é de *classificação*. Dentre os exemplos do início dessa seção, a predição de quais objetos estão presentes em uma imagem e qual a próxima palavra a se escrever a partir das palavras já escritas são tarefas de classificação. Quando a saída esperada no conjunto de dados é um número, a tarefa de aprendizado supervisionado é de *regressão*. Dentre os exemplos do início dessa seção, a predição da probabilidade de chuva a partir dos dados climáticos e do valor de um ativo a partir de seus dados históricos são tarefas de regressão.

Dentre as categorias de algoritmos de aprendizado supervisionado mais tradicionais, incluem-se (de maneira não-exaustiva):

- Métodos baseados em instâncias, como "k-vizinhos mais próximos", onde não há um modelo treinado, mas um dado recebe a classe de seus vizinhos mais próximos;
- Árvores de decisão, onde o conjunto de dados é sucessivamente particionado de acordo com os valores do atributo escolhido em cada ponto (nó) de decisão;
- Métodos probabilísticos, como Naïve Bayes, onde a probabilidade de uma instância ser de uma dada classe depende das ocorrência de seus atributos em cada classe do conjunto de treino;
- Métodos de combinação de modelos (*ensembles*), onde múltiplos modelos preditivos são combinados para melhor desempenho geral;
- Métodos conexionistas, como as redes neurais, que inspiram-se na capacidade de processamento de sinais do cérebro biológico.

A seguir, apresentamos uma breve descrição de redes neurais, visto que elas tem sido o componente principal dos sistemas mais modernos e disruptivos de IA.

8.2.2.2. Redes neurais e aprendizado profundo

Redes neurais são modelos computacionais inspirados na estrutura e funcionamento do cérebro biológico. De maneira simplificada, os neurônios que compõem o cérebro biológico são elementos capazes de processar sinais elétricos, recebendo sinais de outros neurônios, atenuando-os ou amplificando-os e combinando-os em um sinal de saída a ser processado por outros neurônios. [McCulloch and Pitts 1943] foram pioneiros ao propor uma versão matemática desse elemento: o neurônio artificial recebe "sinais" numéricos como entrada, e combina-os em uma soma ponderada, na qual pesos numéricos fazem o papel de atenuar ou amplificar os números recebidos como entrada. Os pesos são os parâmetros do neurônio e podem ser modificados para melhorar seu desempenho. Uma função de ativação é então aplicada a essa soma ponderada, resultando na saída do neurônio.

Quando os neurônios artificiais são organizados em camadas interconectadas e a função de ativação é não-linear, temos o perceptron multicamada, o modelo mais tradicional de redes neurais. Quando o perceptron multicamada tem pelo menos uma camada intermediária (ou oculta) entre a entrada e a saída, já é possível dizer que trata-se de aprendizado profundo, pois tal rede já é capaz de detectar e combinar características não-lineares nos dados de entrada [Goodfellow et al. 2016], embora alguns pesquisadores somente "reconhecem" como profundas as redes com dezenas de camadas.

O treinamento de uma rede neural é uma forma de otimização baseada em gradiente. Nessa modalidade, uma função de custo, contínua e diferenciável, avalia as diferenças entre predições da rede e saídas esperadas. O gradiente dessa função de custo indica a mudança necessária em cada parâmetro (peso) da rede para que o custo, e consequentemente os erros da rede, diminuam. Em termos práticos, o treino envolve a apresentação dos dados de entrada (números colocados na primeira camada), o cálculo das ativações da camada inicial até a final e a comparação com a saída esperada. Os pesos de todos os neurônios da rede são ajustados na direção determinada pela derivada parcial da função de custo com relação a cada peso. Essa derivada parcial é exatamente a mudança necessária em cada peso para que o custo diminua. Essas derivadas são calculadas da camada final da rede em retropropagação até a camada inicial. Esse procedimento é o clássico algoritmo *backpropagation* de [Rumelhart et al. 1986], cujos princípios são a base de praticamente todos os sistemas de aprendizado profundo.

De uma maneira simplificada, uma rede neural é um conjunto de parâmetros (números) que realiza transformações numéricas em suas entradas. A clássica organização em perceptron multicamadas é uma das formas de se estruturar uma rede. Ela é especialmente eficaz para dados estruturados, com características já extraídas. Porém, para processamento de dados não-estruturados, a extração de características é necessária. Arquiteturas específicas de redes neurais são capazes de fazer essa extração de características, também chamada de aprendizado de representações. A seguir, descrevemos brevemente dois tipos de redes neurais que se tornaram modelos básicos em suas respectivas áreas: redes neurais convolucionais (CNNs do inglês *convolutional neural networks*), extensivamente usadas em visão computacional (ver Seção 8.3) e Transformers, responsáveis por grandes avanços em processamento de linguagem natural (ver Seção 8.4).

Em processamento de imagens, convoluções são operações matemáticas, nas quais filtros (ou *kernels*) percorrem a imagem de entrada, na forma de uma janela deslizante, gerando um mapa de características. Os filtros definem uma característica de interesse a ser detectada e o mapa de característica resultante é uma "imagem", na qual os pixels da imagem original contendo a característica de interesse são destacados e os demais são atenuados. O objetivo é capturar padrões e características específicas, como bordas, texturas e formas, em diferentes partes da imagem. [LeCun et al. 1989] foram pioneiros ao vislumbrar que os filtros convolucionais não precisavam ser pré-definidos, mas poderiam ser aprendidos via *backpropagation* para extraírem características relevantes na tarefa em questão. Redes convolucionais estão no centro de várias aplicações em visão computacional, e o leitor interessado pode encontrar mais detalhes na Seção 8.3.

Textos em linguagem natural são sequências de palavras, caracterizando-se como dados não-estruturados. O aprendizado de representações em texto envolve uma série de

desafios significativos, em grande parte devido à natureza complexa, variável e ambígua da linguagem natural. Notáveis avanços foram feitos com a ideia de mapear uma palavra para um vetor cujas coordenadas no espaço dão uma ideia de significado [Mikolov et al. 2013]. Nessa abordagem, palavras similares ficam em coordenadas próximas e há possibilidade de operações aritméticas entre palavras, como o clássico exemplo onde "rei - homem + mulher = rainha". No entanto, essa abordagem tem a limitação de que palavras com diferentes significados são mapeadas para uma única representação, sendo insuficiente para saber, por exemplo se "banco" se refere à agência bancária ou ao local de se sentar. Transformers [Vaswani et al. 2017] resolvem esse problema com o mecanismo de atenção: a representação de cada palavra não mais é fixa, agora ela depende do contexto (demais palavras anteriores e posteriores). Tal mecanismo de atenção é composto de matrizes numéricas, cujos valores são aprendidos via *backpropagation*. Transformers são o motor dos sistemas de processamento de linguagem natural mais impressionantes da atualizade, como o ChatGPT. A Seção 8.4 apresenta mais informações sobre processamento de linguagem natural.

8.2.2.3. Aprendizado não-supervisionado

Aprendizado não-supervisionado lida com tarefas de descrição, em contraste com aprendizado supervisionado (predição) e por reforço (controle). Tarefas de descrição envolvem a extração de padrões, similaridades e informações ocultas nos dados sem a necessidade de rótulos ou supervisão explícita. De maneira sucinta, os principais tipos de tarefas descritivas são:

- Agrupamento (Clustering): o objetivo é dividir o conjunto de dados em grupos de acordo com medidas de similaridade. Tipos de agrupamento incluem particional, no qual o espaço de estados é dividido em subregiões disjuntas, hierárquico, no qual a divisão particional pode ter múltiplas granularidades, e por densidade, na qual os dados são agrupados de acordo com a densidade (muita aglomeração ou dispersão).
- Associação: o objetivo é identificar conjuntos de itens ou características que ocorrem juntos com alta frequência dentro de um conjunto de dados, revelando relações intrínsecas e estruturas subjacentes. Dentre as aplicações mais recorrentes destas técnicas, estão análise de cestas de compras e recomendação de produtos online.
- Redução de Dimensionalidade: o objetivo é obter uma representação simplificada dos dados de entrada, reduzindo o número de dimensões (equivalente às "colunas" em conjuntos de dados tabulares) onde cada dimensão na nova representação é uma combinação das dimensões da representação original. Técnicas de redução de dimensionalidade são rotineiramente utilizadas para visualização e compressão de dados, sendo parte essencial do *pipeline* de projetos em ciências de dados.

8.2.2.4. Aprendizado por reforço

Aprendizado por reforço (AR) é geralmente associado a tarefas de controle, ou seja, aprender a melhor ação a se realizar em cada situação do ambiente. Em contraste com

aprendizado supervisionado, em AR não se diz explicitamente ao aprendiz ou agente o que fazer ou qual a ação correta em uma dada situação. Ao invés disso, cada ação do agente é avaliada com um sinal numérico de recompensa, o qual dá ideia de qualidade imediata daquela ação. O próprio agente deve, por tentativa-e-erro, encontrar a ação que traz a maior quantidade de recompensas a longo prazo.

Dentre os métodos tradicionais de aprendizado por reforço, o clássico Q-learning [Watkins and Dayan 1992] é um dos pioneiros, e vários dos métodos mais bem sucedidos da atualidade usam seus princípios. No Q-learning, o agente mantém estimativas de valor (relacionado à soma das recompensas esperadas) para cada ação que possa executar em cada estado do ambiente. Ao interagir com o ambiente, o agente obtém uma amostra da recompensa real da ação que realizou no estado que estava. O Q-learning usa essa recompensa, além do valor do estado atingido, para atualizar suas estimativas. O Q-learning possui garantias teóricas de convergência de suas estimativas para os valores corretos [Watkins 1989]. Um agente treinado com Q-learning pode garantir o máximo possível de recompensa simplesmente selecionando a ação de maior valor em cada estado.

O Q-learning mantém as estimativas de valor das ações para cada estado em uma tabela. Se há muitos estados e/ou ações, tal representação não é viável. Uma maneira de resolver isso é usar uma função ao invés de uma tabela para as estimativas de valor. Tais funções podem receber representações contendo características dos estados e aplicar pesos, que podem ser aprendidos, para ponderar essas características [Sutton and Barto 2018, Parte II]. Em especial, a própria representação do estado pode ser aprendida, por exemplo, por uma rede neural profunda. Essa abordagem é colocada em prática no algoritmo Deep Q-Networks (DQN) [Mnih et al. 2015], o qual recebia pixels da tela e aprendeu a jogar jogos de Atari sem nenhum conhecimento prévio, obtendo desempenho sobrehumano em certos jogos.

Pode-se dizer que DQN inaugurou a "era do Aprendizado por Reforço Profundo", onde avanços substanciais continuam acontecendo. Tais avanços levaram métodos de aprendizado por reforço a obterem grande sucesso em jogos, desde jogos de tabuleiro [Silver et al. 2017b, Silver et al. 2017a] até video-games muito mais complexos que Atari [Berner et al. 2019, Vinyals et al. 2019], onde múltiplos jogadores devem responder rapidamente aos acontecimentos da tela enquanto traçam planos de longo prazo para vencer uma partida.

No entanto, a aplicação mais disruptiva de aprendizado por reforço foi em processamento de linguagem natural. Parte da metodologia de treino do ChatGPT consistiu em obter um modelo de recompensa através de *feedback* humano para textos gerados por um modelo pré-treinado e o refinamento do modelo gerador de textos para maximizar esta recompensa [Ouyang et al. 2022].

Para o leitor intressado, o livro de [Sutton and Barto 2018] é o principal livro-texto sobre aprendizado por reforço.

Parte II: Impactos da IA

Esta parte discute, de maneira não exaustiva, diversas áreas impactadas pela IA. Em cada área, são discutidas algumas aplicações, riscos e tendências. Inicialmente, discutimos "áreas meio", nas quais a IA tem relação com habilidades cognitivas humanas de visão (Seção 8.3) e linguagem (Seção 8.4). Avanços nas referidas áreas têm reflexo nas "áreas fim", nas quais a IA afeta diferentes aspectos da sociedade: Saúde (Seção 8.5), Indústria (Seção 8.6), Finanças (Seção 8.7) e Mobilidade Urbana (Seção 8.8).

8.3. Visão computacional e Processamento de imagens

Visão Computacional é um campo interdisciplinar que combina elementos de inteligência artificial, ótica e processamento de imagem. Trata-se de tornar as máquinas capazes de interpretar e extrair informações significativas a partir de imagens ou vídeos, permitindo a realização de tarefas como detecção de objetos, reconhecimento facial, análise de cenas, entre outras.

8.3.1. Aplicações

As áreas de visão computacional e processamento de imagens testemunharam enormes avanços nos últimos anos, em grande parte impulsionados por técnicas de aprendizado profundo. O aprendizado profundo, com sua capacidade de aprender automaticamente padrões complexos em grandes conjuntos de dados, revolucionou a maneira como abordamos as tarefas visuais. A sinergia entre visão computacional/processamento de imagens e aprendizado profundo levou a avanços significativos e abriu uma infinidade de aplicações práticas em vários setores e aplicações, algumas das quais brevemente listadas a seguir:

- Restauração e melhoria de imagens: algoritmos que integram processamento de imagens e aprendizado de máquina têm possibilitado a restauração visual de imagens degradadas, como remoção do ruído, aumento de resolução, correção de borramento por desfoco ou movimento, e correção de iluminação (sobretudo para imagens subexpostas).
- Detecção, Segmentação e Reconhecimento de Objetos: Algoritmos de detecção ou segmentação de objetos baseados em aprendizado profundo permitiram a identificação precisa e em tempo real de objetos em imagens e vídeos. Esta aplicação encontra uso prático em sistemas de vigilância, veículos autônomos e robótica. Por exemplo, carros autônomos usam a detecção de objetos para detectar pedestres, veículos e sinais de trânsito, e exploram segmentação para avaliar a área navegável.
- Reconhecimento facial e biometria: O reconhecimento facial é amplamente utilizado para fins de identificação e autenticação. Modelos de aprendizado profundo, como redes neurais convolucionais (CNNs), são capazes de identificar atributos faciais discriminatórios de cada indivíduo, permitindo o reconhecimento facial preciso mesmo em condições desafiadoras. Por exemplo, a biometria facial é empregada para autenticação de usuários em *smartphones*, sistemas de segurança e aplicativos de controle de acesso, agilizando processos e aumentando a segurança.

- Análise de Imagens Biomédicas: Conforme já discutido na Seção 8.5, o aprendizado profundo fez contribuições significativas para a análise de imagens biomédicas, auxiliando os profissionais de saúde a diagnosticar doenças com mais precisão e eficiência. Os modelos de aprendizado profundo podem detectar anomalias em raios-X, ressonâncias magnéticas e tomografias computadorizadas, auxiliando na detecção precoce de condições como câncer, doenças cardiovasculares e distúrbios neurológicos. Além disso, a visão computacional também facilitou a análise das lâminas histopatológicas, ajudando os patologistas a identificar e classificar as células cancerígenas.

- Realidade Aumentada (RA) e Realidade Virtual (RV): a visão computacional baseada em aprendizado profundo desempenha um papel crucial no desenvolvimento de aplicativos para RA e RV. Em particular, algoritmos de reconstrução tridimensional (3D) a partir de uma ou mais imagens podem ser usados para modelar ambientes reais, permitindo uma experiência imersiva com óculos de VR. Além disso, permitem rastrear e reconhecer objetos e cenas do mundo real, possibilitando a inserção de objetos sintéticos no ambiente do usuário (RA).

- Agronegócios e análise ambiental: a visão computacional baseada em aprendizado profundo transformou a agricultura com aplicativos como monitoramento de colheitas, detecção de doenças em vegetais e estimativa de rendimento. Drones equipados com câmeras e algoritmos de aprendizado profundo podem analisar vastas terras agrícolas, identificando áreas de preocupação e permitindo a aplicação precisa de fertilizantes e pesticidas. Além disso, o processamento de imagens aéreas e de satélite permitem o monitoramento de condições ambientais, como desmatamento, incêndios e enchentes.

- Varejo e *E-commerce*: A visão computacional também encontra lugar nos setores de varejo e comércio eletrônico. Os varejistas podem usar a visão computacional para rastrear o comportamento do cliente em suas lojas, analisar o tráfego de pedestres e otimizar os *layouts* das lojas para um melhor envolvimento do cliente. As plataformas de comércio eletrônico usam o reconhecimento de imagem para oferecer recomendações de produtos visualmente semelhantes, aprimorando a experiência de compra dos clientes.

- Automação Industrial: técnicas de visão computacional podem ser utilizadas em ambientes industriais, para a identificação de defeitos em produtos de maneira rápida e eficiente. Além disso, conjuntamente com a robótica, por exemplo, pode-se fazer com que robôs equipados com câmeras acessem ambientes de difícil acesso e realizem monitoramento continuado da infraestrutura.

8.3.2. Riscos

Apesar dos grandes avanços nos últimos anos, uma série de precauções devem ser tomadas antes do uso irrestrito de algoritmos de visão computacional baseados em aprendizado de máquina.

Um potencial problema se refere a *ataques*, nos quais uma imagem é manipulada para “enganar” um algoritmo de aprendizado de máquina. Por exemplo, Eykholt e

colegas [Eykholt et al. 2018] apresentaram uma estratégia para realizar *ataques físicos* focados no problema de detecção de sinais de trânsito (crucial para veículos autônomos). Como exemplo, mostraram que adesivos brancos e pretos colados a uma placa de trânsito mudam completamente o resultado de classificação de uma rede neural, apesar de manter o sinal completamente compreensível para o ser humano.

Um outro perigo potencial no uso de algoritmos de visão computacional baseados em aprendizado de máquina é a imprevisibilidade dos resultados em dados novos. Como a maioria das técnicas é supervisionada, são necessários dados de treinamento anotados. E como dados anotados são custosos de produzir e altamente dependentes do problema e aplicação, eles são disponibilizados em quantidade limitada. Por outro lado, aplicações de visão computacional envolvem dados nunca vistos pelas redes, que podem apresentar características distintas dos dados de treinamento, potencialmente causando degradação da qualidade. Essa variabilidade de características dos *datasets* é chamada de mudança de domínio (*domain shift*), e pode envolver diversos parâmetros (e.g., treinamento em dados capturados durante o dia e teste com dados capturados durante a noite). Como exemplo, Hasan et al. [Hasan et al. 2021] avaliaram o a capacidade de generalização de diversos algoritmos de detecção de pedestres, concluindo que a maioria apresenta resultados impressionantes em uma validação *intra-dataset*, mas com degradação acentuada em validações *cross-datasets*, mesmo quando o domínio alvo possui poucas diferenças visuais com relação ao domínio fonte.

Outra questão importante envolve os vieses, como os diversos problemas demonstrados por pesquisadores em sistemas de reconhecimento facial[3]. Notadamente, [Buolamwini and Gebru 2018] mostraram que algoritmos de reconhecimento facial discriminavam por raça e gênero.

8.3.3. Tendências

Por muito tempo, a grande maioria das redes neurais envolvendo imagens era baseada em camadas convolucionais. Embora elas tenha uma representação compacta em termos de número de parâmetros e estejam relacionadas com o funcionamento do sistema visual humano, o uso de convoluções assume um filtro com suporte espacial limitado. Assim, *pixels* muito distantes entre si podem não se relacionar em uma rede convolucional. Uma tendência crescente envolve o uso de camadas de atenção espacial, dentre as quais os *Transformers* são populares. O modelo Visual Transformer (ViT) [Dosovitskiy et al. 2020] estende o conceito de *Transformers*, originalmente desenvolvidos para textos, para o domínio de imagens.

Outra tendência atual é o uso integrado de dados visuais e textuais, dando origem às *Vision Language Models*. Como exemplo, o modelo CLIP (*Contrastive Language Image Pre-raining*) [Radford et al. 2021] usa uma base de 400 milhões de imagens pareadas com as respectivas descrições textuais, treinando os *embeddings* de texto e de imagens de tal maneira que eles sejam similares. Com esse tipo de abordagem, se pode fazer con-

[3]"Study finds gender and skin-type bias in commercial artificial-intelligence systems: Examination of facial-analysis software shows error rate of 0.8 percent for light-skinned men, 34.7 percent for dark-skinned women."MIT News, 11 de Fevereiro de 2018. https://news.mit.edu/2018/study-finds-gender-skin-type-bias-artificial-intelligence-systems-0212. Acesso em 28 de Agosto de 2023

sultas a imagens com dados de entrada textuais, e vice-versa. Com isso, diversos novos modelos e bases de dados têm sido propostas nos últimos anos. O *dataset* LAION-5B, por exemplo, fornece 5,85 bilhões de pares texto-imagens. Em particular, se mostrou que redes profundas treinadas com uma quantidade muito grande de dados (como CLIP) podem ser customizadas com poucos dados adicionais para tarefas específicas. Tais redes são chamadas de *Foundation Models*.

8.4. Processamento de Linguagem Natural

Processamento de Linguagem Natural (PLN) é uma área multidisciplinar que combina ciência da computação e linguística com o objetivo de permitir que as máquinas realizem tarefas úteis com a linguagem humana. PLN geralmente envolve o processamento de grandes volumes de dados textuais ou de fala, que são chamados de *corpus* (ou *corpora* no plural). O PLN tem uma longa história e vem sendo estudado desde os anos 1950. A ideia de fazer com que os computadores consigam compreender a linguagem humana é tão antiga quanto a computação. Alguns dos exemplos mais icônicos vieram da ficção científica, como o HAL-9000 do filme “2001: Uma Odisseia no Espaço” dirigido por Stanley Kubrick (1968) e escrito por Kubrick e Artur C. Clarke, baseado em obra anterior deste intitulada "The Sentinel". O computador HAL-9000 exibia uma série de habilidades impressionantes envolvendo a compreensão e a geração de linguagem. Nos últimos anos, algumas dessas habilidades foram enfim atingidas por algoritmos reais. O imenso avanço em PLN se deve principalmente à evolução dos algoritmos de aprendizado profundo. Mais especificamente, o desenvolvimento da arquitetura de Transformers [Vaswani et al. 2017] usada nos grandes modelos de linguagem, do inglês *large language model* (LLM), como BERT [Devlin et al. 2019] e GPT [Brown et al. 2020], melhoraram sensivelmente os resultados em diversas tarefas de compreensão e de geração de texto. Pode-se dizer que as mudanças de maior impacto na imprensa e, notadamente disruptivas na computação nos últimos anos, vieram da área de PLN.

8.4.1. Aplicações

O PLN pode ser empregado em uma vasta gama de aplicações, tanto científicas como industriais. A seguir, fornecemos uma lista não exaustiva de aplicações.

Diversas tarefas de PLN podem ser modeladas como problemas de **classificação de textos**, por exemplo: análise de sentimento, filtragem de spam, identificação de discurso de ódio, atribuição de autoria, detecção de plágio, *etc*. Em todos esses casos, a entrada do algoritmo é uma sequência de tokens (*i.e.,* palavras) e a saída é a classe predita. Até meados dos anos 2000, os algoritmos mais empregados para classificação de textos eram os de aprendizado de máquina tradicional como Naïve Bayes, máquinas de vetores de suporte e árvores de decisão. A partir de 2014, as redes neurais profundas como o LSTM [Hochreiter and Schmidhuber 1997] passaram a predominar. De 2019 para cá, a hegemonia de modelos baseados em Transformers, principalmente o BERT [Devlin et al. 2019], é notável. O ganho de qualidade obtido nessas tarefas pode ser atribuído à ideia de trabalhar em duas etapas: o *pré-treinamento* e o *ajuste fino*. No pré-treinamento, o algoritmo analisa um corpus de grande volume (*i.e.,* contendo bilhões de palavras) a fim de aprender características da linguagem como relações entre palavras e coerência entre frases. O corpus usado nessa fase é apenas texto puro, que pode ser facilmente obtido a

partir da Web. Uma vez que o modelo tenha sido pré-treinado com grandes corpora, ele pode ser ajustado para desempenhar uma tarefa específica, como análise de sentimento, por exemplo. Nessa fase, o modelo precisa de um conjunto de dados rotulado com a classe esperada – o processo tradicional de aprendizado supervisionado. A vantagem aqui, é que o "conhecimento"que o modelo pré-treinado já possuía sobre a linguagem é aproveitado na tarefa específica e traz ganhos na qualidade das predições geradas.

A **sumarização de textos** tem por objetivo gerar uma versão reduzida do texto de entrada que contemple suas ideias principais. Diferentemente das tarefas de classificação em que a saída é uma só (*e.g.*, a classe predita), na sumarização a saída, assim como a entrada, é uma sequência de tokens. Algoritmos de sumarização podem ser empregados em diversos domínios como financeiro, jurídico, científico para reduzir a quantidade de texto que as pessoas precisam processar. As técnicas de sumarização dividem-se em *extrativas* e *abstrativas*. As técnicas extrativas usam técnicas estatísticas para selecionar as k frases mais significativas do texto original e as usam para compor o resumo. A desvantagem é que a coerência do texto gerado é prejudicada pois a conexão entre as frases pode ser perdida. Já as técnicas abstrativas usam LLMs para tentar simular o comportamento de um sumarizador humano combinando sentenças e utilizando paráfrases para gerar um texto mais fluido. A desvantagem aqui é que esses modelos podem "alucinar"e adicionar conteúdo no resumo que não tem base no texto original (veja mais sobre esse problema na Seção 8.4.2).

A **tradução automática** (TA) é outra aplicação importante de PLN que obteve melhorias significativas com a introdução dos algoritmos de aprendizado profundo. A tarefa consiste em transformar uma sequência de um idioma fonte para um idioma alvo. O exemplo mais conhecido é de tradutor automático é o GoogleTranslate[4]. Assim como a sumarização, tanto a entrada como a saída do algoritmo são sequências de tokens. A TA usa aprendizado supervisionado: durante o treinamento, o sistema processa um grande número de sentenças paralelas (as mesmas sentenças escritas no idioma fonte e no idioma alvo) e assim aprende o mapeamento entre idiomas. As arquiteturas mais usadas em sistemas de TA são LSTMs bidirecionais [Schuster and Paliwal 1997] e Transformers [Vaswani et al. 2017]. Além disso, vale ressaltar que esses sistemas trabalham em nível de subpalavras. Assim, eles conseguem aprender traduções para fragmentos, *e.g.*, o sufixo de gerúndio "*endo*"em português é normalmente traduzido para "*ing*"em inglês. A tradução automática é desafiadora por uma série de razões: (*i*) a ambiguidade que já é problemática em um idioma, fica ainda mais complexa quando adicionamos mais idiomas, (*ii*) a estrutura dos idiomas pode ser muito diferente e isso faz com que a ordem das palavras no idioma alvo possa ser muito diferente da ordem no idioma fonte, (*iii*) nem sempre há uma palavra correspondente no idioma alvo – pode não haver nenhuma ou podem haver várias traduções possíveis e (*iv*) vocabulários de domínios específicos (*e.g.*, medicina, computação, direito) dificilmente terão grandes volumes de sentenças paralelas para permitir a geração de mapeamentos precisos.

Os **chatbots**, também chamados de agentes conversacionais são programas que se comunicam com as pessoas usando linguagem natural. O exemplo mais marcante de

[4]`https://translate.google.com`

chatbot é o ChatGPT[5] que foi lançado em novembro de 2022 e em menos de dois meses já tinha mais de 100 milhões de usuários. O ChatGPT também é um modelo baseado em Transformers e suas capacidades vão além de conseguir manter diálogos, ele consegue escrever código de programas e compor músicas e poemas. Apesar de não ter representado um avanço científico (pois as tecnologias utilizadas já haviam sido justificadas), o impacto do ChatGPT foi gigantesco – poucas ferramentas geraram tanto interesse como essa. O mecanismo por trás dessas habilidades é a geração de texto que, assim como a geração de imagens, faz parte da IA generativa. A geração do texto usando LLMs é conhecida por geração autorregressiva ou geração causal. Ela consiste basicamente em escolher a próxima palavra a ser gerada condicionada às escolhas anteriores e à pergunta feita pelo usuário. Esse processo é conhecido como *next word prediction* (ou predição da próxima palavra). Essas escolhas dependem das estatísticas de ocorrência das palavras em grandes corpora de textos e em uma série de parâmetros que controlam a aleatoriedade do processo de geração. Os riscos desses sistemas são discutidos na Seção 8.4.2.

Até agora, esta seção abordou apenas texto. Contudo, PLN também trata de fala. A habilidade de **reconhecer e produzir fala** são muito relevantes e úteis em uma série de aplicações que interagem com os usuários por meio de voz como assistentes inteligentes (como a Siri da Apple e a Alexa da Amazon). A comunicação por meio de voz envolve o reconhecimento da fala (*Automatic Speech Recognition*) (ASR) que transforma de fala para texto e a conversão de texto para fala *Text to Speech* (TTS). ASR é uma tarefa bastante desafiadora pois precisa lidar com variações na forma de pronunciar as palavras (diferentes sotaques e velocidades de fala), ruídos de fundo e disfluências (sons como "hum"e "hã"). Tanto ASR como TTS atualmente são implementados utilizando LSTMs bidirecionais [Schuster and Paliwal 1997] ou Transformers [Vaswani et al. 2017]. Por ser mais difícil, ASR comumente precisa de mais dados de treinamento (*i.e.,* mais horas de áudio pareadas com o texto correspondente).

8.4.2. Riscos

Os principais riscos associados ao uso de aplicações de PLN advém, principalmente, de quatro problemas: o viés dos dados, as alucinações, o potencial para mau uso e o custo do treinamento de LLMs.

Os LLMs são treinados com grandes volumes de textos coletados a partir da web. Esses dados não passam por um processo de curadoria e podem conter diversos tipos de **viés** (racismo, sexismo, homofobia, xenofobia, *etc.*). O problema é que, ao gerar modelos a partir desses dados, os modelos passam a replicar esses vieses. [Papakyriakopoulos et al. 2020] observaram que até mesmo representações geradas a partir de textos da Wikipedia apresentam sexismo, homofobia e xenofobia. Também investigando vieses, mas na área de tradução automática, [Prates et al. 2020] mostraram que o Google Translate apresentava uma forte tendência de tradução para *defaults* masculinos em experimentos realizados a partir de uma lista abrangente de cargos do "Bureau of Labor Statistics"dos EUA. No artigo, traduções como “Ele/Ela é um Engenheiro” (onde “Engenheiro” é substituído por o cargo de interesse) em 12 idiomas diferentes de gênero neutro mostram que tradutor (que usa técnicas de IA) não consegue reproduzir uma distribuição real de traba-

[5] `https://chat.openai.com/`

lhadoras. O artigo mostra que o Google Translate produz padrões masculinos com muito mais frequência do que seria esperado apenas com base nos dados demográficos. Esses trabalhos indicam que é necessário o desenvolvimento de abordagens mais sofisticadas para a construção de sistemas que sigam princípios éticos, respeitando as diversidades populacionais, culturais, nacionais, de gênero, raça e muitas outras [Russell et al. 2015].

O segundo risco afeta os sistemas que geram texto de maneira autorregressiva: as **alucinações**. As alucinações referem-se a situações em que um modelo de linguagem gera texto que contém informações que não estão presentes nos dados de treinamento, ou seja, o modelo gera fatos falsos. Há vários casos que foram divulgados na imprensa e mídias sociais envolvendo desde erros mais inofensivos até a imputação de crimes a pessoas inocentes. A principal causa é a forma como esses modelos geram os textos: eles não têm nenhuma compreensão acerca da realidade que os textos descrevem. Pesquisadoras críticas dessa abordagem referem-se a esses modelos como "papagaios estocásticos"[Bender et al. 2021]. É importante ressaltar que ferramentas que apenas geram texto não substituem motores de busca (como o Google e Bing, por exemplo) pois elas não têm como apontar as fontes para as informações. Quando solicitadas, elas podem até mesmo criar referências falsas.

A qualidade dos textos gerados automaticamente pode ser útil em uma série de tarefas, mas por outro lado, abre possibilidades para o mau uso. Há relatos de advogados que usaram ferramentas como ChatGPT e Bard[6] para redigir processos, de candidatos a empregos que geraram currículos automaticamente contendo dados "inflados", de alunos que entregaram códigos de programa escritos pela ferramenta como sendo de sua autoria, de geração de notícias falsas, entre outros.

Por fim, com o aumento de ordens de grandeza no tamanho dos LLMs (*e.g.,* de 117 milhões de parâmetros do GPT2 para 175 bilhões no GPT3 – e um número desconhecido no GPT4) , o custo do treinamento desses modelos e o seu impacto ambiental também vêm sendo discutidos. Estimativas mencionam [Sharir et al. 2020] que o treinamento de um modelo com 1,5 bilhões de parâmetros possa chegar a US$ 1,6 bilhões.

8.4.3. Tendências

Sob a perspectiva acadêmica, obter contribuições de impacto em PLN está cada vez mais difícil pois as universidades com seus orçamentos reduzidos precisam competir com gigantes do mercado de tecnologia como a Microsoft e Google. Levando isso em consideração, um artigo recente de pesquisadores da Universidade de Michigan [Ignat et al. 2023] aponta algumas futuras direções de pesquisa. Dentre elas, destacamos o desenvolvimento de modelos multilíngues e para idiomas com poucos recursos, a incorporação de um raciocínio que tenha fundamentação no mundo real para reduzir o problema das alucinações, o investimento em interpretabilidade dos modelos para possibilitar que as predições sejam explicadas e a aplicação de PLN em domínios relevantes como a saúde e a educação.

A maturidade das técnicas de PLN e os bons resultados que vêm atingindo contribuem que elas sejam disseminadas e adotadas na indústria. A ampla disponibilidade de LLMs e modelos ajustados para as mais diversas tarefas facilita a sua implantação em

[6] `https://bard.google.com/`

sistemas, ferramentas e aplicativos que venham ser usado por um número cada vez maior de pessoas.

8.5. Saúde

A Saúde tem sido apontada desde cedo como uma das áreas de aplicação mais promissoras para a IA. Os primeiros exemplos de sucesso, ainda na década de 1970, tratavam-se de sistemas especialistas dependentes de conhecimento humano prévio e um conjunto de regras definidas para apoio à tomada de decisão clínica. Estes sistemas demonstraram utilidade para auxiliar na definição de diagnóstico ou na recomendação de tratamentos para pacientes, mas com um potencial muito limitado devido aos desafios de se representar um conhecimento complexo via regras e da incapacidade de extrapolar o conhecimento prévio a fim de aprimorar a tomada de decisão [Yu et al. 2018].

Desde então, impulsionada pelo aumento na disponibilidade de dados em saúde e pelo rápido progresso de algoritmos capazes de aprender padrões relevantes e acionáveis a partir de dados volumosos e complexos, a IA vem gradualmente revolucionando a área da Saúde. Aplicações inovadoras baseadas em IA, especialmente em aprendizado de máquina, estão provocando mudanças significativas na forma como abordamos a medicina e os cuidados de saúde, sejam individuais ou coletivos. Esta seção revisa os aspectos principais da intersecção entre IA e Saúde no que tange aplicações, riscos e tendências.

8.5.1. Aplicações

Embora praticamente todos os aspectos da prestação de cuidados de saúde sejam passíveis de uso e implementação de IA, quatro eixos se destacam nos esforços recentes, incluindo aqueles concentrados em países de baixa e média renda (LMICs, segundo sua sigla em inglês): (i) diagnóstico, (ii) avaliação do risco de morbidade ou mortalidade do paciente, (iii) previsão e vigilância de surtos de doenças e (iv) planejamento de políticas de saúde pública. [Schwalbe and Wahl 2020].

Sistemas para diagnóstico médico baseado em IA têm sido amplamente explorados em diversas áreas, mas alcançaram uma maturidade particular em especialidades como a radiologia, oftalmologia, patologia e dermatologia. Estes sistemas baseiam-se principalmente em dados de imagens médicas (e.g., ressonância magnética, tomografia computadorizada, fotografias de lesões ou de lâminas histopatológicas), demonstrando um desempenho diagnóstico via IA equivalente ao desempenho dos especialistas da saúde para casos de câncer de pele, câncer de mama, retinopatia diabética, doenças respiratórias, dentre outros [Liu et al. 2019]. Sinais biomédicos (e.g., eletrocardiograma e eletroencefalograma), exames laboratoriais, dados genéticos (e.g., mutações no DNA e expressão gênica) e informações de prontuários médicos eletrônicos também foram utilizados com sucesso nas mais diversas especialidades médicas, e a evolução da IA vem possibilitando que os médicos façam diagnósticos mais rápidos e precisos. A IA foi empregada, por exemplo, para estratificação de risco em pacientes com infarto do miocárdio por oclusão [Al-Zaiti et al. 2023], detecção precoce da doença de Alzheimer [Mahendran and PM 2022] e estimativa de risco de câncer de pulmão em 3 anos a partir de tomografia computadorizada e outras informações clínicas [Huang et al. 2019].

A IA também tem sido uma tecnologia fundamental para aprimorar a capacidade

de quantificar riscos de eventos desfavoráveis ou agravos relacionados à saúde de um paciente. Durante a pandemia da COVID-19, algoritmos de aprendizado de máquina foram amplamente aplicados para estimar quais pacientes infectados têm maior probabilidade de sofrer com uma doença mais severa ou vir a óbito pela COVID-19 ou suas complicações [Van der Schaar et al. 2021]. No estudo de [Phakhounthong et al. 2018], indicadores clínicos e laboratoriais foram utilizados para desenvolver um modelo baseado em IA para prever casos graves de dengue entre pacientes pediátricos durante a admissão. A avaliação de riscos propicia um melhor monitoramento do paciente e um tratamento mais efetivo através da antecipação de condutas clínicas, além de possibilitar uma melhor gestão de recursos hospitalares.

Os benefícios da IA também podem ser observados na análise de riscos de Saúde em nível coletivo ou populacional, contribuindo para o estudo da dinâmica de doenças e para uma melhor vigilância epidemiológica explorando uma grande variedade de dados, inclusive traços digitais (e.g., pesquisas na internet, atividades em redes sociais) [Brownstein et al. 2023]. [Jiang et al. 2018] utilizaram aprendizado de máquina para estimar a probabilidade de surto epidêmico de Zika em nível global, conseguindo melhor modelar a complexidade e não-linearidade da relação entre o risco de transmissão por Zika vírus e fatores climáticos, ambientais e sócio-econômicos. [Brownstein et al. 2023] apontam que a IA tornou-se grande aliada na vigilância de doenças infecciosas, viabilizando o desenvolvimento de sistemas de alerta precoce para surtos de doenças, a identificação de focos de surtos ou de patógenos causadores de doenças, o rastreamento de contato e a previsão eficaz do risco de transmissão. Assim, a IA possibilita que autoridades de saúde pública respondam adequadamente ao risco que se apresenta, por exemplo, alocando recursos ou suprimentos adequados diante da expectativa de aumento de casos de uma determinada doença em uma região. Adicionalmente, o uso da IA possui grande impacto no planejamento de políticas de saúde pública, ao permitir a elaboração de medidas mais eficazes para proteção e promoção da saúde, como o planejamento de campanhas de vacinação e do direcionamento de materiais de divulgação de prevenção e cuidados com saúde com base no perfil de risco pessoal e padrões comportamentais [Panch et al. 2019].

Por fim, a IA se estende à análise de dados moleculares e genômicos, desempenhando um papel fundamental nas pesquisas biomédicas e possibilitando expandir nosso conhecimento sobre o funcionamento das doenças. A IA tem sido uma das principais propulsoras da medicina de precisão, especialmente na área da oncologia, revelando assinaturas moleculares associadas a subtipos ou estágios tumorais [Marczyk et al. 2023], e identificando novos biomarcadores [Colombelli et al. 2022], incluindo aqueles úteis para detecção não invasiva de câncer e avaliação de prognóstico [Xu et al. 2019]. Adicionalmente, avanços recentes como o AlphaFold [Jumper et al. 2021], que se utiliza de aprendizado profundo para prever com alta precisão as estruturas tridimensionais das proteínas a partir de sua sequência de aminoácidos, permitem facilmente avaliar o impacto funcional de variantes genéticas e acelerar a descoberta de novas drogas.

8.5.2. Riscos

Apesar do notável aumento nas pesquisas relacionadas às aplicações da IA na área da Saúde, é importante destacar que apenas um conjunto limitado destas soluções foi efetivamente implementado na prática clínica [Rajpurkar et al. 2022]. A Food and Drug

Administration (FDA), agência reguladora vinculada ao Departamento de Saúde e Serviços Humanos dos Estados Unidos, tem desempenhado um papel ativo na revisão e autorização para comercialização de um número crescente de dispositivos médicos que incorporam IA. No entanto, até a última atualização em outubro de 2022[7], constatou-se a aprovação de apenas 521 dispositivos médicos pelo FDA. Dentre esses dispositivos, 56,23% são voltados para aplicações em Radiologia e 10,94% na Cardiologia. A disparidade evidente entre a extensa quantidade de pesquisas científicas conduzidas nesse domínio e o número limitado de soluções práticas adotadas reflete uma série de desafios que permeiam a integração da IA na prática médica.

Embora diversos fatores possam contribuir para esse cenário, existe um consenso na comunidade acadêmica de que a falta de validação dos modelos baseados em IA por meio de dados externos constitui um dos principais fatores que inibem a aplicação efetiva do conhecimento científico adquirido [Liu et al. 2019]. Esta validação deveria ser feita com dados prospectivamente coletados a partir do mundo real para este propósito específico, e seguindo uma metodologia criteriosa de avaliação, como aquelas adotadas em ensaios clínicos randomizados. Modelos de IA podem falhar na generalização para novos tipos de dados nos quais não foram treinados. Alguns trabalhos já demonstram que a capacidade preditiva de um modelo é impactada negativamente quando o modelo é aplicado a uma população de pacientes diferente dos seus dados de treinamento [Wong et al. 2021]. Isto se deve à ampla heterogeneidade dos dados neste domínio devido a diferenças existentes nas práticas hospitalares e nos dados demográficos dos pacientes entre diferentes hospitais.

Outro ponto crítico é que o treinamento de modelos de IA em conjuntos de dados com pouca representatividade de grupos marginalizados ou com variações injustificadas para determinados grupos resulta em sistemas tendenciosos que apresentam baixo desempenho preditivo nesses grupos. Assim, sem o controle adequado, o uso da IA introduz o risco de perpetuar vieses ocultos nos dados e reforçar preconceitos e desigualdades sociais existentes. Por exemplo, um viés racial foi detectado em um algoritmo de avaliação de risco clínico utilizado nos Estados Unidos, que atribuía menor risco a pacientes negros em comparação com pacientes brancos igualmente doentes por utilizar custos de saúde como um proxy para as necessidades de saúde [Obermeyer et al. 2019]. Em outro estudo, um viés étnico foi identificado em escores de risco poligênico utilizados para estimar o risco de um indivíduo desenvolver doenças como câncer com base em fatores genéticos, possuindo acurácia muito superior em indivíduos de ascendência Europeia do que para outras ancestralidades em razão da coleta desequilibrada de dados genéticos e genômicos entre continentes [Martin et al. 2019].

Estes riscos são exacerbados na impossibilidade de explicar a tomada de decisão pelos modelos e avaliar até que ponto a mesma reflete as abordagens humanas especializadas e não fere princípios éticos fundamentais. A Organização Mundial da Saúde (OMS) [World Health Organization 2021] chama atenção, ainda, para os vieses derivados de exclusão digital. Em alguns LMICs, mulheres têm menos acesso a telefone celular

[7] `https://www.fda.gov/medical-devices/software-medical-device-samd/artificial-intelligence-and-machine-learning-aiml-enabled-medical-devices`. Acesso em 14 de Agosto de 2023

ou internet móvel do que homens, contribuindo com menos dados para treinamento de modelos de IA e sendo menos propensas a se beneficiar do uso desta tecnologia [World Health Organization 2021]. A fim de gerar modelos baseados em IA que possam promover equidade em Saúde, é imprescindível garantir disponibilidade e qualidade de dados, com diversidade em relação a contextos sociais, culturais e econômicos. Por fim, é inevitável apontar o risco de violação da privacidade por se tratar de dados sensíveis, e o risco de uso indevido de dados pessoais, visto que muitas vezes os modelos são treinados com base de dados retrospectivas, originalmente coletadas para outros propósitos de pesquisa.

8.5.3. Tendências

A Organização Mundial da Saúde (OMS) [World Health Organization 2021] reconhece o enorme potencial da IA para promover melhorias na medicina e alavancar a equidade dos cuidados em saúde. Para que este potencial se concretize, avanços ainda se fazem necessários em diversas frentes, sendo algumas mais críticas para o domínio da saúde.

Técnicas para mitigar vieses, sejam estes oriundos dos próprios dados ou resultantes do processo de treinamento dos modelos, são primordiais para evitar que se perpetuem desigualdades sociais existentes ou que se introduzam comportamentos tendenciosos nos modelos que possam produzir resultados discriminatórios contra determinados grupos. Entretanto, garantir maior equidade através do uso dos modelos também requer uma capacidade mais apurada de explicar as predições realizadas pelos mesmos a fim de identificar erros sistemáticos indesejáveis. Neste sentido, pesquisas em torno da explicabilidade de modelos são essenciais a fim de expandir a capacidade de encontrar fatores relevantes para as predições realizadas com base não somente em associações, mas em relações causais entre as variáveis de entrada e o resultado do modelo. Uma explicabilidade baseada em causalidade tornaria a interação especialistas-IA muito mais efetiva para a investigação da tomada de decisão feita pelos modelos, resultando em maior confiança na implementação prática destes modelos.

Por fim, salienta-se que a tomada de decisão em um ambiente clínico é inerentemente baseada em múltiplas evidências, sendo portanto crucial ampliar a capacidade dos algoritmos de aprenderem a partir de dados multimodais. Desta forma, modelos multimodais de IA visam possibilitar o uso de todas as fontes de dados normalmente disponíveis aos médicos ou que possam enriquecer a definição de um diagnostico, como dados clínicos e laboratoriais, exames de imagens, testes genéticos, determinantes sociais, fatores ambientais ou comportamentais, informações coletadas por *wearables*, dentre outros. Estas direções de pesquisa estão entre os principais pontos de acesso para fornecer cuidados ao paciente mais oportunos, precisos e justos com auxílio da IA.

8.6. Indústria

Da mesma forma que a eletricidade transformou drasticamente a indústria na segunda revolução industrial, a IA tem sido apontada como uma das promotoras de grandes transformações na indústria atualmente. Nos últimos anos, temos testemunhado uma crescente adoção de abordagens de IA nos mais diferentes setores da indústria, tais como energia [Pivetta et al. 2023, Rahmanifard and Plaksina 2019], manufatura [Li et al. 2017], indústria química [Baum et al. 2021], agricultura industrial [Benos et al. 2021] , etc.

Atualmente, a IA é apontada como um fator viabilizador com papel crucial na chamada *indústria 4.0* (quarta revolução industrial), que tem como característica fundamental o foco no desenvolvimento de *indústrias inteligentes*. Este cenário surge graças ao desenvolvimento e integração da IA com diferentes tecnologias, tais como redes de dados, internet das coisas, computação em nuvem, automação de processos físicos, sistemas ciber-físicos, etc.

Em um cenário típico de uma indústria inteligente alinhada à industria 4.0, a fábrica é constituída por coleções de sistemas ciber-físicos, que estabelecem uma interação profunda entre processos físicos e processos computacionais. Neste contexto, processos computacionais distribuídos controlam elementos físicos e sensores realizam continuamente o monitoramento dos processos e componentes físicos, retroalimentando os processos computacionais. Neste contexto, a IA desempenha um papel crucial na tomada de decisão que controla estes processos continuamente [dos Anjos et al. 2023].

A seguir, serão discutidas algumas aplicações de IA na indústria, bem como os riscos e as tendências associadas ao uso de IA neste contexto.

8.6.1. Aplicações

A IA vem sendo aplicada nos mais diversos setores industriais, das mais diferentes formas, incluindo otimização e automação de processos produtivos, previsão de demandas e de produção, identificação de perfil de clientes, desenvolvimento de produtos com IA, automação de atendimento ao cliente, desenvolvimento de novos produtos, etc. Nestes contextos, a aplicação de técnicas de IA visa aumentar a produtividade, reduzir custos, tornar as linhas de produção mais seguras, etc.

Uma das aplicações mais notórias da IA na indústria diz respeito ao uso destas tecnologias para automação de processos produtivos [Ribeiro et al. 2021, Fragapane et al. 2022]. A automação, neste caso, envolve principalmente a utilização de robôs, ou sensores e atuadores distribuídos ao longo das linhas de produção. Neste cenário, sistemas de IA utilizam dados de sensores para tomar decisões e controlar os atuadores ao longo da linha de produção. Nestes cenários, o uso da IA promove o aumento da produtividade e o desenvolvimento de linhas de produção mais flexíveis.

Técnicas de IA também vêm sendo largamente utilizadas na indústria para detectar anomalias em comportamentos de sistemas, de processos produtivos, etc [Stojanovic et al. 2016, Zipfel et al. 2023]. Anomalias, neste cenários, são padrões de comportamento diferentes do comportamento esperado [Martí et al. 2015]. Nestes contextos, em geral, são aplicadas técnicas de aprendizado de máquina para treinar algoritmos que identifiquem estados normais e anômalos dos sistemas e processos de interesse. Estes algoritmos costumam ser treinados a partir de dados de sensores que caracterizam os estados dos processos e sistemas ao longo do tempo. Esta abordagem é utilizada, por exemplo, para detecção em tempo real de possíveis vazamentos em oleodutos [Aljameel et al. 2022]. Em alguns casos, anomalias podem ser detectadas do modo visual também [Roth et al. 2022], em cenários em que as anomalias não são bem representadas por medidas de sensores convencionais (como medidas de pressão e temperatura, etc), mas se tornam aparentes através da inspeção visual. Estas abordagens são muito comuns, por exemplo, para detectar defeitos em produtos em linhas de produção [Birlutiu et al. 2017],

permitindo a remoção do produto defeituoso do processo para eventuais correções dos defeitos. Nestas abordagens, técnicas de aprendizado de máquina podem ser utilizadas para aprender padrões que caracterizam produtos com e sem defeitos a partir de grandes conjuntos de imagens previamente rotuladas.

Além de aperfeiçoar os processos produtivos, tecnologias de IA também vêm sendo utilizadas na indústria para a previsão de demandas e para o gerenciamento da cadeia de suprimentos necessários para suprir estas demandas [Zhu et al. 2021, Toorajipour et al. 2021], incluindo a previsão de oferta de insumos e seleção de fornecedores. Muitas das aplicações nesta área vêm utilizando técnicas de aprendizado de máquina capazes de lidar com dados em séries temporais.

No contexto da indústria 4.0, os sistemas produtivos tendem a ser altamente sensorizados, de modo que uma grande quantidade de medidas são continuamente adquiridas dos equipamentos ao longo do tempo. Neste cenário, estes dados podem fornecer valiosos *insights* sobre o estado dos equipamentos. Estes fatores vêm permitindo o desenvolvimento de técnicas de *manutenção preditiva* [Paolanti et al. 2018, Paolanti et al. 2018, Serradilla et al. 2022, Dalzochio et al. 2020] baseadas em técnicas de aprendizado de máquina. Abordagens de manutenção preditiva visam monitorar o estado dos equipamentos com o intuito de prever eventuais momentos de falha antes que elas ocorram, permitindo a redução de custos oriundos de paradas não programadas na produção, ou ainda evitando falhas que podem comprometer drasticamente as plantas de produção.

Nos últimos anos, a IA vem sendo utilizada até mesmo no processo de *design* de novos produtos [Aphirakmethawong et al. 2022]. Aplicações típicas de IA em design de produtos vêm utilizando as mais diversas abordagens de IA, incluindo desde algoritmos genéticos [Kielarova and Pradujphongphet 2023] a aprendizado de máquina [Zhang et al. 2019, Fournier-Viger et al. 2021, Hamolia and Melnyk 2021]. Cabe destacar que nos últimos anos técnicas de IA generativa vêm demonstrando capacidades impressionantes em tarefas de design [Grisoni et al. 2021]. Modelos de IA generativa, como ChatGPT e Dall-E, são capazes de aprender padrões a partir de grandes massas de dados (imagens, textos, etc) e gerar saídas que reproduzem esses padrões de forma verossímil. Algumas aplicações representativas de IA em design de produtos incluem o projeto de circuitos integrados [Gubbi et al. 2022, Wang and Luo 2019, Hamolia and Melnyk 2021], desenvolvimento de novas drogas [Grisoni et al. 2021], desenvolvimento de peças de vestuário na indústria da moda [Liang et al. 2020, Giri et al. 2019], desenvolvimento de produtos na indústria alimentícia [Zhang et al. 2019], etc.

É importante salientar que as aplicações industriais de técnicas de IA são vastas, abrangendo muitos setores e muitas tarefas diferentes, de modo que nesta seção são discutidos apenas alguns exemplos.

8.6.2. Riscos

A aplicação da IA na indústria herda boa parte dos riscos da IA aplicada em contextos gerais. Um destes riscos, e que pode impactar aplicações industriais de diversas formas, é o da falta de generalização de modelos de aprendizado de máquina. Em caso de falha na generalização destes modelos, sistemas de IA podem cometer erros em casos em que precisam lidar com situações muito diferentes das representadas nos dados de treinamento

ou com dados capturados por sensores com características técnicas diferentes dos sensores que coletaram os dados de treinamento. Em contextos industriais, erros no processo de decisão acarretados por modelos que não generalizaram adequadamente podem causar diversos impactos negativos. Por exemplo, em casos em que o ambiente industrial possui atuadores controlados por modelos sem a devida generalização, falhas no processo de decisão podem disparar ações (como movimentos de braços robóticos) que podem eventualmente ferir seres humanos que também atuam no ambiente industrial [Franklin et al. 2020]. Outros exemplos do impacto negativo da falta de generalização incluem detectar incorretamente defeitos em produtos, o que pode fazer com que produtos defeituosos sejam mantidos ou produtos sem defeitos sejam removidos das linhas de produção.

Apesar da extensa pesquisa na área de aprendizado de máquina visando encontrar maneiras de mitigar o problema da generalização, ainda existem diversos desafios relacionados à própria identificação adequada do domínio de validade dos modelos de aprendizado de máquina. Ou seja, dado um modelo de aprendizado de máquina treinado em um certo conjunto de dados, não é trivial determinar quais são os conjuntos de situações em que ele funciona adequadamente ou não. Esta dificuldade pode tornar modelos de aprendizado de máquina suscetíveis aos chamados ataques adversários [Narodytska and Kasiviswanathan 2017], em que alguém mal intencionado pode alterar sutilmente os dados de entrada (de um modo imperceptível para seres humanos) de certos modelos com o intuito de perverter o comportamento esperado. Estas dificuldades estão em grande parte associadas à dificuldade de se explicar de forma significativa o que de fato foi aprendido pelo modelo. Essa dificuldade associada à explicabilidade de modelos de aprendizado de máquina vem sendo apontada como um risco pela indústria em geral.

Além disso, atualmente há uma grande discussão a respeito das consequências da aplicação da IA no mercado de trabalho [Agrawal et al. 2019]. A utilização da IA na indústria, em geral, promove um aumento da automatização dos mais diferentes processos. No passado, os processos de automatização atingiram principalmente os aspectos físicos dos processos industriais. Mas com aplicações de tecnologias de IA estamos testemunhando também o impacto em aspectos intelectuais do trabalho. Esta tendência gera ainda mais impactos na oferta de empregos, diminuindo a oferta de certos postos, mas eventualmente proporcionando o surgimento de novas profissões.

8.6.3. Tendências

Uma pergunta-chave, tendo em vista a ubiquidade da IA, é como identificar tendências relevantes para negócios? Um análise ampla de tendências na área de IA pode ser realizada de diversas formas. Muitas vezes, a abordagem acadêmica utilizada é da identificação de áreas de classificação de artigos em publicações. No entanto, esta é uma abordagem obviamente limitada às bases de consulta e a preferências das conferências e revistas no que se refere a áreas de pesquisas. O atual impacto da IA - ressalte-se - surge a partir de uma subárea que era pouco valorizada na academia por um período de mais de uma década: redes neurais artificiais. Entre meados da década de 1990 até 2006, quando Geoffrey Hinton e seus alunos publicaram o primeiro artigo no qual os autores se referem

a redes neurais profundas[8] [Hinton et al. 2006], poucos autores consideravam o aprendizado conexionista como sendo uma grande tendência futura em IA. Assim, pensar em tendências, como diria Niels Bohr, é prever o futuro - e não há algo mais difícil do que prever o futuro em ciência.[9]

Consultorias especializadas em tecnologia, como Gartner, IDC, McKinsey e diversas outras analisam e identificam periodicamente diversas áreas da computação que terão impacto ao longo do tempo, bem como sua maturidade [10]. Do ponto de vista de mercado, tais estudos têm grande relevância, pois orientam profissionais e gestores. Na última década, outros relatórios sobre análise e tendências em IA têm sido publicados por centros de pesquisa, reunindo parcerias entre a academia e as empresas. Entre eles destacamos o AI Index Report, produzido sob a coordenação do Human-Centered AI Institute da Universidade de Stanford[11]. Este relatório, publicado anualmente desde 2017, destaca as tendências em pesquisa e desenvolvimente (através de análise de publicações), performance técnica (onde se analisam os progressos tecnológicos e seus impactos), ética em IA (equidade, viéses e suas implicações), impacto econômico (utlização da IA em negócios, investimentos públicos e privados), educação (nas escolas e universidades), políticas e governança (estratégias nacionais e multilaterais de governos), diversidade (notadamente na academia e as iniciativas para seu incremento) e opinião pública (análise da percepção pública sobre o impacto da IA). Uma observação relevante do AI Index Report[12] é que até 2014, a maior parte dos sistemas de aprendizado de máquina eram produzidos pela academia. Desde então, as empresas passaram a dominar a produção destas tecnologias. Os dados do relatório indicam que produzir sistemas de aprendizado de máquina de estado-da-arte requer grandes volumes de dados, poder computacional e recursos financeiros não disponíveis às universidades. Isto pode sugerir que, assim como demais tecnologias do passado, a partir do momento em que a viabilidade técnica e o alto potencial econômico de uma tecnologia são demonstradas, os investimentos neste setor tendem e se consolidar nas empresas.

8.7. Mercado de Capitais e Finanças

Finanças tem dois aspectos interessantes, sendo ao mesmo tempo uma arte e uma ciência. Então, pode-se entender finanças como a arte e a ciência da gestão de ativos financeiros. Mas as pessoas frequentemente se deparam a questão recorrente a seguir: *porque eu deveria me interessar pelo que acontece no mercado de capitais?*. Para abordar esta questão seria interessante discutir o que acontece no mercado de capitais.

O mercado de capitais é para onde os governos recorrem para fechar suas contas (geralmente pedindo empréstimos pela venda de pequenos 'pedaços' da dívida do governo, ex: títulos de dívida como os conhecidos 'títulos do tesouro'). A razão principal

[8]Deep neural networks, no caso do artigo [Hinton et al. 2006] se referem a "deep belief networks", uma arquitetura de redes neurais para aprendizado.

[9]"Prediction is very difficult, especially if it's about the future."Frase atribuída a Niels Bohr e, também, a Yogi Berra.

[10]`https://www.gartner.com` e `https://www.idc.com/`

[11]`https://aiindex.stanford.edu/report/`

[12]AI Index Report 2023, Capítulo 1, Página 50. `https://aiindex.stanford.edu/wp-content/uploads/2023/04/HAI_AI-Index-Report_2023.pdf`

para o governo ir ao mercado de capitais é honrar seus compromissos, tais como pagar benefícios sociais (ex: aposentadorias, seguro-desemprego, etc.), pagar outras contas obrigatórias (ex: saúde, educação, etc.), ou desenvolver seus projetos (ex: casa própria, saneamento básico, etc.). As empresas públicas e privadas também recorrem ao mercado de capitais para pedir empréstimos e cumprir com suas obrigações (ex: pagar dívidas vencendo em prazos curtos), desenvolver seus projetos (ex: investir em serviços licitados tais a expansão da rede de saneamento básico, do sistema de geração e distribuição de energia, etc.). As pessoas físicas também recorrem ao mercado de capitais para obter recursos e atingir diversos objetivos (ex: projetos futuros, casa própria, aposentadoria, etc.).

Como o mercado de capitais mobiliza a poupança, gere riscos, aloca eficientemente recursos e promove o aumento da disciplina corporativa, toda a sociedade é beneficiada. Ao aplicar sua poupança em capital produtivo, os investidores (individuais ou institucionais) causam movimentos de capitais e buscam uma alocação eficiente e com menor custo. Isso aumenta a liquidez da economia e os prazos dos investimentos. Para fazer essa alocação de capitais (recursos), os participantes do mercado exigem qualidade na governança corporativa e o compartilhamento de informação por parte das empresas que captam estes recursos, levando a maior disciplina e transparência, o que impacta na produtividade e no retorno sobre o investimento realizado. O resultado final, em um nível macro-econômico, é mais emprego, renda, investimento e crescimento econômico, o que impacta positivamente os principais indicadores socioeconômicos do país. Esse ciclo virtuoso apontado acima também traz resultados indiretos, tais como a viabilização e o desenvolvimento de mais projetos, a expansão da produção e da criação de riqueza, a criação de mais empregos, e o aumento da renda do cidadão. Portanto, desenvolvendo o mercado de capitais, promove-se o desenvolvimento socioeconômicos do país.

Então, parece natural que alguém tenha interesse pelo que acontece no mercado de capitais e no mundo das finanças, mas para identificar a relação entre a computação, o mercado de capitais e o mundo das finanças, precisamos observar mais detalhadamente o que acontece no dia-a-dia do mercado de capitais.

Qualquer transação no mercado de capitais envolve um acordo entre duas partes: (a) tomador de capital e (b) provedor de capital. Geralmente, quem provê capital o faz esperando algum retorno (ex: um valor referente ao aluguel do capital via juros, um lucro para remunerar o capital investido em uma transação, etc.). Por outro lado, quem toma capital também o faz esperando algum retorno (ex: atingir o objetivo de adquirir um bem de capital ou ativo real, desenvolver um projeto, etc.). Como há muitas partes interagindo direta ou indiretamente no mercado de capitais, com propósitos muito diferentes, existe uma grande variedade de instrumentos disponíveis para atender aos diferentes interesses das partes, possibilitado que elas interajam e atinjam seus objetivos através do uso de instrumentos específicos. Estas interações podem ocorrer em diferentes ambientes (ex: há ambientes em que existe livre negociação de instrumentos entre partes, como nas bolsas de valores, e existem ambientes onde a negociação ocorre dentro de restrições, como no caso das interações entre as empresas e seus clientes). Esta diversidade de tipos de instrumentos transacionados entre partes e o grande volume de transações tendem a tornar o dia-a-dia do mercado de capitais complexa. Cada instrumento transacionado entre partes provê informações sobre o mercado, seus segmentos, partes envolvidas e sobre a própria economia, em diferentes níveis (ex: micro ou macro-econômico, local, regional, nacional

ou mesmo internacional). Devido a grande complexidade das operações realizadas, da necessidade de rastrear e armazenar o enorme volume de informações gerado, o mercado de capitais migrou quase em sua totalidade para o ambiente digital e as transações são computadorizadas.

8.7.1. Aplicações

Hoje, a IA participa da automação de tarefas rotineiras, prove acessibilidade a serviços, e capacidade de aprendizado para aperfeiçoar tarefas rotineiras nos mercados, de forma exata, eficiente e rápida. Alguns dos temas abordados com o auxílio da IA no dia-a-dia do mercado de capitais e em finanças são: 1) análise ou inferência de sentimento dos participantes do mercado com base em textos, ou mesmo de mídias sociais; 2) detecção de transações suspeitas ou fraudulentas, ameaças e/ou crimes financeiros, ou ainda ameaças cibernéticas; 3) identificação de riscos e vantagens potenciais de ativos transacionados nos mercados; 4) avaliação e recomendação de instrumentos financeiros (ex: produtos e serviços) para potenciais interessados(as), com base nas suas preferências e objetivos; 5) processamento de dados estruturados e não estruturados, tais como documentos, para extrair dados relevantes e alimentar os processos de análise, previsão e recomendação (ex: descoberta de oportunidades de investimento); 6) uso de estimativas de risco, dados de transações e complementares para prever com razoabilidade resultados futuros; 7) sintetize de informações relevantes para a tomada de decisão usando IA generativa.

Por exemplo, já é comum alguém 'falar' com um robot ao tentar abrir uma conta bancária pela internet, ou mesmo ser entrevistado(a) por um robot para direcionar uma chamada telefônica a uma instituição financeira. O Business Insider[13] estima que as aplicações de IA vão economizar para as instituições financeiras nos EUA, em 2023, cerca de USD 447 bilhões. A maioria dos bancos (cerca de 80%) avaliam como positivo o impacto da IA neste segmento da indústria, e pretendem acelerar a migração das suas operações físicas para o ambiente digital (ex: um dos maiores investimentos do Banco Mercantil do Brasil em 2023 é focado no aumento da acessibilidade digital). Mani Nagasundaram (Senior VP, Global Financial Services, HCL Technologies) afirmou recentemente em um artigo na AI News[14] que a IA tende a liberar pessoal de tarefas rotineiras, e ao mesmo tempo melhorar a qualidade e a segurança do acesso a serviços financeiros, além de contribuir para trazer inovação ao ambiente corporativo. Já a Forbes[15] sugere em um artigo recente que 70% das empresas do setor financeiro já usam IA para prever eventos que possam afetar o seu fluxo de caixa (antecipando a necessidade de caixa para as operações diárias), ajustar os limites de crédito dos clientes e detectar fraudes no uso dos serviços (ex: cartões de crédito). Também, de acordo com a Forbes[16], IA tem sido usada para identificar as tendências mais recentes dos mercados, e avaliar os perfis das carteiras de

[13] https://www.insiderintelligence.com/insights/ai-in-finance/

[14] https://www.artificialintelligence-news.com/2020/12/15/from-experimentation-to-implementation-how-ai-is-proving-its-worth-in-financial-services/

[15] https://www.forbes.com/sites/louiscolumbus/2020/10/31/the-state-of-ai-adoption-in-financial-services/?sh=711a8c4e2aac

[16] https://www.forbes.com/sites/jayadkisson/2019/01/23/artificial-intelligence-will-replace-your-financial-adviser-and-thats-a-good-thing/?sh=1a02118e6b40

investimento disponíveis e dos clientes, para então sugerir os quais investimentos seriam adequados para cada perfil de cliente. Como IA é usada frequentemente para analisar padrões em grandes conjuntos de dados, naturalmente esta habilidade da IA tem sido usada em negociações nos mercados abertos (ex: bolsas). Pois os métodos computacionais baseados em IA podem analisar dados mais rápido e com maior exatidão que os humanos, além de poderem aprender a serem mais eficientes e otimizar as negociações nestes mercados (ex: algoritmos inteligentes já são usados para achar interessados em fixar a taxa de conversão Dolar-Real de um contrato de exportação, e os interessados em contratar esta taxa de conversão Dolar-Real na data futura desejada; ou ainda, algoritmos inteligentes já são usados para identificar tendências nos mercados e sugerir transações de ativos financeiros que estejam alinhadas com estas tendências).

8.7.2. Riscos

Não se pode ignorar que existem desafios éticos e riscos a serem mitigados para que o uso da IA nos mercados e em finanças seja efetivo, especialmente no que se refere a proteção de informações sensíveis e financeiras dos participantes, e a proteção dos participantes dos riscos que o uso das informações providas por algoritmos podem trazer. O Fintech Times [17] aponta três temas sensíveis que merecem atenção ao introduzir recursos de IA no setor financeiro e nos mercados em geral:

- *Ausência de Viés*: Antever que podem ocorrer falhas no projeto de algoritmos, e consequências indesejadas. Por exemplo, um algoritmo falho poderia adquirir um comportamento predatório ao considerar a necessidade da instituição financeira de otimizar sua lucratividade nas operações. Um algoritmo falho poderia se tornar predatório errando na estimativa da capacidade de um cliente de se endividar e de assumir riscos, e então oferecer ativos muito arriscados para este cliente que não teria condições de correr tais riscos nem de assumir as dívidas associadas a eles;

- *Responsabilização e Regulação*: É importante: a) regulamentar quais fontes de informações podem ser usadas e o uso correto destas informações, e b) definir antecipadamente de quem seria a responsabilidade se houverem consequências indesejadas de possíveis erros gerados por algoritmos, e como lidar com estas consequências (ex: informações incorretas ou inexatas podem induzir a erros na tomada de decisão e/ou em transações);

- *Transparência*: O que levou o algoritmo a tomar uma decisão, ou executar uma ação, deveria ser conhecido e rastreável.

Também é importante chamar a atenção para o fato de que *algoritmos podem ser usados como armas*. Portanto, o uso de algoritmos para propósitos antiéticos, tais como roubo de informações sensíveis de clientes, deve ser evitado e responsabilizado.

[17] `https://thefintechtimes.com/the-ethics-of-ai-ai-in-the-financial-services-sector-grand-opportunities-and-great-challenges/`

8.7.3. Tendências

Ao consolidar tarefas e analisar dados de forma mais rápida e exata que os humanos, espera-se que o uso intensivo de IA economize mais de USD 1 trilhão para os bancos e instituições financeiras nos EUA até 2030[18]. McKinsey Co.[19] estima que o sistema financeiro deverá ser transformado pelas mudanças tecnológicas, e as instituições financeiras precisarão aumentar seus investimentos em tecnologia da informação e IA para atingir altos níveis de digitalização, com qualidade. Será uma questão de sobrevivência, pois McKinsey Co. também estima que mais de 78% dos clientes jovens não iriam na agência física de uma instituição financeira se tivessem uma alternativa.

8.8. Mobilidade Urbana

A agenda em torno de cidades inteligentes tem como um dos focos a mobilidade urbana inteligente (uso racional dos diversos meios de transporte, integrando-os e adaptando-os à demanda). Existem diversas possibilidades em relação ao uso de IA em geral – e aprendizado de máquina em particular – em tal agenda. O restante desta seção joga luz em alguns aspectos a respeito de como a IA vem contribuindo, e como seu papel se torna cada vez mais decisivo. Desta forma, um verdadeiro trânsito inteligente resultará de indivíduos, semáforos e veículos conectados e trabalhando em conjunto. Nesta visão, semáforos inteligentes são alimentados com informação a respeito do estado da rede de tráfego, sobre os sobre semáforos vizinhos, eventos imprevistos, e outras informações.

Uma explicação mais detalhada sobre sistemas de transporte e simulação de tráfego pode ser encontrada em [Bazzan and Klügl 2013, Bazzan 2021]. A seguir, serão abordados dois problemas centrais, os quais motivam diversas aplicações de IA. O primeiro se dá pelo lado da demanda (como se deslocar de A até B de maneira eficiente), enquanto que o segundo se refere ao lado da oferta (controle e gerenciamento de tráfego). Devido à limitação de espaço, nos concentramos no tráfego veicular urbano.

8.8.1. Aplicações

Em relação à oferta, quando se fala em mobilidade inteligente, as pessoas em geral pensam em semáforos inteligentes. Algoritmos e técnicas de controle semafórico existem há várias décadas e derivam principalmente de técnicas de pesquisa operacional e da área de controle. Mais recentemente, técnicas de IA e aprendizado de máquina têm sido empregadas, em especial aqueles que se baseiam em AR (Seção 8.2.2.4). Neste caso, os semáforos devem aprender uma política que mapeia os estados (normalmente as filas nas interseções) para ações. Devido ao número de trabalhos que empregam AR no controle semafórico, e às diversas modelagens e técnicas empregadas, sugere-se consultar os *surveys* [Bazzan 2009, Wei et al. 2019, Yau et al. 2017, Noaeen et al. 2022].

Já pelo lado da demanda, entender como um motorista se comporta é fundamental em um sistema de recomendação de rotas e disseminação de informação aos motoristas. Não são muitos os trabalhos que consideram IA neste contexto. Redes neurais são utilizadas em [Dia and Panwai 2014] e em [Barthélemy and Carletti 2017] para prever e guiar,

[18] https://www.processmaker.com/blog/why-ai-is-the-future-of-finance/

[19] https://www.mckinsey.com/industries/financial-services/our-insights/ai-bank-of-the-future-can-banks-meet-the-ai-challenge

respectivamente, a escolha de rota dos motoristas. No caso da pesquisa mais recente, o foco é em: disseminação de informação, comunicação veicular, como aprender a escolher rotas, efeito de mudanças de comportamento da parte dos motoristas na presença de informação, e como disseminar informação de modo a garantir um determinado nível de desempenho do sistema. Para atingir tais objetivos, diversos métodos foram propostos no nosso grupo de pesquisa; para uma visão geral, ver [Bazzan 2022]. Alguns destes trabalhos foram pioneiros ao abordar a disseminação de informação via dispositivos móveis quando o *smartphone* não existia como o conhecemos hoje [Klügl and Bazzan 2004]. Outros métodos envolvem comunicação interveicular [Santos and Bazzan 2021], escolha de rota via AR [Bazzan and Grunitzki 2016], e efeito de recomendação de rotas [Ramos et al. 2018].

Por fim, vale lembrar que também é possível utilizar IA em cenários que combinam controle semafórico com gerenciamento da demanda. De fato, esta integração, tão óbvia quanto importante, tem recebido pouca atenção na literatura. No trabalho de [Wiering 2000] foi um dos pioneiros a tratar motoristas e semáforos aprendendo simultaneamente. Em [Lemos et al. 2018] foi proposta uma abordagem baseada em jogos repetidos (para a classe motorista) e jogos estocásticos (para os semáforos). Por se tratar de naturezas diversas de aprendizado, o artigo também discute os desafios encontrados em termos de AR.

8.8.2. Riscos

De modo geral, os riscos de emprego de IA em aplicações na área de mobilidade urbana são similares a outras já discutidas neste capítulo. Entretanto, entre algumas características específicas, destacam-se as seguintes. Em primeiro lugar, a área de controle semafórico tem a segurança como absoluta prioridade. Desta forma, qualquer método de controle, seja ou não baseado em IA, deve fornecer garantias de que a sinalização não resultará em situações que violem os preceitos fundamentais. Em segundo lugar, no que tange questões de comunicação interveicular, é obviamente fundamental garantir não apenas a privacidade dos envolvidos, mas também a segurança geral do sistema (por exemplo contra ataques maliciosos).

8.8.3. Tendências

Esta seção focou apenas nas questões anteriormente mencionadas – a maioria relacionada a tráfego veicular urbano –. Entretanto, além das questões relacionadas a comunicação interveicular, existem pelo menos três áreas nas quais espera-se avanços significativos pelo uso da IA. A primeira está ligada a otimização do uso da rede de recarga de veículos elétricos. A segunda está, obviamente, relacionada com veículos autônomos e, principalmente, a como acomodar frotas mistas (autônomos e convencionais interagindo no mesmo ambiente). Por fim – e em um horizonte mais concreto de tempo – as aplicações de *mobility as a service* já estão maduras e prontas para serem empregadas em conjuntos de políticas públicas visando dar acesso mais eficiente à populações cada vez mais heterogêneas em suas necessidades de mobilidade.

Parte III: Conclusões e Perspectivas

8.9. Visão geral

Este capítulo apresentou uma introdução aos fundamentos de IA e discutiu aplicações, riscos e tendências em suas múltiplas áreas. Destacamos que tal apresentação não é exaustiva. Há muitos outros conceitos envolvendo IA e muitas outras áreas impactadas que apenas tangenciamos. Este capítulo pode ser visto como um ponto de partida, onde o leitor interessado poderá usar as referências apresentadas para se aprofundar nos tópicos de interesse.

Ao longo do capítulo, é possível ver que a aplicação da IA apresenta riscos em comum nas diferentes áreas. Especificamente, sistemas baseados em aprendizado supervisionado, incluindo aprendizado profundo, possuem questões críticas relacionadas à semântica, explicabilidade, transparência (como e porquê determinada saída foi produzida) e vieses (resultados discriminatórios contra determinados grupos de pessoas). Uma interpretação dos modelos de aprendizado é importante do ponto de vista tecnológico (e de produto) para oferecer garantias sobre o comportamento de um sistema. Ademais, entender exatamente porque um determinado sistema apresenta tal comportamento é um requisito básico de qualquer produto tecnológico. Nesse sentido, modelos de aprendizado profundo, embora tenham apresentados resultados tecnológicos relevantes, não apresentam uma semântica rigorosa (isto é, não são modelos que tenham associados uma interpretação lógico-matemática).

Porém, riscos e acidentes não são exclusividade da IA. Todas as novidades tecnológicas da história da humanidade vieram com seus riscos. Como exemplos: junto com a introdução do automóvel vieram os acidentes automobilísticos e com a eletricidade, vieram os riscos de incêndios causados por curto-circuitos e acidentes por descarga elétrica, entre outros riscos que acompanham tecnologias. Uma questão importante é que nessas tecnologias, um acidente ou evento indesejado ocorre quando "algo vai mal", por falha humana, de hardware, de software, entre outras. Por exemplo, um acidente com automóvel ocorre por falha humana ou em algum de seus componentes; um curto-circuito ocorre por sobrecarga na fiação elétrica. Em contraste, sistemas de IA baseados em aprendizado profundo tem a peculiaridade de que um evento indesejado ocorre mesmo quando "tudo vai bem". Mesmo com toda a implementação correta, e sem falhas no hardware, um sistema como o ChatGPT pode produzir saídas incorretas ou prejudiciais, conforme consta no próprio *disclaimer* em sua página inicial[20].

Um grande tópico de pesquisa envolve, portanto, a identificação e mitigação desses riscos associados à IA. Alguns avanços foram feitos no caminho da explicabilidade [Ribeiro et al. 2016, Lundberg and Lee 2017] e na mitigação de vieses e outros riscos de segurança [Thomas et al. 2019]. Um promissor caminho integrador entre a IA baseada em aprendizado (eficiente, mas pouco transparente e por vezes pouco previsível) e a IA simbólica (menos eficiente até o momento, mas transparente, explicável e previsível) é a abordagem neuro-simbólica, cujos estudos visam, entre outros objetivos, oferecer in-

[20] "Prévia de Pesquisa Gratuita. O ChatGPT pode produzir informações imprecisas sobre pessoas, lugares ou fatos. Versão do ChatGPT de 3 de Agosto", conforme acesso em 22/09/2023.

terpretações (ou explicações, se a posteriori) dos métodos e mecanismos de aprendizado atualmente utilizados em IA, conforme discussão a seguir.

8.10. Integração para lidar com os desafios: A IA Neuro-simbólica

Historicamente, a IA iniciou sua trajetória buscando a integração entre diversas habilidades cognitivas, dentre elas, o raciocínio e o aprendizado. Ambas dimensões são vistas como centrais à ideia de inteligência de máquina, já nos trabalhos originais de Turing, von Neumann, McCulloch, Pitts, entre outros [Turing 1950]. von Neumann, em seus trabalhos iniciais, já identificava a relação entre a lógica intuicionista [von Neumann 1956, d'Avila Garcez et al. 2006] e as redes neurais propostas por [McCulloch and Pitts 1943][21].

A área de IA neuro-simbólica integra os dois principais paradigmas da IA: conexionismo (notadamente associado ao uso de redes neurais artificiais como seu modelo principal) e simbolismo (onde o processo de raciocínio em IA é representado através de lógicas, incluindo diversas modalidades como tempo, espaço, conhecimento e incerteza) [d'Avila Garcez et al. 2007, Lamb et al. 2007, d'Avila Garcez and Lamb 2023]. Tradicionalmente, estas áreas foram desenvolvidas por correntes diversas, por terem fundamentos computacionais e lógicos distintos [Besold et al. 2022, d'Avila Garcez et al. 2009]. A área recebeu certa atenção inicial nos anos 1990 e 2000 [Hinton 1990], quando pesquisadores passaram a desenvolver abordagens neuro-simbólicas que aprendessem a realizar inferência lógica clássica, mesmo que para fragmentos de lógicas de predicados [Audibert et al. 2022, d'Avila Garcez and Zaverucha 1999]. À época, foram desenvolvidos sistemas neuro-simbólicos que aprendiam a computar (fragmentos) de programas escritos em linguagens lógicas, como Prolog. Posteriormente, pesquisadores demonstraram que modelos conexionistas poderiam ser treinados para aprender regras de inferência lógica, notadamente sobre lógicas não-clássicas, permitindo a expressão de multi-modalidades [d'Avila Garcez and Lamb 2003, d'Avila Garcez and Lamb 2006, Lamb et al. 2007], antecipando, de certa forma, a pesquisa atual em grandes modelos de linguagens que visa expressar multimodalidades na interação entre usuários humanos e esses sistemas [Kiros et al. 2014].

As grandes contribuições que a área de IA neuro-simbólica pode oferecer são sumarizadas em artigos recentes, publicados na *Communications of the ACM* [Monroe 2022, Hochreiter 2022]. Monroe ressalta a necessidade de desenvolvimento de uma semântica rigorosa para os modelos de IA, como defendido em [d'Avila Garcez and Lamb 2023, Lamb et al. 2020], enquanto Hochreiter aponta que a forma de desenvolver uma IA ampla, que contemple múltiplas habilidades cognitivas, pode ser melhor atingida através da IA neuro-simbólica, sugerindo a abordagem de redes grafos neurais neuro-simbólicos. Hochreiter cita especificamente o trabalho [Lamb et al. 2020] como sendo promissor para esta linha de pesquisa em IA ampla. É relevante ressaltar que esta necessidade de integração neuro-simbólica foi apontada como promissora em eventos recentes, como nas conferências AAAI e NeurIPS, bem como nos debates organizados pela Montreal AI, de-

[21] [von Neumann 1956, Seção 2] afirma que "It has been pointed out by A. M. Turing [5] in 1937 and by W. S. McCulloch and W. Pitts [2] in 1943 that effectively constructive logics, that is, intuitionistic logics, can be best studied in terms of automata. Thus logical propositions can be represented as electrical networks or (idealized) nervous systems."As referências [5] e [2] na citação são, respectivamente, [Turing 1937] e [McCulloch and Pitts 1943].

nominadas de "AI Debates"[22] números 1, 2 e 3. Nestes eventos, foi apontado que para construir sistemas que representem as duas formas de raciocínio - na forma de "AI Fast and Slow"inspirado em [Kahneman 2011] (ver Seção 8.2) - visando a integração de múltiplas habilidades cognitivas em IA, possíveis abordagens promissoras seriam justamente as adotadas na IA neuro-simbólica. Estas abordagens oferecem a possibilidade do desenvolvimento de modelos com fundamentação rigorosa e transparente do ponto de vista lógico, que se integrados ao aprendizado profundo podem levar a tecnologias de inteligência artificial mais robustas, explicáveis e transparentes, oferecendo maior segurança a todos que fazem uso desta tecnologia de propósito geral [d'Avila Garcez and Lamb 2023].

Mais informações sobre a evolução da IA, em particular IA neuro-simbólica podem ser encontradas em [Audibert et al. 2022, d'Avila Garcez et al. 2009, d'Avila Garcez and Lamb 2023].

8.11. Epílogo

Os impactos sociais e éticos da IA levantam questões preponderantes no debate científico e na grande mídia. Atualmente, inúmeras organizações científicas[23] e academias nacionais de ciência têm debatido o impacto da IA na ciência e na sociedade. Além das entidades científicas, organismos multilaterais, historicamente dedicados a temáticas econômicas e sociais[24], constituíram grupos de trabalho sobre o impacto da IA. Muitas vezes este debate é alimentado por não-especialistas, o que demanda cuidados na interpretação do que é publicado. Mesmo com esta ressalva, é muito relevante perceber que nos últimos 10 a 15 anos, com o crescente impacto da IA, particularmente do aprendizado profundo, os cientistas identificaram uma série de limitações e preocupações com o uso da IA sem curadoria - i.e., sem a própria análise de uso por especialistas.

Por fim, deixamos uma demonstração da capacidade utilitária da IA e como ela pode potencializar a capacidade criativa dos seres humanos. Em resposta ao seguinte *prompt*: "escreva um pequeno texto de conclusão para este capítulo de livro "A Nova Eletricidade": Aplicações, Riscos e Tendências da IA Moderna", a auspiciosa última frase do texto gerado pelo ChatGPT foi: "À medida que fechamos este capítulo, é imperativo olhar para o horizonte da IA com olhos críticos e curiosos, prontos para navegar nas águas emocionantes, porém desafiadoras, deste novo mundo alimentado por algoritmos e dados. O futuro da IA está nas mãos daqueles que a guiam com sabedoria e visão."

Cabe aos seres humanos, portanto, a contínua busca por sabedoria e visão para guiar a IA e todas as tecnologias presentes e futuras para o próprio bem da humanidade.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 e CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico. Agradecimentos também

[22] `https://www.quebecartificialintelligence.com/aidebate2/`

[23] e.g. AAAI (Association for the Advancement of Artificial Intelligence), ACM (Association for Computing Machinery), IEEE (Institute of Electrical and Electronic Engineers), e Royal Society, entre outras.

[24] e.g. Fórum Econômico Mundial (WEF), Organização para Cooperação e Desenvolvimento Econômico (OCDE), e as Nações Unidas.

a Cláudio Geyer por comentários no texto e ajuda na revisão.

Referências

[Agrawal et al. 2019] Agrawal, A., Gans, J. S., and Goldfarb, A. (2019). Artificial intelligence: the ambiguous labor market impact of automating prediction. *Journal of Economic Perspectives*, 33(2):31–50.

[Al-Zaiti et al. 2023] Al-Zaiti, S. S., Martin-Gill, C., Zègre-Hemsey, J. K., Bouzid, Z., Faramand, Z., Alrawashdeh, M. O., Gregg, R. E., Helman, S., Riek, N. T., Kraevsky-Phillips, K., et al. (2023). Machine learning for ECG diagnosis and risk stratification of occlusion myocardial infarction. *Nature Medicine*, pages 1–10.

[Aljameel et al. 2022] Aljameel, S. S., Alomari, D. M., Alismail, S., Khawaher, F., Alkhudhair, A. A., Aljubran, F., and Alzannan, R. M. (2022). An anomaly detection model for oil and gas pipelines using machine learning. *Computation*, 10(8):138.

[Aphirakmethawong et al. 2022] Aphirakmethawong, J., Yang, E., and Mehnen, J. (2022). An overview of artificial intelligence in product design for smart manufacturing. In *2022 27th International Conference on Automation and Computing (ICAC)*, pages 1–6. IEEE.

[Audibert et al. 2022] Audibert, R. B., dos Santos, H. L., Avelar, P. H. C., Tavares, A. R., and Lamb, L. C. (2022). On the evolution of A.I. and machine learning: Towards measuring and understanding impact, influence, and leadership at premier A.I. conferences. *arXiv preprint arXiv:2205.13131*.

[Barthélemy and Carletti 2017] Barthélemy, J. and Carletti, T. (2017). A dynamic behavioural traffic assignment model with strategic agents. *Transportation Research Part C: Emerging Technologies*, 85:23–46.

[Baum et al. 2021] Baum, Z. J., Yu, X., Ayala, P. Y., Zhao, Y., Watkins, S. P., and Zhou, Q. (2021). Artificial intelligence in chemistry: current trends and future directions. *Journal of Chemical Information and Modeling*, 61(7):3197–3212.

[Bazzan 2021] Bazzan, A. L. (2021). Contribuições de aprendizado por reforço em escolha de rota e controle semafórico. *Estudos Avançados*, 35(101):95–110.

[Bazzan 2009] Bazzan, A. L. C. (2009). Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multiagent Systems*, 18(3):342–375.

[Bazzan 2022] Bazzan, A. L. C. (2022). Improving urban mobility: using artificial intelligence and new technologies to connect supply and demand. `https://arxiv.org/abs/2204.03570`.

[Bazzan and Grunitzki 2016] Bazzan, A. L. C. and Grunitzki, R. (2016). A multiagent reinforcement learning approach to en-route trip building. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 5288–5295.

[Bazzan and Klügl 2013] Bazzan, A. L. C. and Klügl, F. (2013). *Introduction to Intelligent Systems in Traffic and Transportation*, volume 7 of *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan and Claypool.

[Bender et al. 2021] Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

[Benos et al. 2021] Benos, L., Tagarakis, A. C., Dolias, G., Berruto, R., Kateris, D., and Bochtis, D. (2021). Machine learning in agriculture: A comprehensive updated review. *Sensors*, 21(11):3758.

[Berner et al. 2019] Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. (2019). Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.

[Besold et al. 2022] Besold, T. R., d'Avila Garcez, A. S., Bader, S., Bowman, H., Domingos, P. M., Hitzler, P., Kühnberger, K., Lamb, L. C., Lima, P. M. V., de Penning, L., Pinkas, G., Poon, H., and Zaverucha, G. (2022). Neural-symbolic learning and reasoning: A survey and interpretation. In Hitzler, P. and Sarker, M. K., editors, *Neuro-Symbolic Artificial Intelligence: The State of the Art*, volume 342 of *Frontiers in Artificial Intelligence and Applications*, pages 1–51. IOS Press.

[Birlutiu et al. 2017] Birlutiu, A., Burlacu, A., Kadar, M., and Onita, D. (2017). Defect detection in porcelain industry based on deep learning techniques. In *2017 19th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pages 263–270. IEEE.

[Brachman and Levesque 2004] Brachman, R. J. and Levesque, H. J. (2004). *Knowledge Representation and Reasoning*. Elsevier.

[Broda et al. 2004] Broda, K., Gabbay, D., Lamb, L., and Russo, A. (2004). *Compiled Labelled Deductive Systems: A Uniform Presentation of Non-Classical Logics*. Institute of Physics/Research Studies Press, Hertfordshire.

[Brown et al. 2020] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165*.

[Brownstein et al. 2023] Brownstein, J. S., Rader, B., Astley, C. M., and Tian, H. (2023). Advances in artificial intelligence for infectious-disease surveillance. *New England Journal of Medicine*, 388(17):1597–1607.

[Buolamwini and Gebru 2018] Buolamwini, J. and Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR.

[Colombelli et al. 2022] Colombelli, F., Kowalski, T. W., and Recamonde-Mendoza, M. (2022). A hybrid ensemble feature selection design for candidate biomarkers discovery from transcriptome profiles. *Knowledge-Based Systems*, 254:109655.

[Dalzochio et al. 2020] Dalzochio, J., Kunst, R., Pignaton, E., Binotto, A., Sanyal, S., Favilla, J., and Barbosa, J. (2020). Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges. *Computers in Industry*, 123:103298.

[d'Avila Garcez and Lamb 2006] d'Avila Garcez, A. and Lamb, L. (2006). A connectionist computational model for epistemic and temporal reasoning. *Neur. Computation*, 18(7):1711–1738.

[d'Avila Garcez et al. 2006] d'Avila Garcez, A., Lamb, L., and Gabbay, D. (2006). Connectionist computations of intuitionistic reasoning. *Theor. Comput. Sci.*, 358(1):34–55.

[d'Avila Garcez and Lamb 2023] d'Avila Garcez, A. and Lamb, L. C. (2023). Neurosymbolic AI: The 3rd Wave. *Artificial Intelligence Review*.

[d'Avila Garcez et al. 2007] d'Avila Garcez, A., Lamb, L. C., and Gabbay, D. M. (2007). Connectionist modal logic: Representing modalities in neural networks. *Theor. Comput. Sci.*, 371(1-2):34–53.

[d'Avila Garcez and Zaverucha 1999] d'Avila Garcez, A. and Zaverucha, G. (1999). The connectionist inductive learning and logic programming system. *Applied Intelligence*, 11(1):59–77.

[d'Avila Garcez and Lamb 2003] d'Avila Garcez, A. S. and Lamb, L. C. (2003). Reasoning about Time and Knowledge in Neural-symbolic Learning Systems. In Thrun, S., Saul, L. K., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 921–928. MIT Press.

[d'Avila Garcez et al. 2009] d'Avila Garcez, A. S., Lamb, L. C., and Gabbay, D. M. (2009). *Neural-Symbolic Cognitive Reasoning*. Cognitive Technologies. Springer.

[Devlin et al. 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

[Dia and Panwai 2014] Dia, H. and Panwai, S. (2014). *Intelligent Transport Systems: Neural Agent (Neugent) Models of Driver Behaviour*. LAP Lambert Academic Publishing.

[dos Anjos et al. 2023] dos Anjos, J. C. S., Matteussi, K. J., Orlandi, F. C., Barbosa, J. L. V., Silva, J. S., Bittencourt, L. F., and Geyer, C. F. R. (2023). A Survey on Collaborative Learning for Intelligent Autonomous Systems. *ACM Comput. Surv.*, 1(1):1–36.

[Dosovitskiy et al. 2020] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[Eykholt et al. 2018] Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. (2018). Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634.

[Fagin et al. 1995] Fagin, R., Halpern, J. Y., Moses, Y., and Vardi, M. Y. (1995). *Reasoning About Knowledge*. MIT Press.

[Fournier-Viger et al. 2021] Fournier-Viger, P., Nawaz, M. S., Song, W., and Gan, W. (2021). Machine learning for intelligent industrial design. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 158–172. Springer.

[Fragapane et al. 2022] Fragapane, G., Ivanov, D., Peron, M., Sgarbossa, F., and Strandhagen, J. O. (2022). Increasing flexibility and productivity in industry 4.0 production networks with autonomous mobile robots and smart intralogistics. *Annals of Operations Research*, 308(1-2):125–143.

[Franklin et al. 2020] Franklin, C. S., Dominguez, E. G., Fryman, J. D., and Lewandowski, M. L. (2020). Collaborative robotics: New era of human–robot cooperation in the workplace. *Journal of Safety Research*, 74:153–160.

[Geffner 2018] Geffner, H. (2018). Model-free, model-based, and general intelligence. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI2018*.

[Giri et al. 2019] Giri, C., Jain, S., Zeng, X., and Bruniaux, P. (2019). A detailed review of artificial intelligence applied in the fashion and apparel industry. *IEEE Access*, 7:95376–95396.

[Goodfellow et al. 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[Grisoni et al. 2021] Grisoni, F., Huisman, B. J., Button, A. L., Moret, M., Atz, K., Merk, D., and Schneider, G. (2021). Combining generative artificial intelligence and on-chip synthesis for de novo drug design. *Science Advances*, 7(24):eabg3338.

[Gubbi et al. 2022] Gubbi, K. I., Beheshti-Shirazi, S. A., Sheaves, T., Salehi, S., PD, S. M., Rafatirad, S., Sasan, A., and Homayoun, H. (2022). Survey of machine learning for electronic design automation. In *Proceedings of the Great Lakes Symposium on VLSI 2022*, pages 513–518.

[Hamolia and Melnyk 2021] Hamolia, V. and Melnyk, V. (2021). A survey of machine learning methods and applications in electronic design automation. In *2021 11th International Conference on Advanced Computer Information Technologies (ACIT)*, pages 757–760. IEEE.

[Hart et al. 1968] Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

[Hasan et al. 2021] Hasan, I., Liao, S., Li, J., Akram, S. U., and Shao, L. (2021). Generalizable pedestrian detection: The elephant in the room. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11328–11337.

[Helmert and Domshlak 2009] Helmert, M. and Domshlak, C. (2009). Landmarks, critical paths and abstractions: What's the difference anyway? In *International Conference on Automated Planning and Scheduling*, pages 162–169.

[Hinton 1990] Hinton, G. (1990). Connectionist symbol processing - preface. *Artif. Intell.*, 46(1-2):1–4.

[Hinton et al. 2006] Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554.

[Hochreiter 2022] Hochreiter, S. (2022). Toward a broad AI. *Communications of the ACM*, 65(4):56–57.

[Hochreiter and Schmidhuber 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[Hoffmann 2011] Hoffmann, J. (2011). Everything you always wanted to know about planning: (but were afraid to ask). In *Advances in Artificial Intelligence*, pages 1–13.

[Huang et al. 2019] Huang, P., Lin, C. T., Li, Y., Tammemagi, M. C., Brock, M. V., Atkar-Khattra, S., Xu, Y., Hu, P., Mayo, J. R., Schmidt, H., et al. (2019). Prediction of lung cancer risk at follow-up screening with low-dose CT: a training and validation study of a deep learning method. *The Lancet Digital Health*, 1(7):e353–e362.

[Ignat et al. 2023] Ignat, O., Jin, Z., Abzaliev, A., Biester, L., Castro, S., Deng, N., Gao, X., Gunal, A., He, J., Kazemi, A., Khalifa, M., Koh, N., Lee, A., Liu, S., Min, D. J., Mori, S., Nwatu, J., Perez-Rosas, V., Shen, S., Wang, Z., Wu, W., and Mihalcea, R. (2023). A PhD student's perspective on research in NLP in the era of very large language models.

[Jiang et al. 2018] Jiang, D., Hao, M., Ding, F., Fu, J., and Li, M. (2018). Mapping the transmission risk of zika virus using machine learning models. *Acta Tropica*, 185:391–399.

[Jumper et al. 2021] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589.

[Kahneman 2011] Kahneman, D. (2011). *Thinking, Fast and Slow*. Farrar, Straus and Giroux.

[Kielarova and Pradujphongphet 2023] Kielarova, S. W. and Pradujphongphet, P. (2023). Genetic algorithm for product design optimization: An industrial case study of halo setting for jewelry design. In *International Conference on Swarm Intelligence*, pages 219–228. Springer.

[Kiros et al. 2014] Kiros, R., Salakhutdinov, R., and Zemel, R. (2014). Multimodal neural language models. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–595–II–603. JMLR.org.

[Klügl and Bazzan 2004] Klügl, F. and Bazzan, A. L. C. (2004). Route decision behaviour in a commuting scenario. *Journal of Artificial Societies and Social Simulation*, 7(1).

[Kowalski 1979] Kowalski, R. A. (1979). *Logic for problem solving*. North-Holland.

[Lamb et al. 2007] Lamb, L., Borges, R., and d'Avila Garcez, A. (2007). A connectionist cognitive model for temporal synchronisation and learning. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 1*, AAAI'07, page 827–832.

[Lamb et al. 2020] Lamb, L. C., d'Avila Garcez, A. S., Gori, M., Prates, M. O. R., Avelar, P. H. C., and Vardi, M. Y. (2020). Graph neural networks meet neural-symbolic computing: A survey and perspective. In Bessiere, C., editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4877–4884. ijcai.org.

[LeCun et al. 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551.

[Lemos et al. 2018] Lemos, L. L., Bazzan, A. L. C., and Pasin, M. (2018). Co-adaptive reinforcement learning in microscopic traffic systems. In *2018 IEEE Congress on Evolutionary Computation, CEC 2018, Rio de Janeiro, Brazil, July 8-13, 2018*, pages 1–8.

[Li et al. 2017] Li, B.-h., Hou, B.-c., Yu, W.-t., Lu, X.-b., and Yang, C.-w. (2017). Applications of artificial intelligence in intelligent manufacturing: a review. *Frontiers of Information Technology & Electronic Engineering*, 18:86–96.

[Liang et al. 2020] Liang, Y., Lee, S.-H., and Workman, J. E. (2020). Implementation of artificial intelligence in fashion: Are consumers ready? *Clothing and Textiles Research Journal*, 38(1):3–18.

[Liu et al. 2019] Liu, X., Faes, L., Kale, A. U., Wagner, S. K., Fu, D. J., Bruynseels, A., Mahendiran, T., Moraes, G., Shamdas, M., Kern, C., et al. (2019). A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. *The Lancet Digital Health*, 1(6):e271–e297.

[Lundberg and Lee 2017] Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

[Lynch 2017] Lynch, S. (2017). Andrew Ng: Why AI Is the New Electricity. `https://www.gsb.stanford.edu/insights/andrew-ng-why-ai-new-electricity`. Acesso em 15/09/2023.

[Mahendran and PM 2022] Mahendran, N. and PM, D. R. V. (2022). A deep learning framework with an embedded-based feature selection approach for the early detection of the Alzheimer's disease. *Computers in Biology and Medicine*, 141:105056.

[Marczyk et al. 2023] Marczyk, V. R., Recamonde-Mendoza, M., Maia, A. L., and Goemann, I. M. (2023). Classification of thyroid tumors based on DNA methylation patterns. *Thyroid*, 33(9):1090–1099.

[Martí et al. 2015] Martí, L., Sanchez-Pi, N., Molina, J. M., and Garcia, A. C. B. (2015). Anomaly detection based on sensor data in petroleum industry applications. *Sensors*, 15(2):2774–2797.

[Martin et al. 2019] Martin, A. R., Kanai, M., Kamatani, Y., Okada, Y., Neale, B. M., and Daly, M. J. (2019). Clinical use of current polygenic risk scores may exacerbate health disparities. *Nature Genetics*, 51(4):584–591.

[McCulloch and Pitts 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133.

[Mikolov et al. 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mitchell 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

[Mnih et al. 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

[Monroe 2022] Monroe, D. (2022). Neurosymbolic AI. *Communications of the ACM*, 65(10):11–13.

[Narodytska and Kasiviswanathan 2017] Narodytska, N. and Kasiviswanathan, S. P. (2017). Simple black-box adversarial attacks on deep neural networks. In *CVPR Workshops*, volume 2, page 2.

[Noaeen et al. 2022] Noaeen, M., Naik, A., Goodman, L., Crebo, J., Abrar, T., Abad, Z. S. H., Bazzan, A. L., and Far, B. (2022). Reinforcement learning in urban network traffic signal control: A systematic literature review. *Expert Systems with Applications*, page 116830.

[Obermeyer et al. 2019] Obermeyer, Z., Powers, B., Vogeli, C., and Mullainathan, S. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453.

[Ouyang et al. 2022] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. (2022). Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

[Panch et al. 2019] Panch, T., Pearson-Stuttard, J., Greaves, F., and Atun, R. (2019). Artificial intelligence: opportunities and risks for public health. *The Lancet Digital Health*, 1(1):e13–e14.

[Paolanti et al. 2018] Paolanti, M., Romeo, L., Felicetti, A., Mancini, A., Frontoni, E., and Loncarski, J. (2018). Machine learning approach for predictive maintenance in industry 4.0. In *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pages 1–6. IEEE.

[Papakyriakopoulos et al. 2020] Papakyriakopoulos, O., Hegelich, S., Serrano, J. C. M., and Marco, F. (2020). Bias in word embeddings. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 446–457. Association for Computing Machinery.

[Phakhounthong et al. 2018] Phakhounthong, K., Chaovalit, P., Jittamala, P., Blacksell, S. D., Carter, M. J., Turner, P., Chheng, K., Sona, S., Kumar, V., Day, N. P., et al. (2018). Predicting the severity of dengue fever in children on admission based on clinical features and laboratory indicators: application of classification tree analysis. *BMC Pediatrics*, 18:1–9.

[Pivetta et al. 2023] Pivetta, M. V. L., Simon, A. H., Costa, M. M., Abel, M., and Carbonera, J. L. (2023). A systematic evaluation of machine learning approaches for petroleum production forecasting. In *IEEE 35th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 768–774. IEEE.

[Prates et al. 2020] Prates, M. O., Avelar, P. H., and Lamb, L. C. (2020). Assessing gender bias in machine translation: a case study with google translate. *Neural Computing and Applications*, 32:6363–6381.

[Radford et al. 2021] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR. ISSN: 2640-3498.

[Rahmanifard and Plaksina 2019] Rahmanifard, H. and Plaksina, T. (2019). Application of artificial intelligence techniques in the petroleum industry: a review. *Artificial Intelligence Review*, 52(4):2295–2318.

[Rajpurkar et al. 2022] Rajpurkar, P., Chen, E., Banerjee, O., and Topol, E. J. (2022). AI in health and medicine. *Nature Medicine*, 28(1):31–38.

[Ramos et al. 2018] Ramos, G. de. O., Bazzan, A. L. C., and da Silva, B. C. (2018). Analysing the impact of travel information for minimising the regret of route choice. *Transportation Research Part C: Emerging Technologies*, 88:257–271.

[Ribeiro et al. 2021] Ribeiro, J., Lima, R., Eckhardt, T., and Paiva, S. (2021). Robotic process automation and artificial intelligence in industry 4.0–a literature review. *Procedia Computer Science*, 181:51–58.

[Ribeiro et al. 2016] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144.

[Roth et al. 2022] Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., and Gehler, P. (2022). Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328.

[Rumelhart et al. 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

[Russell et al. 2015] Russell, S., Hauert, S., Altman, R., and Veloso, M. (2015). Ethics of artificial intelligence: Four leading researchers share their concerns and solutions for reducing societal risks from intelligent machines. *Nature*, 521:415–418.

[Russell and Norvig 2020] Russell, S. and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson.

[Santos and Bazzan 2021] Santos, G. D. dos. and Bazzan, A. L. C. (2021). Sharing diverse information gets driver agents to learn faster: an application in en route trip building. *PeerJ Computer Science*, 7:e428.

[Schuster and Paliwal 1997] Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

[Schwalbe and Wahl 2020] Schwalbe, N. and Wahl, B. (2020). Artificial intelligence and the future of global health. *The Lancet*, 395(10236):1579–1586.

[Serradilla et al. 2022] Serradilla, O., Zugasti, E., Rodriguez, J., and Zurutuza, U. (2022). Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects. *Applied Intelligence*, 52(10):10934–10964.

[Sharir et al. 2020] Sharir, O., Peleg, B., and Shoham, Y. (2020). The cost of training nlp models: A concise overview. *arXiv preprint arXiv:2004.08900*.

[Silver et al. 2017a] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2017a). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*.

[Silver et al. 2017b] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. (2017b). Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359.

[Stojanovic et al. 2016] Stojanovic, L., Dinic, M., Stojanovic, N., and Stojadinovic, A. (2016). Big-data-driven anomaly detection in industry (4.0): An approach and a case study. In *2016 IEEE International Conference on Big Data*, pages 1647–1652. IEEE.

[Sutton and Barto 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. The MIT Press, second edition.

[Thomas et al. 2019] Thomas, P. S., Castro da Silva, B., Barto, A. G., Giguere, S., Brun, Y., and Brunskill, E. (2019). Preventing undesirable behavior of intelligent machines. *Science*, 366(6468):999–1004.

[Toorajipour et al. 2021] Toorajipour, R., Sohrabpour, V., Nazarpour, A., Oghazi, P., and Fischl, M. (2021). Artificial intelligence in supply chain management: A systematic literature review. *Journal of Business Research*, 122:502–517.

[Turing 1937] Turing, A. M. (1937). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265.

[Turing 1950] Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236):433–460.

[Van der Schaar et al. 2021] Van der Schaar, M., Alaa, A. M., Floto, A., Gimson, A., Scholtes, S., Wood, A., McKinney, E., Jarrett, D., Lio, P., and Ercole, A. (2021). How artificial intelligence and machine learning can help healthcare systems respond to COVID-19. *Machine Learning*, 110:1–14.

[Vaswani et al. 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

[Vinyals et al. 2019] Vinyals, O., Babuschkin, I., Czarnecki, W., Mathieu, M., Dudzik, A., Chung, J., Choi, D., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J., Jaderberg, M., and Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575:350—-354.

[von Neumann 1956] von Neumann, J. (1956). Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Automata Studies*, 34:43–98.

[Wang and Luo 2019] Wang, L. and Luo, M. (2019). Machine learning applications and opportunities in ic design flow. In *2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, pages 1–3. IEEE.

[Warren et al. 2023] Warren, D. S., Dahl, V., Eiter, T., Hermenegildo, M. V., Kowalski, R. A., and Rossi, F., editors (2023). *Prolog: The Next 50 Years*, volume 13900 of *Lecture Notes in Computer Science*. Springer.

[Watkins 1989] Watkins, C. (1989). *Learning from Delayed Rewards*. PhD thesis, University of Cambridge.

[Watkins and Dayan 1992] Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3):279–292.

[Wei et al. 2019] Wei, H., Li, Z., Xu, N., Zhang, H., Zheng, G., Zang, X., Chen, C., Zhang, W., Zhu, Y., and Xu, K. (2019). Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1913–1922. Association for Computing Machinery.

[Wiering 2000] Wiering, M. (2000). Multi-agent reinforcement learning for traffic light control. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, pages 1151–1158.

[Wong et al. 2021] Wong, A., Otles, E., Donnelly, J. P., Krumm, A., McCullough, J., DeTroyer-Cooley, O., Pestrue, J., Phillips, M., Konye, J., Penoza, C., et al. (2021). External validation of a widely implemented proprietary sepsis prediction model in hospitalized patients. *JAMA Internal Medicine*, 181(8):1065–1070.

[World Health Organization 2021] World Health Organization (2021). Ethics and governance of artificial intelligence for health: WHO guidance.

[Xu et al. 2019] Xu, J., Yang, P., Xue, S., Sharma, B., Sanchez-Martin, M., Wang, F., Beaty, K. A., Dehan, E., and Parikh, B. (2019). Translating cancer genomics into precision medicine with artificial intelligence: applications, challenges and future perspectives. *Human Genetics*, 138(2):109–124.

[Yau et al. 2017] Yau, K.-L. A., Qadir, J., Khoo, H. L., Ling, M. H., and Komisarczuk, P. (2017). A survey on reinforcement learning models and algorithms for traffic signal control. *ACM Comput. Surv.*, 50(3).

[Yu et al. 2018] Yu, K.-H., Beam, A. L., and Kohane, I. S. (2018). Artificial intelligence in healthcare. *Nature Biomedical Engineering*, 2(10):719–731.

[Zhang et al. 2019] Zhang, X., Zhou, T., Zhang, L., Fung, K. Y., and Ng, K. M. (2019). Food product design: a hybrid machine learning and mechanistic modeling approach. *Industrial & Engineering Chemistry Research*, 58(36):16743–16752.

[Zhu et al. 2021] Zhu, X., Ninh, A., Zhao, H., and Liu, Z. (2021). Demand forecasting with supply-chain information and machine learning: Evidence in the pharmaceutical industry. *Production and Operations Management*, 30(9):3231–3252.

[Zipfel et al. 2023] Zipfel, J., Verworner, F., Fischer, M., Wieland, U., Kraus, M., and Zschech, P. (2023). Anomaly detection for industrial quality assurance: A comparative evaluation of unsupervised deep learning models. *Computers & Industrial Engineering*, 177:109045.