Capítulo 1

Ciência de Dados Aplicada à Cibersegurança: Teoria e Prática

Michele Nogueira (UFMG), Ligia Francielle Borges (UFMG), Anderson Begamini de Neira (UFPR), Lucas Albano Olive Cruz (UFMG), Kristtopher Kayo Coelho (UFV)

Abstract

The popularization of computational devices and communication technologies highlights new security threats. The deployment of the Internet of Things expands the attack surface, generating new threats due to intrinsic security vulnerabilities. Data Science and Artificial Intelligence have emerged as powerful tools. They offer new possibilities for Cybersecurity, including the analysis of a large volume of data, identifying and predicting vulnerabilities, and detecting intrusions. This chapter presents the concepts, methodology, and techniques of Data Science for Cybersecurity, with multiple goals: (i) to disseminate the culture of Data Science for Cybersecurity; (ii) to demonstrate the potential of Artificial Intelligence and Machine Learning techniques for this area; (iii) to encourage the collaboration between research groups in Brazil and our group at UFMG and UFPR; and (iv) to share some of the results achieved in the MCTI/FAPESP MENTORED project. The chapter concludes with a description of a demonstration from the entire Data Science pipeline in Cybersecurity and a discussion of future perspectives, challenges, and open research questions.

Resumo

A popularização dos dispositivos computacionais e das tecnologias de comunicação evidencia novas ameaças de segurança. A implantação da Internet das Coisas amplia a superfície de ataques, gerando mais ameaças diante das vulnerabilidades intrínsecas de segurança. A Ciência de Dados e a Inteligência Artificial emergem como ferramentas poderosas. Elas oferecem novas possibilidades para a Cibersegurança, incluindo a análise de grandes volumes de dados, a identificação e a predição de vulnerabilidades, além da detecção de intrusos. Este capítulo apresenta os conceitos, a metodologia e as técnicas

da Ciência de Dados para Cibersegurança, com múltiplos objetivos: (i) disseminar a cultura da Ciência de Dados na Cibersegurança; (ii) demonstrar o potencial das técnicas de Inteligência Artificial e Aprendizado de Máquina para essa área; (iii) incentivar colaborações entre grupos de pesquisa no Brasil e o nosso na UFMG e UFPR; e (iv) compartilhar resultados do projeto MCTI/FAPESP MENTORED. O capítulo descreve a demonstração de todo o pipeline da Ciência de Dados na Cibersegurança, além de discutir perspectivas futuras, desafios e questões de pesquisa em aberto.

1.1. Introdução

O mundo está cada vez mais interconectado. O ciberespaço, que integra tecnologia, dados e pessoas, tornou-se essencial para a realização de nossas atividades como sociedade, abrangendo não apenas usuários individuais, mas também instituições e governos. Os usuários têm acesso a uma gama cada vez maior de aplicativos e serviços disponíveis na Internet, proporcionando múltiplas formas de trabalho, educação, entretenimento e governança. Desde a pandemia do novo coronavírus, esse cenário se intensificou, com a conectividade impulsionando o aprendizado remoto, a telemedicina e o teletrabalho. Esses novos hábitos acentuaram a diversidade nas formas de acesso aos sistemas e, consequentemente, um aumento no número e na heterogeneidade de vulnerabilidades de segurança cibernética. Isso demanda uma busca incessante por soluções e por profissionais de cibersegurança qualificados, capazes de atuar em diferentes níveis da área, desde a segurança de *hardware* e *software* até a análise e o tratamento de dados.

Apesar dos benefícios da conectividade para nossos sistemas, as instituições são forçadas a mudar suas estratégias e prioridades para evitar fraudes, prejuízos financeiros e comprometimentos na reputação institucional. Por um lado, a cibersegurança ultrapassou aspectos meramente técnicos, exigindo hoje um forte alinhamento com outras vertentes, como governança, economia, ética e a visão das organizações e governos [Jurgens and Cin 2024]. As empresas investem cada vez mais em tecnologias avançadas. Os governos estão implementando regulamentações mais rígidas para proteger infraestruturas críticas e dados pessoais. A cibersegurança, portanto, não é mais apenas uma questão técnica, mas uma prioridade estratégica que envolve políticas públicas, educação e conscientização contínua de todos os usuários da rede. Por outro lado, observamos um aumento significativo na escala de ameaças digitais, tornando complexa a gestão desse ecossistema e, portanto, exigindo soluções inovadoras. Para ilustrar sua importância, é crucial destacar que o cibercrime causou uma perda empresarial global de um bilhão de dólares só em 2020. Até 2025, estima-se que os prejuízos com o cibercrime em todo o planeta atinjam US$ 10,5 trilhões por ano, segundo a *Cybersecurity Ventures*[1], empresa que pesquisa dados sobre a economia cibernética global.

Na era digital, a cibersegurança é uma prioridade. A proliferação de dispositivos conectados na Internet das Coisas (do inglês, *Internet of Things – IoT*) e das tecnologias de comunicação facilita a oferta de uma vasta gama de serviços online, o que, por sua vez, aumenta exponencialmente a quantidade de dados gerados e a dependência de indivíduos

[1]Relatório Oficial sobre Cibercrimes da Ventures 2023: https://www.esentire.com/resources/library/2023-official-cybercrime-report. Último acesso: 30 de Abril de 2024.

e empresas nas tecnologias digitais. Estima-se que, nos últimos dois anos, 90% dos dados atuais foram produzidos, abrangendo desde informações coletadas por dispositivos em rede, como câmeras e sensores, até conteúdos em mídias sociais e transações online. Esse aumento exponencial de dados e a consequente dependência digital reforçam o desafio crítico de garantir a segurança e a integridade dos sistemas, assim como a privacidade dos dados. Os ataques cibernéticos são ameaças ativas à disponibilidade, confidencialidade, integridade, autenticidade e não-repúdio, que são atributos fundamentais da cibersegurança. Esses ataques assumem diversas formas, incluindo DDoS (do inglês, *Distributed Denial of Service*), que visam desabilitar ou interromper o acesso a serviços e dados, impactando fortemente sistemas e serviços de empresas, governos e indivíduos. Os ataques que alteram dados, sejam eles armazenados ou em trânsito, representam uma séria ameaça à integridade das informações, levando a consequências graves, como a manipulação de resultados financeiros, violação de privacidade e comprometimento de dados sensíveis.

Neste contexto, a integração entre ciência de dados, inteligência artificial – IA (do inglês, *Artificial Intelligence – AI*) e aprendizado de máquina – AM (do inglês, *Machine Learning – ML*) torna-se vital para a cibersegurança. Essas tecnologias permitem a análise de grandes volumes de dados, identificando padrões e anomalias e apontando possíveis ameaças. Por exemplo, os algoritmos de ML são treinados para detectar comportamentos suspeitos, bloqueando automaticamente tentativas de ataque antes que causem danos significativos. O vasto volume de dados disponível, combinado com avanços no poder computacional e nas técnicas de aprendizado de máquina, está transformando radicalmente a segurança cibernética. Hoje, é possível prever ataques cibernéticos com maior precisão, melhorar a resposta a incidentes e implementar defesas mais robustas. Além disso, a colaboração entre diferentes setores e a troca de informações sobre ameaças cibernéticas são essenciais para desenvolver uma postura de segurança mais proativa.

A aplicação da ciência de dados à cibersegurança segue duas perspectivas principais: (1) defensiva em que o uso de técnicas de ciência de dados apoia no desenvolvimento de sistemas de proteção e prevenção, como a detecção de *malware* e de ataques de rede; e (2) ofensiva em que o acesso aos dados e suas análises servem de base para o projeto e realização de ataques mais sofisticados. Este capítulo se concentra na primeira perspectiva, destacando como a ciência de dados e a IA são utilizadas para criar defesas mais eficazes. A ciência de dados e, particularmente, a sua intersecção com a IA e o aprendizado de máquina estão revolucionando a segurança cibernética ao permitir a análise de grandes volumes de dados, a identificação de padrões e a realização de previsões com precisão e rapidez superiores às capacidades humanas. Essas tecnologias são fundamentais para detectar e mitigar ameaças cibernéticas em tempo real, melhorando a segurança organizacional. Por exemplo, a IA auxilia na identificação de padrões anômalos indicativos de ataques cibernéticos nos estágios iniciais, crucial para enfrentar métodos sofisticados como *malware* sem arquivo. Além disso, a IA auxilia no desenvolvimento de sistemas de resposta automática a incidentes e ferramentas de análise preditiva que antecipam vulnerabilidades. Ela também é capaz de simular cenários de ataque para treinar profissionais, promovendo uma cultura de segurança proativa.

A academia possui um histórico significativo de integração entre inteligência artificial e segurança cibernética. Diversas técnicas de IA, incluindo redes neurais, algoritmos genéticos e aprendizado de máquina, são aplicadas em sistemas de detecção de intrusão,

análise de tráfego de rede para detecção de anomalias, e na classificação de e-mails e identificação de SPAMs. Muitas empresas estão aproveitando as vantagens da IA para fortalecer suas defesas de segurança cibernética. Por exemplo, a plataforma Reveal(x) da ExtraHop realiza uma análise detalhada baseada em regras e comportamentos, fornecendo *insights* sobre o tráfego de rede e identificando potenciais ameaças. Da mesma forma, o Vectra Cognito utiliza inteligência artificial para detectar ameaças futuras ou desconhecidas através da análise de cargas de trabalho. Este capítulo extrapola a visão limitada a IA aplicada à cibersegurança e vai além. Este capítulo apresenta os conceitos, a metodologia e as técnicas da ciência de dados para cibersegurança, com múltiplos objetivos: (i) disseminar a cultura da ciência de dados na cibersegurança; (ii) demonstrar o potencial das técnicas de IA e AM para essa área; (iii) incentivar colaborações entre outros grupos de pesquisa no Brasil e o nosso na UFMG e UFPR; e (iv) demonstrar resultados alcançados no projeto MCTI/FAPESP MENTORED relacionados ao tema. O capítulo conclui com a descrição de um *pipeline* de ciência de dados aplicado na cibersegurança, além de discutir perspectivas futuras, desafios e questões de pesquisa em aberto.

Este capítulo é o material complementar ao minicurso de mesmo título, a ser ministrado em conjunto com o Simpósio Brasileiro de Sistemas de Computadores e de Sistemas Computacionais (SBSeg) 2024. Ele instiga os participantes a conhecer o tema, de forma teórica e prática. Será apresentado como a ciência de dados automatiza processos e aprimora a detecção e resposta a ameaças, passando pela análise do comportamento de usuários e sistemas para identificar atividades suspeitas ou anomalias indicativas de ataques e identificação de vulnerabilidades.

A organização do capítulo segue. A Seção 1.2 apresenta os conceitos básicos relacionados à cibersegurança, ciência de dados, inteligência artificial e aprendizado de máquina. A Seção 1.3 faz uma visão geral do estado da arte. A Seção 1.4 descreve os principais ambientes experimentais, ferramentas e bases de dados usadas na área. A Seção 1.5 descreve o estudo de caso prático a ser apresentado durante o evento. Finalmente, a Seção 1.6 apresenta os principais desafios e a Seção 1.7 discute as conclusões.

1.2. Conceitos Básicos

Essa seção apresenta os conceitos e definições relevantes relacionados à cibersegurança e ciência de dados. Na subseção 'cibersegurança' (Subseção 1.2.1), são discutidos os princípios fundamentais, incluindo os conceitos de confidencialidade, integridade, disponibilidade, autenticidade e não-repúdio, além das definições de ameaças cibernéticas, vetores de ataques, vulnerabilidades e *exploits*. Na subseção 'ciência de dados' (Subseção 1.2.2), apresentamos os conceitos relacionados a dados, conhecimento, etapas no *pipeline* de ciência de dados, diferença entre os conceitos de ciência de dados, IA, ML, análise preditiva e outros. Detalharemos como essas técnicas extraem *insights* de grandes conjuntos de dados. Especificamente, são discutidos os principais estágios, desde a coleta de dados até a análise e interpretação. Também são exploradas as tarefas de AM aplicadas à segurança cibernética, incluindo o uso de redes neurais, aprendizagem profunda, modelos *Transformers* e *Large Language Models* para detectar e mitigar ameaças.

1.2.1. Cibersegurança

O termo "cibersegurança" vem ganhando ampla popularidade nos últimos anos. O termo é usado por pessoas e áreas diferentes com significados diferentes em contextos diversos. Durante muitos anos, a cibersegurança teve como foco principal a informação. Nada mais justo, pois até então o interesse maior era proteger a informação, ainda mais considerando o legado das grandes guerras mundiais, guerra fria e outros. Estamos falando de uma época em que a Internet nem existia e, portanto, essa hiper conectividade existente atualmente não era um fator de preocupação. Assim, a cibersegurança era definida com base nos principais atributos para a segurança da informação, compondo a famosa tríade CIA, em inglês, ***C****onfidentiality*, ***I****ntegrity* e ***A****vailability*, conhecida em português como CID: **C**onfidencialidade, **I**ntegridade e **D**isponibilidade [Anderson et al. 1972]. Com as mudanças nos nossos sistemas, a conexão dos nossos dispositivos, a geração de dados e impacto da cibersegurança nos negócios, empresas, governos e nações, essa definição é considerada limitada e requer um foco maior nas atividades e riscos.

Este capítulo e trabalho seguem uma visão de cibersegurança como uma ciência. A ciência é uma ferramenta poderosa através da qual nós humanos conseguimos gerar avanços tecnológicos e sociais consideráveis através da aplicação rigorosa do método científico. A ciência representa filosofia, conhecimento e processo. A cibersegurança é um campo recente de pesquisa. Os sistemas digitais datam de menos de 100 anos. As redes de computadores não tem nem 50 anos. Os grandes problemas de segurança não haviam sido observados até as décadas de 1980 e 1990, com o aparecimento e evolução da Internet. Nesses últimos anos, os avanços e a complexidade do ciberespaço cresceram exponencialmente. Brevemente, a cibersegurança é definida como a ciência que estuda e propõe soluções para tornar o ciberespaço seguro contra danos e ameaças, sendo o ciberespaço a integração entre dados, tecnologia e pessoas.

Essa visão mais moderna da cibersegurança engloba diferentes áreas, desde a visão mais técnica, que se expandiu ao longo dos anos com a evolução dos nossos sistemas, até visões voltadas às pessoas, incluindo direito, economia, psicologia, governo, ciências sociais, políticas e outras. Essa visão mais ampla expande as perspectivas da cibersegurança que na definição baseada na tríade CIA trazia limitações por sugerir medidas absolutas, onde um ativo era considerado seguro ou não, o que cria uma falsa sensação de realização, desconsiderando a natureza contínua dos riscos de segurança. Além disso, o foco da tríade incentiva a proteção de ativos individuais em vez de uma abordagem mais ampla e contextual, resultando em soluções temporárias que não abordam o panorama de riscos [Ham 2021]. Para Ham et al. 2021, a tríade não leva em conta o contexto em que os ativos operam, tratando a confidencialidade como uma propriedade isolada, o que pode levar a medidas de segurança ineficazes. Com a evolução da infraestrutura digital e a mudança na natureza das ameaças, é necessário adotar uma abordagem mais dinâmica e contínua para a cibersegurança, em vez de depender de um modelo estático CIA [Lipner and Anderson 2018]. Assim, embora a tríade tenha sido fundamental na cibersegurança, ele não deve mais ser vista como o objetivo final, mas sim como parte de uma atividade contínua que se adapta às mudanças no contexto e nos riscos.

O comportamento dos adversários cibernéticos evoluiu significativamente [de Neira et al. 2020]. Os atacantes agora empregam técnicas avançadas, como engenharia social,

zero day exploits (*i.e.*, falha ou vulnerabilidade explorada para criar e liberar ameaças antes que os desenvolvedores tenham tempo de criar um pacote para corrigir a vulnerabilidade) e ataques coordenados, tornando a detecção e a resposta mais desafiadoras. Além disso, as motivações por trás dos ataques cibernéticos se diversificaram, abrangendo não apenas ganhos financeiros, mas também objetivos políticos, ideológicos e de espionagem. Essa mudança requer uma compreensão mais abrangente da cibersegurança. A cibersegurança moderna vai além das defesas tradicionais, incorporando estratégias proativas e adaptativas. Essa evolução inclui a integração da ciência de dados, a inteligência artificial e aprendizado de máquina para detecção de ameaças em tempo real e resposta automatizada [Brito et al. 2023]. A cibersegurança contemporânea abrange não apenas soluções tecnológicas, mas também fatores humanos, políticas e estruturas de governança [Nogueira et al. 2021].

Neste sentido, Ham 2021 propôs uma nova perspectiva sobre a cibersegurança, destacando que, após mais de 40 anos de prática, há uma compreensão mais profunda dos riscos associados. O autor argumenta que a cibersegurança deve ser vista como uma atividade contínua, envolvendo a identificação de ativos, avaliação de riscos e adaptação a novas ameaças. O Framework de Cibersegurança do Instituto Nacional de Padrões e Tecnologia (NIST) exemplifica essa abordagem abrangente, incluindo atividades como identificar, proteger, detectar, responder e recuperar, seguindo uma abordagem cíclica e de continuidade. Essa nova visão promove uma avaliação contínua dos riscos e uma resposta mais eficaz às mudanças no cenário de ameaças. O NIST desenvolveu o *Cybersecurity Framework* (CSF) baseado em padrões, diretrizes e práticas existentes para avaliar e gerenciar riscos cibernéticos. Originalmente, o CSF foi desenvolvido em resposta a ordem executiva 13636, que indicava a melhora da segurança cibernética da infraestrutura crítica dos Estados Unidos. O CSF foi formalizado a partir da aprovação da Lei de Aprimoramento da Segurança Cibernética (do inglês, *Cybersecurity Enhancement Act* - CEA) de 2014, em que ficou definido que o NIST deveria desenvolver um arcabouço de segurança flexível, repetível, eficaz com uma boa relação de custo-benefício. Esta lei contribuiu para o contínuo desenvolvimento do CSF, além de fornecer direções futuras [NIST 2018].

A Figura 1.1 organiza as funções do CSF do NIST em um círculo, evidenciando que elas não formam um caminho sequencial ou um estado final estático, mas são realizadas simultânea e continuamente, criando uma cultura operacional adaptável aos riscos de segurança cibernética [NIST 2018]. A função de identificação desenvolve um entendimento organizacional para gerenciar riscos de segurança cibernética, compreendendo o contexto de negócios, recursos críticos e riscos associados. A função de proteção desenvolve e implementa soluções para garantir a entrega de serviços e limitar o impacto de ameaças cibernéticas. A detecção implementa atividades para identificar eventos de segurança cibernética, possibilitando a descoberta e compreensão de incidentes. A função de responder desenvolve e implementa atividades para conter o impacto de incidentes de segurança. A função de recuperar implementa atividades para restaurar recursos ou serviços afetados por ameaças de segurança. Por fim, a função da Governança é proporcionar o necessário para atingir e priorizar os resultados das outras cinco funções.

Enquanto a tríade CIA estabeleceu a base confidencialidade, integridade e disponibilidade [Lipner and Anderson 2018], a evolução da cibersegurança e a contribuição de diversos especialistas e organizações ao longo dos anos ajudaram a expandir e refinar es-

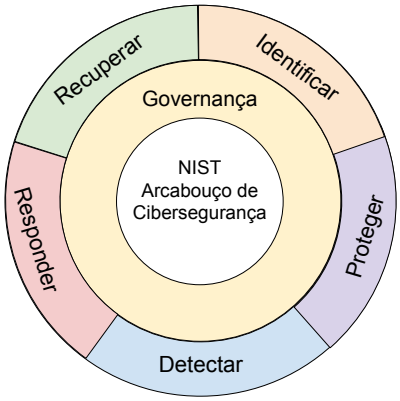


Figura 1.1: Arcabouço de cibersegurança do NIST (adaptado [de Padrões e Tecnologia (NIST) 2024]).

ses conceitos, incluindo a autenticidade e não repúdio, para atender às necessidades mais complexas da cibersegurança. A **confidencialidade** garante que as informações sensíveis sejam acessíveis apenas para aqueles com autorização, protegendo contra violações de dados e divulgações não autorizadas. A **integridade** assegura a precisão e a confiabilidade dos dados, prevenindo alterações não autorizadas que possam comprometer sua veracidade [Laprie et al. 2004]. A **disponibilidade** garante que as informações e os recursos estejam acessíveis aos usuários autorizados sempre que necessário, protegendo contra interrupções como ataques de negação de serviço. A **autenticidade** confirma que os usuários e os sistemas são genuínos, garantindo que as comunicações e transações sejam feitas por entidades legítimas. O **não repúdio** assegura que uma transação ou comunicação não possa ser negada posteriormente por nenhuma das partes envolvidas, garantindo que as ações realizadas sejam rastreáveis e verificáveis.

1.2.1.1. As bases da Cibersegurança

Constantemente, criminosos cibernéticos buscam por vulnerabilidades em seus alvos para obter algum tipo de vantagem. Uma **vulnerabilidade** em um sistema é uma fraqueza no projeto, configuração ou processos que pode ser explorada, comprometendo a segurança. Isso inclui falhas inerentes na arquitetura, parâmetros mal configurados ou procedimentos inadequados que abrem brechas (portas) para ataques. No sistema vulnerável, existe uma oportunidade para uma ameaça quebrar um atributo de segurança (*e.g.*, confiabilidade, disponibilidade, integridade, autenticidade e não repúdio). Além dos sistemas, os humanos desempenham um papel importante no campo da segurança cibernética [Alsharif et al. 2022]. A engenharia social é uma técnica de ataque na qual os invasores manipulam pessoas para obter informações confidenciais ou acesso a sistemas fraudulentamente. Neste caso, em vez de explorar as vulnerabilidades técnicas, esses ataques exploram a confiança, curiosidade ou a falta de conhecimento das pessoas, persuadindo-as a revelar dados sensíveis, clicar em links maliciosos ou executar ações que comprometem a segurança.

Quando uma vulnerabilidade é explorada, um invasor pode comprometer o funcionamento de *softwares* e serviços, roubar identidades e dados pessoais e coordenar ataques

contra outros sistemas. Para explorar uma vulnerabilidade de segurança é necessário um **exploit**, ou seja, uma técnica ou *software* projetado para se beneficiar de uma vulnerabilidade específica. Existem vários tipos de *exploits*, cada um seguindo diferentes técnicas e propósitos. Por exemplo, o SQL Injection é uma técnica onde um invasor insere código SQL malicioso em uma entrada de um aplicativo para manipular ou acessar a base de dados de forma não autorizada [Nair 2024]. Outro exemplo são os *zero-day exploits*, que se referem a vulnerabilidades desconhecidas e sem correção disponível [Vegesna 2023].

Na cibersegurança, uma **ameaça** (do inglês, *threat*) representa qualquer potencial perigo de exploração de uma vulnerabilidade para causar danos, perda de dados, ou interrupção dos serviços, sendo uma fonte deliberada de perigo ou dano potencial. Isso inclui o impacto adverso na operação do sistema ou nos recursos do sistema, incluindo dados. O **ator da ameaça** (do inglês, *threat actor*) é especificamente o indivíduo, grupo, organização ou governo que tem a intenção ou é responsável (o ator) por executar, ou planejar um ataque cibernético [Bruijne et al. 2017]. O termo **adversário** é usualmente utilizado para referenciar qualquer entidade ou indivíduo que realiza atividades maliciosas com a intenção de comprometer a segurança de sistemas, redes ou dados.

Existem diferentes conceitos e níveis de adversários. O Departamento de Defesa dos Estados Unidos (do inglês, *US Department of Defense* — DoD) define um adversário como qualquer entidade (nacional, transnacional ou grupo) envolvida em atividades hostis, conflitantes ou de oposição, incluindo forças ou agentes com a intenção e capacidade de comprometer ou influenciar adversamente as operações militares, a segurança nacional ou os interesses dos Estados Unidos[2]. O Conselho de Ciência e Defesa (do inglês, *Defense Science Board* — DSB) adota uma definição mais específica e técnica, considerando um adversário como uma entidade com capacidade técnica, intenção e motivação para comprometer ou atacar os sistemas de defesa, especialmente em contextos cibernéticos e tecnológicos [3]. Ou seja, o DSB enfatiza as capacidades técnicas e os métodos usados pelos adversários, como ciberataques sofisticados, engenharia social e outras técnicas avançadas que comprometam os sistemas de defesa e as infraestruturas críticas. Assim, enquanto todos os atores da ameaça são adversários, nem todos os adversários são atores da ameaça. Um adversário é qualquer entidade com a intenção de causar danos, enquanto um ator da ameaça é quem realmente executa ou planeja o ataque.

O **modelo de adversário** (do inglês, *adversary model*) é uma formalização que descreve as capacidades, objetivos e comportamentos de um atacante em sistemas computacionais ou redes. Ele é fundamental na área de segurança, especialmente em criptografia, onde é utilizado para validar a segurança de esquemas e protocolos criptográficos. Os modelos de adversário variam em complexidade, desde representações simples até definições detalhadas que incluem diferentes tipos de atacantes com habilidades e recursos específicos. A utilização adequada desses modelos permite que pesquisadores e profissionais de segurança avaliem e testem a robustez de sistemas contra possíveis ameaças, contribuindo para o desenvolvimento de soluções mais seguras [Do et al. 2019].

O primeiro modelo amplamente reconhecido de adversário é o Modelo Dolev-Yao, introduzido na década de 1980 [Dolev and Yao 1983]. Esse modelo assume que um

[2] https://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/dictionary.pdf
[3] https://dsb.cto.mil/reports/2020s/DSB-CyberSupplyChain_ExecutiveSummary.pdf

atacante pode escutar toda a comunicação em uma rede e enviar mensagens, mas não pode quebrar a criptografia. Ele serve como base para a análise de protocolos de segurança, permitindo avaliar esquemas criptográficos sob um adversário poderoso e idealizado. Outro exemplo é o Modelo Bellare-Rogaway, que expande as capacidades do adversário, permitindo a modelagem de diferentes tipos de atacantes, como passivos e ativos [Bellare and Rogaway 1993]. Este modelo introduz a noção de consultas, ações que o adversário pode realizar, como enviar mensagens ou revelar segredos, oferecendo uma estrutura mais flexível e abrangente para a análise de segurança em protocolos modernos, incluindo os utilizados em ambientes de Internet das Coisas (IoT) [Do et al. 2019].

Os **vetores de ataque** (do inglês, *attack vectors*) são os métodos ou meios usados por uma ameaça (ou atacante) para explorar as vulnerabilidades em sistemas e redes. Eles representam as diversas formas pelas quais um invasor pode explorar vulnerabilidades em um sistema e obter acesso não autorizado a dados ou recursos. Um dos principais vetores de ameaça é o funcionário comprometido, que pode ser manipulado para fornecer acesso não autorizado. A infecção por e-mail, frequentemente via *phishing* ou *SPAM*, é outra técnica comum, onde mensagens maliciosas induzem os usuários a clicar em links ou abrir anexos infectados. Além disso, vulnerabilidades em sistemas e pacotes de terceiros podem ser exploradas, assim como a introdução de *malware* através de mídias removíveis, como pen drives. Dispositivos móveis também são alvos, devido a falhas em aplicativos ou sistemas operacionais. Por fim, ataques direcionados à rede do usuário, visam comprometer a infraestrutura de uma organização [Tiwari and Dwivedi 2016].

O ataque de negação de serviço distribuído ou *Distributed Denial of Service* (DDoS) combina vários dispositivos conectados para atacar um alvo [Neira et al. 2023b]. No modo clássico, os ataques DDoS esgotam os recursos de computação da infraestrutura vítima, criando várias conexões de fontes diferentes [Douligeris and Mitrokotsa 2004]. As partes padrão de um ataque DDoS são os atacantes, os dispositivos infectados e a vítima. Um zumbi, robô da web ou simplesmente *bot* é um dispositivo infectado por *malware* conectado à Internet que executa tarefas programadas [Ngo et al. 2020]. Uma rede de robôs ou *botnet* é um grupo de vários *bots* controlados remotamente por atacantes ou *botmasters*. Uma vítima é um servidor ou uma rede de computadores que contém os recursos para o correto funcionamento de um serviço [Salim et al. 2020]. Para conduzir um ataque DDoS, um *botmaster* envia comandos à *botnet* para iniciar conexões com a vítima. A duração dos ataques DDoS varia de minutos a dias, podendo atingir milhões de solicitações por segundo[4].

A Figura 1.2 ilustra a operação de um ataque DDoS. Um *botmaster* gerencia os *bots* por meio do tráfego de controle, fazendo com que os vários *bots* enviem o tráfego de ataque para a vítima. O *botmaster* explora vários tipos de fraquezas de diferentes dispositivos conectados à Internet para espalhar seu código malicioso. O alvo pode ser dispositivos com mais recursos, como computadores de mesa, servidores, *tablets* e *smartphones* [Wlosinski 2019] ou dispositivos com recursos limitados, como dispositivos que compõem a Internet das coisas (*Internet of Things* — IoT) [Wlosinski 2019], como câmeras de segurança ou *smart* TVs. Após a infecção, os atacantes controlam os *bots* por coman-

[4] ttps://cloud.google.com/blog/products/identity-security/google-cloud-mitigated-largest-ddos-attack-peaking-above-398-million-rps

dos. Um ataque DDoS ocorre quando o atacante instrui os *bots* a criar conexões com a infraestrutura vítima para consumir todos os recursos disponíveis.

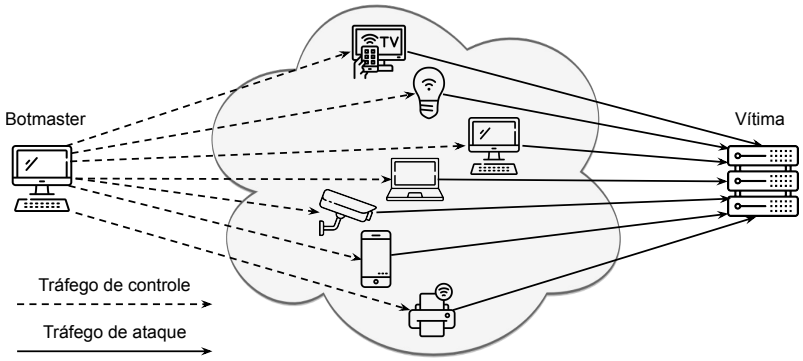


Figura 1.2: Estrutura Básica do Ataque DDoS (adaptado de [Bhatia et al. 2018])

Um *malware* é qualquer *software* malicioso projetado para explorar vulnerabilidades em sistemas de computadores e redes. Exemplos de *malware* incluem vírus, *worms*, *bots*, *adware*, *trojans*, *spyware*, *cavalos de troia*, *botnets* e *ransomware*. Os vírus são programas que se anexam a arquivos legítimos e se espalham quando esses arquivos são compartilhados, como o vírus ILOVEYOU, que corrompe arquivos e se propaga por e-mails. Os *Worms* são *malwares* que se replicam automaticamente por meio de redes, como o Conficker, que infectou milhões de computadores explorando vulnerabilidades no Windows [Shin et al. 2011]. Um *adware* é um *software* que exibe anúncios indesejados e coleta dados do usuário, como o AdsExhaust, que captura telas, interage com navegadores usando teclas simuladas e redireciona para URLs específicas para gerar receita [5]. Os *trojans* são *malwares* disfarçados de *software* legítimo, usados para roubar informações ou instalar outras ameaças. Um *ransomware* é um tipo de *malware* que criptografa arquivos e exige pagamento de resgate para liberar o acesso, como o WannaCry [Martin et al. 2018]. Esses *malwares* se espalham por meio de anexos de e-mail infectados, *downloads* de *software* malicioso e exploração de falhas de segurança em *software* desatualizado.

Um ***phishing*** (pescaria em português) é uma técnica de ataque cibernético cujo objetivo é enganar as pessoas para que revelem informações pessoais, como senhas, números de cartão de crédito, CPF e contas bancárias. A infecção por e-mail, frequentemente associada ao *phishing*, envolve e-mails maliciosos projetados para parecer legítimos, induzindo os destinatários a clicar em links ou abrir anexos que podem comprometer seus sistemas. O método mais comum inclui o envio de e-mails ou mensagens de texto (SMS), mensagens em aplicativos de conversa e redes sociais que direcionam os usuários para sites falsos, onde são induzidos a inserir seus dados confidenciais [Montagner and Westphall 2022]. Esses sites fraudulentos imitam entidades legítimas, como bancos, redes sociais ou outras instituições confiáveis, para aumentar a credibilidade e enganar as vítimas. Uma vez que o invasor obtém essas informações, elas podem ser usadas para roubo de identidade, fraude financeira, entre outros.

Enquanto o *phishing* é uma técnica de ataque cibernético focada em roubar in-

[5] https://www.esentire.com/blog/adsexhaust-a-newly-discovered-adware-masquerading-oculus-installer

formações pessoais, o **spam** refere-se ao envio massivo de mensagens indesejadas, geralmente com conteúdo publicitário. Essas mensagens são frequentemente enviadas por bots ou serviços automatizados, e visam promover produtos, serviços ou, em alguns casos, disseminar *malwares*. Um spam ocorre em diversos formatos, incluindo e-mails, mensagens de texto (SMS), mensagens em redes sociais e comentários em blogs [Tiwari and Dwivedi 2016]. No entanto, há uma interseção significativa entre spam e *phishing*. Muitas vezes, e-mails de spam são usados como vetores para ataques de *phishing*, onde mensagens aparentemente inofensivas contêm links para sites fraudulentos ou anexos maliciosos. A capacidade dos invasores de enviar grandes volumes de spam aumenta as chances de sucesso de seus ataques de *phishing*.

1.2.2. Ciência de Dados

Esta subseção aborda os fundamentos da ciência de dados, aprendizado profundo, aprendizado de máquina, inteligência artificial e suas interseções. A ciência de dados é um campo interdisciplinar que envolve a aquisição de dados, preparação, pré-processamento de características, visualização de dados e análise. Todas essas etapas são realizadas sobre um grande volume de dados (*big data*). *Big data* refere-se a conjuntos de dados extremamente grandes e complexos que não podem ser processados de maneira eficiente por técnicas convencionais. Existem intersecções entre a ciência de dados e o campo de Inteligência Artificial, conforme ilustrado na Figura 1.3. A inteligência artificial busca definir soluções e algoritmos para que as máquinas imitem a inteligência humana e da natureza. A IA compreende o subcampo do aprendizado de máquina que por sua vez engloba o subcampo do aprendizado profundo. O aprendizado de máquina se concentra no desenvolvimento de algoritmos que permitem aos computadores aprenderem e fazerem previsões baseadas em dados conforme detalhado na próxima subseção.

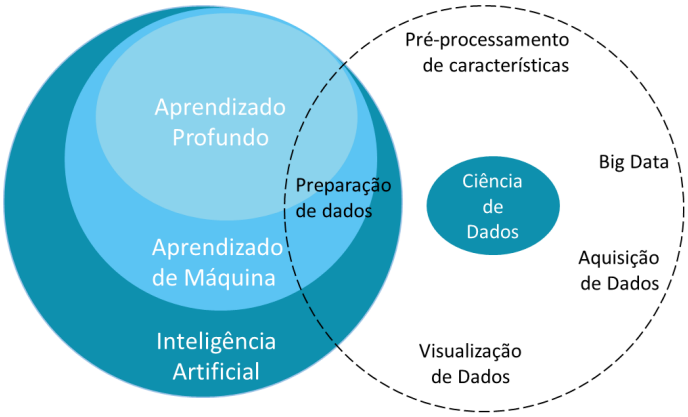


Figura 1.3: Inteligência artificial vs ciência de dados

No contexto de ciência de dados em cibersegurança, a aquisição de dados envolve a coleta de dados de diversas fontes, como logs de rede, registros de sistemas, dados de sensores e informações de usuários, para monitorar e detectar atividades suspeitas [Neira et al. 2023b]. A preparação de dados é crucial, pois os dados coletados frequentemente contêm ruídos, valores ausentes ou inconsistências que devem ser limpos e organizados para análise eficaz [Borges et al. 2024]. O pré-processamento de características inclui a transformação dos dados brutos em um formato adequado para modelagem, como a nor-

malização de valores, codificação de variáveis categóricas e seleção das características mais relevantes para detectar anomalias [Brito et al. 2023]. A visualização de dados é utilizada para explorar e obter *insights*, permitindo que os analistas de segurança/sistemas automatizados identifiquem padrões e tendências em atividades maliciosas de forma mais intuitiva [Neira et al. 2023a]. Finalmente, a análise de dados aplica técnicas estatísticas e algoritmos de aprendizado de máquina para detectar ameaças e predizer ataques. Com o aumento exponencial na geração de dados, impulsionado pelo uso da Internet e dispositivos conectados, as organizações enfrentam o desafio de gerenciar e analisar essas informações. As características principais do *big data* incluem volume, velocidade, variedade, veracidade e valor, que juntos definem a complexidade do seu processamento. Na área de cibersegurança, o *big data* é utilizado para aprimorar a detecção de ameaças, análise de riscos e resposta a incidentes, permitindo que as empresas se protejam de forma mais eficaz contra ataques cibernéticos [Alani 2021]. A seguir são discutidos os demais conceitos de inteligência artificial e aprendizado de máquina.

1.2.2.1. Inteligência artificial e Aprendizado de Máquina

O Dicionário de Cambridge define a palavra aprender como o ato ou ação de adquirir um novo conhecimento, ou uma nova habilidade[6]. Muito antes da invenção dos computadores, o ser humano já tentava emular o aprendizado em máquinas. No fim da Idade Média, Roger Bacon teria construído bonecos que simulavam a fala, Leonardo da Vinci teria construído um leão que possuía a capacidade de andar [Crevier 1993]. Com o poder de automatização proporcionado pela computação, cientistas identificaram a possibilidade de evoluir a automatização do aprendizado. O objetivo da IA é construir máquinas inteligentes capazes de imitar o comportamento humano e da natureza [Kour and Gondhi 2020]. A IA engloba técnicas como o aprendizado de máquina (AM) e o *deep learning* (DL). A Figura 1.4 representa graficamente a relação entre a IA, AM e DL.

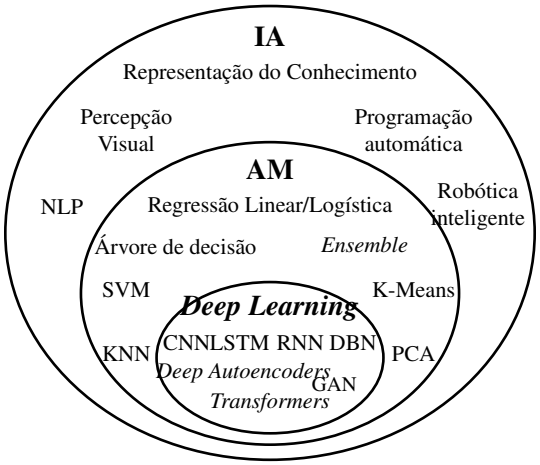


Figura 1.4: Inteligência artificial e subáreas (adaptado de [Kaluarachchi et al. 2021])

Dentre as várias técnicas de IA, aquelas de AM vem ganhando grande abrangên-

[6] https://dictionary.cambridge.org/dictionary/english-portuguese/learn?q=Learn

cia. A Figura 1.5 apresenta dois paradigmas de programação de sistemas computacionais, a programação tradicional e o AM. Na programação tradicional (Figura 1.5a), um conjunto de regras pré-determinadas agem sobre os dados para gerar as respostas. Por exemplo, um modo simplista de quantificar ganhos sobre a venda de um produto é calcular as receitas desse produto e subtrair os custos. Deste modo, regras predefinidas atuam sobre os dados para gerar a saída desejada pelo operador do sistema [O'Reilly 2021]. A literatura do AM propõe algoritmos capazes de analisar e aprender relações entre os dados (Figura 1.5b). Assim, a saída esperada do processo de análise de dados é um modelo capaz de identificar os padrões aprendidos com a análise prévia quando confrontado com novas observações [O'Reilly 2021]. Uma questão fundamental para o sucesso da aplicação do AM é a disponibilidade de dados. Vieses causados pela manipulação humana, por exemplo, na seleção de atributos ou na rotulação dos dados e a baixa representatividade dos dados podem induzir os algoritmos de AM a aprender erradamente. Assim, o modelo gerado fica específico para os dados de treinamento e acaba não generalizando os resultados para outros cenários [Monroe 2021].

(a) Programação tradicional

(b) Aprendizado de máquina

Figura 1.5: Paradigmas de Programação (adaptado de [Raschka 2020, O'Reilly 2021])

Ao longo da evolução do AM, pesquisadores propuseram algoritmos com características e objetivos diferentes. Utilizando essas diferentes características, a literatura classifica os algoritmos de AM quanto à tarefa realizada pelo algoritmo, à estratégia de aprendizado utilizada pelo algoritmo e à profundidade do algoritmo [Ibitoye et al. 2020]. A Figura 1.6 apresenta a classificação dos algoritmos de AM.

O ramo 'Tarefa', o primeiro ramo da Figura 1.6, apresenta quatro tarefas que podem ser realizadas com os algoritmos de AM, sendo eles: classificação, regressão, clusterização e as regras de associação. A **classificação** é o processo de identificar a classe real de dados ainda não rotulados utilizando algoritmos de AM [Muhammad and Yan 2015]. O rótulo identifica a natureza da ação, por exemplo, o rótulo distingue o tráfego em uma rede de computadores, onde as opções são o tráfego normal ou tráfego de ataque. A principal diferença entre a classificação e os outros tipos de tarefas é que o rótulo é uma variável categórica ou discreta [Singh 2019]. Ou seja, o resultado da classificação será interpretado como uma classe do problema.

A Figura 1.7 ilustra o conceito da tarefa de classificar dados. Os círculos e os quadrados representam observações de diferentes classes. Deste modo, os algoritmos de classificação constroem modelos onde é possível distinguir os dados. A linha separando

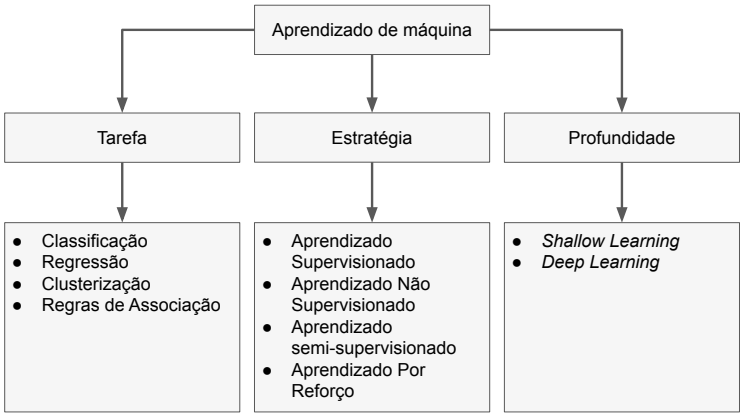


Figura 1.6: Classificação do Aprendizado de Máquina (adaptado de [Ibitoye et al. 2020])

os círculos dos quadrados representa esse modelo. Caso uma nova observação esteja no topo da figura (acima da linha), o modelo irá classificá-la como círculo. Caso a nova observação esteja na parte de baixo da figura (depois da linha) o modelo classificá-la-á como quadrado. Os algoritmos de AM de classificação incluem o perceptron, algoritmo passivo agressivo, *Support Vector Machines* (SVM), classificadores *ensemble* entre outros.

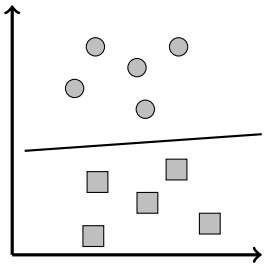


Figura 1.7: Exemplo de Classificação (adaptado de [Singh 2019])

A segunda tarefa realizada com AM é a **regressão**. A principal diferença entre a regressão e a classificação está no tipo da variável alvo (rótulo). Na regressão, o rótulo é composto por uma variável contínua ou numérica. Isso significa que a regressão visa identificar um número, como o total de pacotes trafegados em uma rede durante um ataque DDoS ou o total de endereços IP ativos na rede. A Figura 1.8 ilustra a regressão. Os círculos apresentam o total de pacotes trafegados em uma rede durante um ataque DDoS. Por exemplo, os algoritmos de regressão identificam uma reta relacionando o total de pacotes com a progressão do ataque. Assim, quando uma nova observação estiver disponível, o algoritmo irá buscar na reta onde a nova observação será disposta. Assim, o algoritmo de regressão identifica a quantidade de pacotes conforme o ataque evolui. Os algoritmos de AM para regressão incluem o SVM, *k-Nearest Neighbors* (k-NN), processo gaussiano, árvores de decisão, regressores *ensemble* entre outros.

A terceira tarefa realizada com AM é a **clusterização**. Diferentemente da regressão e da classificação, os algoritmos de clusterização segmentam os dados seguindo as propriedades dos dados. Assim, os algoritmos de AM para a clusterização analisam os dados e possuem como saída agrupamentos de observações com padrões similares. A Figura 1.9 ilustra a clusterização. Na Figura 1.9(a), os dados observados foram agrupa-

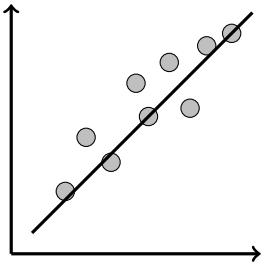


Figura 1.8: Exemplo de Regressão (adaptado de [Singh 2019])

dos nos conjuntos A e B. O conjunto A possui seis observações e o conjunto B possui cinco. Caso uma nova observação esteja disposta próxima ao agrupamento A, ela fará parte desse agrupamento. O agrupamento B cresce caso uma nova observação seja similar às observações contidas no agrupamento B. Definidos os limites, como o tamanho dos agrupamentos ou a quantidade de agrupamentos, é possível obter resultados diferentes. A Figura 1.9(b) apresenta a mesma disposição dos círculos cinza, porém agrupados em quatro grupos diferentes. Esse resultado é obtido adicionando uma regra para encontrar quatro grupos. Algoritmos de AM que podem realizar a clusterização incluem o *K-means*, DBSCAN, *Spectral clustering* entre outros.

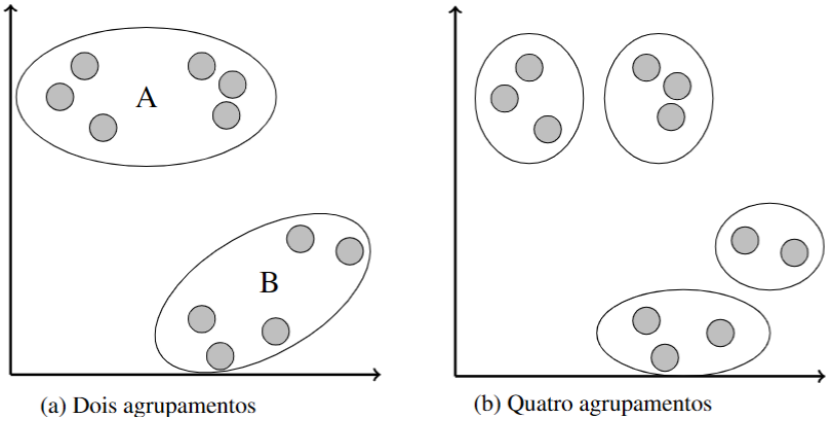


Figura 1.9: Exemplo de Clusterização (adaptado de [Singh 2019])

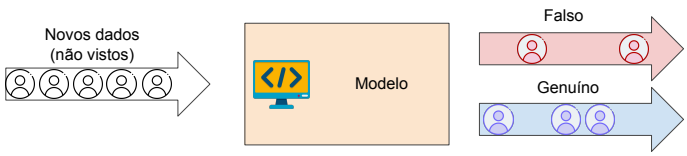
A quarta tarefa realizada com AM é denominada de **regras de associação**. Nesse contexto, a associação tem o sentido de co-ocorrência. Isso ocorre, pois o objetivo é minerar dados comerciais em busca de conjuntos das variáveis que aparecem frequentemente. O exemplo clássico da adoção de regras de associação é na análise de vendas em uma loja. Neste caso, as variáveis representam todos os produtos vendidos pela loja. O objetivo é encontrar conjuntos de itens que são adquiridos juntos. Várias decisões estratégicas se embasam nos resultados das regras de associação identificadas. Por exemplo: facilitar a compra definindo quais itens devem ficar nas prateleiras próximas; melhorar a experiência dos usuários ao manusear catálogos de produtos; segmentar os diferentes tipos de clientes; e auxiliar na definição de estratégias de *marketing* em promoções [Hastie et al. 2009, Singh 2019]. Os algoritmos AIS, SETM e APRIORI são exemplos de algoritmos utilizados para realizar o aprendizado por regras de associação [Kumbhare and Chobe

2014].

O ramo 'Estratégia', o segundo ramo da Figura 1.6, apresenta três tipos de estratégias para aplicar o AM: o aprendizado supervisionado, o aprendizado não supervisionado, e o aprendizado por reforço. Hastie et al. 2009 e Crisci et al. 2012 definem o **aprendizado supervisionado** como a ação de analisar um problema utilizando variáveis explicativas definidas por $X \in \mathscr{X}$, onde o objetivo é aprender a prever uma variável aleatória $Y \in \mathscr{Y}$. Ou seja, o processo de aprendizado é guiado pela busca de padrões nos dados que descrevem o problema (conjunto $\mathscr{X}$). Esses padrões são relacionados aos rótulos verdadeiros (conjunto $\mathscr{Y}$) de elementos dos elementos conhecidos e vão ser utilizados para identificar os rótulos dos elementos ainda não conhecidos pelo AM [Singh 2019].

O aprendizado supervisionado é utilizado para tarefas de regressão ou de classificação. Independentemente do tipo tarefa, o aprendizado supervisionado acontece em duas fases, a fase de treinamento e a fase de teste. Durante o treinamento, o algoritmo de AM supervisionado analisa a relação entre os dados e os rótulos para construir modelos capazes de produzir saídas corretas quando for confrontado com novos dados. A fase de teste é utilizada para avaliar a qualidade do modelo. Nesta fase o modelo gerado no treinamento é utilizado para catalogar as classes de novos dados. Os rótulos reais desses novos dados são conhecidos pelo desenvolvedor, mas não pelo modelo. Deste modo, os rótulos reais são comparados com os resultados obtidos pelo modelo treinado. Após o treinamento e o teste o modelo será utilizado em produção, onde ele irá receber dados que nem o desenvolvedor conhece o verdadeiro rótulo. A Figura 1.10 apresenta o funcionamento do aprendizado supervisionado. Primeiramente, o algoritmo de AM selecionado aprende com os dados do conjunto de treinamento (Figura 1.10a). Um novo conjunto de dados contendo informações sobre observações não utilizadas no treinamento é apresentado para o modelo treinado (Figura 1.10b). O intuito é verificar se o modelo pode identificar corretamente os rótulos das novas observações [Singh 2019].

(a) Ilustração da fase treinamento no aprendizado supervisionado



(b) Ilustração da fase teste no aprendizado supervisionado

Figura 1.10: Aprendizado Supervisionado (adaptado de [Singh 2019])

A segunda estratégia de aprendizado é o **aprendizado não supervisionado**. O aprendizado não supervisionado diversifica as possibilidades e a aumenta relevância que o aprendizado possui. Isso ocorre, pois o aprendizado não supervisionado tenta resolver um dos maiores problemas relacionados ao aprendizado supervisionado, a generalização dos

resultados devido ao uso de rótulos. Durante o treinamento, os algoritmos de aprendizado supervisionado relacionam os dados observados com os rótulos disponíveis, aprendendo assim sobre a natureza do problema. Porém, existem casos em que os dados observados representam apenas parte do problema [Sutton and Barto 2018]. Caso uma nova observação não siga os padrões pré-estabelecidos na base de dados, o modelo produz saídas incorretas. Uma nova fraude relacionada a cartões de créditos pode não ser detectada antes de causar prejuízos, por exemplo. Bem como um tipo diferente de ciberataque pode não ser detectado caso este tipo de ataque difere dos ataques que o algoritmo de AM foi treinado. Assim, o objetivo do aprendizado não supervisionado é inferir relacionamentos entre os dados sem o auxílio dos rótulos reais ou recompensas do ambiente [Hastie et al. 2009]. O aprendizado não supervisionado é realizado por intermédio da clusterização ou das regras de associação [Hastie et al. 2009, Singh 2019]. Apesar das vantagens do aprendizado não supervisionado, escolher o algoritmo e os parâmetros corretos não são triviais. Como citado anteriormente (Figura 1.9), diferentes parâmetros geram diferentes resultados.

A terceira estratégia de aprendizado é o **aprendizado semi-supervisionado**. O aprendizado semi-supervisionado compreende algoritmos que aprendem com parte dos dados rotulados e parte sem rótulos. Existem cenários onde rotular toda a base de dados não é uma tarefa trivial. Nesses casos, o custo de tempo e dinheiro para rotular toda a base pode ser alto, dificultando o uso do AM supervisionado. Por outro lado, existem cenários em que é possível melhorar significativamente os resultados apenas com parte da base rotulada. Para diminuir as desvantagens relacionadas à obtenção dos rótulos e para melhorar os resultados dos algoritmos não supervisionadas, a literatura apresenta o aprendizado semi-supervisionado [Zhou and Belkin 2014]. Áreas como a detecção de ataques DDoS, detecção de anomalias em redes de computadores e detecção de *ransomwares* [Noorbehbahani and Saberi 2020] utilizam o aprendizado semi-supervisionado. *Transductive support vector machines* (TSVMs), *co-training*, *Expectation-Maximization* (EM) são exemplos de algoritmos de AM semi-supervisionado [Pise and Kulkarni 2008].

A quarta estratégia de aprendizado é o **aprendizado por reforço**. Como no aprendizado não supervisionado, o aprendizado por reforço é uma alternativa nos cenários onde coletar dados de todas as classes é impraticável. Porém, o aprendizado por reforço também difere do aprendizado não supervisionado, pois o objetivo do aprendizado por reforço não é inferir relacionamentos entre as observações sem conhecimento prévio [Sutton and Barto 2018]. O objetivo do aprendizado por reforço é interagir com o ambiente a fim de aprender a lidar com o ambiente. O aprendizado acontece baseado em recompensas e penalidades. A cada geração, os agentes interagem com o ambiente testando-o em busca de aprender a aumentar as recompensas e reduzir as penalidades recebidas. A cada geração, o conhecimento acumulado é repassado para as gerações com o intuito de melhorá-las. Assim, ao utilizar o aprendizado por reforço é possível obter automaticamente habilidades comportamentais que maximizam as recompensas e reduzam as penalidades [Sutton and Barto 2018]. *Q-Learning*, *Temporal Difference Learning* e SARSA, (do inglês, *State-Action-Reward-State-Action*) são exemplos de algoritmos de aprendizado por reforço [Sewak 2019].

O terceiro ramo da Figura 1.6 divide os algoritmos de AM em relação à 'profundidade dos algoritmos'. *Shallow learning* e *deep learning* são as duas classificações dis-

poníveis na literatura. Aprendizado raso ou aprendizado superficial são traduções para o termo *shallow learning*. Os termos raso ou superficial se referem ao modo que os algoritmos utilizam para gerar os modelos de AM. Portanto, esse nome não deve ser interpretado como um demérito desses algoritmos, pois ele não tem relação com a qualidade dos resultados obtidos. Assim, para evitar interpretações dúbias acerca do ramo 'profundidade', este trabalho utiliza apenas os termos em inglês (*shallow learning* e *deep learning*).

Shallow learning compreende algoritmos de AM tradicionais que não utilizam várias camadas ocultas [Liu and Lang 2019]. O número de camadas ocultas não é unanimidade na literatura, porém algoritmos com menos de duas camadas ocultas são geralmente associados ao *shallow learning* [Kawaguchi 2016]. SVM, k-NN, *Naïve Bayes*, redes neurais artificiais (do inglês, *Artificial Neural Network* - ANN) e árvores de decisão são exemplos de algoritmos classificados como *shallow learning* [Liu and Lang 2019]. Como cada algoritmo tem sua peculiaridade, é possível utilizar o *Naïve Bayes* em várias áreas. Recentemente, o *shallow learning* vem sendo empregado na medicina [Nilashi et al. 2020], na predição de ataques DDoS [Borges et al. 2024], entre outros.

Nos últimos anos, os algoritmos de ***deep learning*** evoluíram muito. [Zhang et al. 2018] definem *deep learning* como o processo de aprender a relação entre várias variáveis, a relação que governa as variáveis e o conhecimento que dá significado para a relação entre as variáveis. O *deep learning* diferencia-se do *shallow learning*, pois compreende algoritmos que utilizam várias camadas ocultas para realizar o aprendizado [Liu and Lang 2019]. Deste modo, a principal diferença entre os tipos de aprendizado reside no número de transformações que os dados de entrada sofrem até alcançar a saída [Wang et al. 2020]. Um grande benefício dos algoritmos *deep learning* é o fato de não precisar de seleção de atributos prévia. Em muitos casos, a seleção de atributos melhora o resultado obtido pelos algoritmos de AM. Porém, se a seleção de atributos não for realizada corretamente, os resultados sairão enviesados ou prejudicados. Além disso, é necessário despender tempo e conhecimento prévio para realizar essa ação. Como o *deep learning* aprende a representação dos atributos apenas observando os dados originais, este consegue escolher os atributos mais relevantes para atingir aos melhores resultados [Liu and Lang 2019].

Para casos em que grandes volumes de dados estejam disponíveis, é possível que algoritmos de *deep learning* apresentem resultados melhores em comparação com algoritmos de *shallow learning* [Liu and Lang 2019]. Isso ocorre, pois os algoritmos de *deep learning* comparam os dados reais com a saída gerada pelo algoritmo. É esperado que no início do treinamento o resultado esteja longe do real. Porém, ao longo das iterações, o *deep learning* vai ajustando as configurações e os resultados começam a se aproximar dos reais. Como ocorrem muitas iterações, podendo variar entre 100 a 20.000 iterações, é necessário poder computacional suficiente para lidar com todas as iterações [Janos 2020].

1.2.2.2. Automated Machine Learning

Recentemente, a literatura vem evoluindo os algoritmos em direção ao que é denominado de *Automated Machine Learning* (AutoML), cujo objetivo é democratizar, simplificar e reduzir o custo do aprendizado de máquina. Os *Frameworks* AutoML visam encontrar e configurar algoritmos de aprendizado de máquina que reduzam erros de classificação

para o conjunto de dados utilizados pelo usuário [Feurer et al. 2015]. Os *frameworks* AutoML atuam como especialistas em aprendizado de máquina para sugerir o algoritmo de aprendizado de máquina adequado para cada conjunto de dados. Assim, o AutoML acelera e automatiza o processo de obtenção de modelos de aprendizado de máquina.

Atualmente, existem *frameworks* AutoML de dois tipos, o primeiro é do tipo otimização de hiperparâmetros (do inglês, *Hyperparameter Optimization* - HPO) e o segundo é do tipo busca de arquiteturas de redes neurais (do inglês, *Neural Architecture Search* - NAS). A literatura apresenta diferentes *frameworks* AutoML do tipo HPO. Exemplos de *frameworks* de AutoML de código aberto são: AutoGluon AutoML, Auto-sklearn, H2O AutoML e o *Tree-based Pipeline Optimization Tool* (TPOT). [Horsanali et al. 2021] criaram seus próprios *frameworks* AutoML do tipo HPO. Neste caso, os autores definiram que o *framework* iria avaliar seis algoritmos de aprendizado de máquina: Decision Tree, K-nearest neighbors, Logistic Regression, Naive Bayes, Random Forest, e o SVM. Assim, o *framework* AutoML proposto treina e testa todos os algoritmos com os mesmos dados. O algoritmo de aprendizado de máquina que maximiza a acurácia é selecionado.

Cada *framework* AutoML do tipo HPO define sua estratégia para selecionar e configurar os algoritmos de aprendizado de máquina adequados para o contexto dos dados explorados pelo usuário. Isto permite que os *frameworks* AutoML apresentem características únicas, como tempo de execução, algoritmos de aprendizado de máquina avaliados e linguagem de programação [Feurer et al. 2015]. A Figura 1.11 mostra o funcionamento geral dos *frameworks* AutoML do tipo HPO. Na Etapa 1, os *frameworks* AutoML do tipo HPO definem os algoritmos de aprendizado de máquina candidatos. Os algoritmos candidatos podem variar entre os diferentes *frameworks*. Por exemplo, os *frameworks* AutoML podem usar todos os algoritmos implementados por uma biblioteca (por exemplo, Scikit-learn e Weka) ou restringir o espaço de busca a um conjunto de algoritmos que funcionam bem na maioria dos casos.

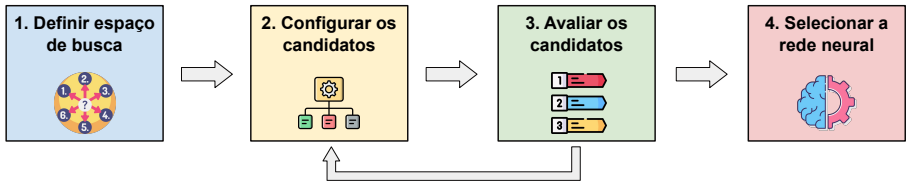


Figura 1.11: Operação Geral dos *Frameworks* AutoML (adaptado de [Ren et al. 2021])

A Etapa 2 visa configurar um subconjunto ou todos os algoritmos de aprendizado de máquina candidatos. Normalmente, esta etapa usa algum processo de otimização como a otimização bayesiana [Feurer et al. 2019]. Assim, não é necessário que os *frameworks* AutoML avaliem todas as combinações de configuração de algoritmos de aprendizado de máquina. Na Etapa 3, os *frameworks* AutoML treinam e testam algoritmos de aprendizado de máquina candidatos configurados usando o conjunto de dados selecionado pelo usuário do *framework*. Os resultados da acurácia podem ser um critério para avaliar algoritmos. Porém, é comum que o usuário escolha diferentes critérios de avaliação, como precisão, *recall* ou F1-score.

Ao final da etapa de avaliação, o *framework* retorna à Etapa 2 para que os algoritmos de aprendizado de máquina recebam novas configurações para maximizar os critérios

de avaliação. O ciclo entre as Etapas 2 e 3 se repete até que o *framework* encontre um critério de parada. O critério de parada pode ser o tempo de execução ou o número de iterações. Por exemplo, no Auto-sklearn, um *framework* AutoML do tipo HPO, o critério de parada é 60 minutos. Portanto, se o usuário não alterar este parâmetro, o Auto-sklearn será executado por 60 minutos. Na Etapa 4, o *framework* AutoML escolhe o algoritmo de aprendizado de máquina que maximiza os critérios de avaliação. Alguns *frameworks* AutoML podem combinar algoritmos de aprendizado de máquina que maximizam os critérios de avaliação para construir um conjunto de algoritmos de aprendizado de máquina ideais para os dados selecionados pelos usuários.

O AutoML do tipo NAS difere do HPO em termos dos algoritmos de aprendizado de máquina usados. Os *frameworks* AutoML do tipo HPO concentram-se em algoritmos de aprendizado de máquina do tipo *shallow learning*, enquanto os *frameworks* NAS usam algoritmos de aprendizado de máquina do tipo *deep learning*. Tanto os *frameworks* AutoML do tipo HPO quanto o NAS têm o mesmo propósito, identificar e configurar o algoritmo de aprendizado de máquina adequado conforme a necessidade dos usuários. Alguns *frameworks* que implementam o AutoML do tipo NAS são o Autokeras, o MetaQNN, o *Neural Architecture Search Network* (NASNet) e o *Mobile Neural Architecture Search Network* (MNasNet). A Google possui o AutoML Vision, um produto comercial que implementa um AutoML do tipo NAS. Além de ser um *framework* pago, com período de teste gratuito, o *framework* do Google limita a customização da execução do AutoML. Por exemplo, o usuário não pode limitar o espaço de pesquisa do AutoML selecionando apenas um conjunto de algoritmos [Feurer et al. 2015].

O AutoML do tipo NAS é especializado em selecionar e configurar algoritmos de aprendizado de máquina do tipo *deep learning* para maximizar a acurácia. Para isso, os *frameworks* AutoML do tipo NAS definem a arquitetura dos modelos *deep learning*. O *deep learning* opera com redes neurais, cuja arquitetura inclui o número de camadas ocultas, pesos, número de neurônios e funções de ativação [Lam and Abbas 2020]. Portanto, para definir a arquitetura das redes neurais, os *frameworks* AutoML do tipo NAS escolhem a combinação de componentes que maximizam a acurácia. Apesar da automação que os *frameworks* AutoML do tipo NAS apresentam, os *frameworks* requerem tempo para identificar a arquitetura apropriada. Em [Lam and Abbas 2020], os autores executaram o NASNet por 24 horas e o MNasNET por três horas para encontrar a arquitetura adequada.

Cada *framework* AutoML do tipo NAS define sua estratégia para selecionar a arquitetura de rede neural que minimiza erros. A Figura 1.11, apresentada anteriormente, expõe o funcionamento geral de *frameworks* AutoML do tipo NAS. Na Etapa 1, cada *framework* define o espaço de busca com um conjunto potencialmente grande de arquiteturas de redes neurais [Lam and Abbas 2020]. Na Etapa 2, cada *framework* usa uma estratégia diferente para selecionar a função de ativação, o número de camadas ocultas, os pesos e o número de neurônios para as arquiteturas candidatas [Lam and Abbas 2020, Imran et al. 2021]. Assim, ao final da Etapa 2, cada *framework* possui arquiteturas candidatas para minimizar erros no conjunto de dados inserido.

O *framework* avalia cada arquitetura candidata usando o conjunto de dados inserido pelo usuário na Etapa 3. O *framework* usa as arquiteturas selecionadas para mini-

mizar erros para criar novas arquiteturas candidatas. Portanto, o processo de seleção da arquitetura da rede neural retorna à Etapa 2 e avalia as novas arquiteturas na Etapa 3. O *framework* repete a iteração entre as Etapas 2 e 3 até atingir o critério de parada. O critério de parada pode ser o tempo de execução do *framework* ou a quantidade de iterações. Por exemplo, Autokeras, um *framework* AutoML do tipo NAS, define o critério de parada como um máximo de 100 tentativas. Portanto, caso o usuário não edite este parâmetro, o Autokeras repetirá os passos 2 e 3 até 100 vezes. Na Etapa 4, o *framework* seleciona a arquitetura que minimiza erros durante o processo de avaliação e a sugere ao usuário [Lam and Abbas 2020].

1.2.2.3. Medidas estatísticas

A estatística está intrinsecamente ligada com a Ciência de Dados e como os tópicos que esta abrange. Deste modo, medidas de tendência central e medidas de dispersão, conceitos essenciais para o desenvolvimento deste trabalho. A média aritmética simples, mediana e a moda são medidas de tendência central que formam a base da comparação entre distribuições e auxiliam na representação de grupos de observações [Azevedo 2016]. A média aritmética simples é uma das medidas de tendência central mais simples [Silva et al. 2015] e difundidas da literatura. A média aritmética simples é uma das formas de caracterizar os dados observados. A Fórmula 1 apresenta o modo para calcular a média aritmética simples para um conjunto de dados. Para obter-se a média aritmética simples de um conjunto de tamanho n tal que $x1; x2; ...; xn$, basta somar todo o conjunto de dados (numerador da Fórmula 1 apresentado como $\sum x$) e dividi-lo pelo total de observações (denominador da Fórmula 1 apresentado como n) [Portella et al. 2015, Silva et al. 2015, Azevedo 2016]. Uma importante característica da média aritmética simples é que esta pode ser muito influenciada por valores extremos ([Silva et al. 2015]). Portanto, outras métricas podem ser utilizadas em conjunto com a média para caracterizar os conjuntos de dados.

$$\textit{Média Aritmética Simples } (\bar{x}) = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{\sum x}{n} \tag{1}$$

A mediana é a segunda medida de tendência central apresentada neste trabalho. A mediana demanda que o conjunto de dados analisado esteja ordenado. Isso ocorre, pois a mediana indica o valor central dos dados, caso o tamanho n do conjunto de dados seja ímpar. Caso o tamanho n do conjunto de dados seja par, a mediana é a média dos dois valores centrais. Assim, a mediana divide o conjunto de dados analisado em duas partes de tamanhos iguais [Portella et al. 2015, Silva et al. 2015, Azevedo 2016]. Portella et al. 2015 ilustram a mediana com os seguintes exemplos:

- Dado o conjunto {3, 4, 4, 5, 6, 8, 8, 8, 10} de tamanho n impar ($n = 9$), o valor central que divide o conjunto em duas partes iguais é 6;
- Dado o conjunto {5, 6, 7, 9, 11, 12, 13, 17} de tamanho n par ($n = 8$), os dois valores que dividem o conjunto em duas partes iguais são 9 e 11. Portanto, a mediana desse conjunto é 10 (média aritmética simples de 9 e 11).

A moda é a última medida de tendência central apresentada neste trabalho. Dado um conjunto de tamanho n tal que $x1, x2, ..., xn$, a moda indica o valor mais frequente encontrado neste conjunto [Silva et al. 2015, Portella et al. 2015]. A moda tem características únicas em relação às outras medidas de tendência central. É possível que um conjunto de dados não apresente uma moda. Isso ocorre quando todos os valores possuem a mesma frequência. Além disso, um conjunto de dados pode ter mais de uma moda quando mais de um valor possuir a maior frequência entre os dados [Portella et al. 2015]. Portella et al. 2015 apresentam os casos exemplos para exemplificar a moda.

- Dado o conjunto {1, 1, 3, 3, 5, 7, 7, 7, 11, 13}, o valor da moda é 7.
- Dado o conjunto {3, 5, 8, 11, 13, 18}, não existe um valor moda (amodal).
- Dado o conjunto {3, 5, 5, 5, 6, 6, 7, 7, 7, 11, 12}, existem duas modas, sendo os valores 5 e 7 (bimodal).

Outro conjunto de medidas muito importantes são as medidas de dispersão. Amplitude total, desvio médio, variância e desvio padrão são as medidas de dispersão apresentadas neste trabalho. Essas métricas complementam as medidas de tendência central apresentadas anteriormente, fornecendo outras formas de representação dos dados. A amplitude total (Fórmula 2) de um conjunto de dados é dado pela diferença entre o maior ($X_{máximo}$ na Fórmula 2) e o menor ($X_{mínimo}$ na Fórmula 2) valor do conjunto de dados [Silva et al. 2015, Portella et al. 2015, Azevedo 2016]. Silva et al. 2015 ilustram o conceito da amplitude total com o seguinte exemplo. Dado o conjunto {30, 45, 48, 62, 72}, a amplitude total é 42 ($72-30$). A amplitude térmica é um exemplo comum do uso da amplitude total. Como a amplitude total não desconsidera os valores intermediários, é sempre importante acrescentar à análise dos dados métricas complementares [Silva et al. 2015].

$$Amplitude\ Total = X_{máximo} - X_{mínimo} \tag{2}$$

O desvio médio absoluto é uma métrica capaz de complementar a amplitude total, pois o desvio médio absoluto utiliza todas as informações disponíveis no conjunto de dados para representá-lo [Silva et al. 2015]. A Fórmula 3 apresenta o modo para calcular o desvio médio absoluto. O desvio médio absoluto é dado pela soma do valor absoluto da subtração de todos os elementos (x_i) pela média do conjunto de dados ($\bar{x}$). Esse valor é divido pela quantidade de elementos (n) do conjunto de dados [Silva et al. 2015]. Silva et al. 2015 ilustram o desvio médio absoluto com o seguinte exemplo. Dado o conjunto de dados {1, 2, 3, 4, 5}, o desvio médio absoluto é 1,2. A média é dado por $\bar{x} = (1+2+3+4+5)/5 = 3$. A seguir, a média é subtraída de cada item do conjunto de dados e o valor absoluto é somado e obtém-se o valor 6 ($|1-3|+|2-3|+|3-3|+|4-3|+|5-3| = 6$). O valor 6 é divido por 5 (total de elementos do conjunto), obtendo-se o valor de 1,2.

$$Desvio\ médio\ absoluto = \frac{\sum |x_i - \bar{x}|}{n} \tag{3}$$

A variância (Fórmula 4) é outra medida de dispersão que utiliza todo o conjunto de dados. A variância indica dispersão dos dados em torno de sua média [Silva et al. 2015, Portella et al. 2015]. Para obter a variância de um conjunto de dados, basta somar o

quadrado da diferença entre cada elemento do conjunto com a média e dividi-lo pelo total de elementos do conjunto [Silva et al. 2015,Portella et al. 2015]. Silva et al. 2015 ilustram a variância com o seguinte exemplo. Dado o conjunto {6, 8, 7, 4, 10}, a variância é 4. Pois, a média ($\bar{x}$) é 7, a soma do quadrado da diferença entre cada elemento do conjunto com a média ($\sum_{i=1}^{n}(x_i-\bar{x})^2$) é 20 ($(6-7)^2+(8-7)^2+(7-7)^2+(4-7)^2+(10-7)^2=20$). O valor de 20, dividido pelo total de elementos ($n=5$), indica que a variância é 4.

$$Variância\ (s^2) = \frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n} \tag{4}$$

O desvio padrão é uma das medidas de dispersão mais utilizadas na literatura [Silva et al. 2015]. Assim como a variância, o desvio padrão também utiliza todos os dados para quantificar a dispersão dos dados em relação à média. Para obter-se o desvio padrão, basta aplicar a raiz quadrada sobre o valor da variância, assim como apresentado na Fórmula 5 [Portella et al. 2015]. No caso do exemplo anterior, o desvio padrão do conjunto de dados {6, 8, 7, 4, 10} é 2 ($\sqrt{s^2}=\sqrt{4}$).

$$DesvioPadrão\ (s) = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n}} = \sqrt{s^2} \tag{5}$$

1.2.2.4. Métricas de avaliação

Ao fim do processo de treinamento dos algoritmos de AM é imprescindível avaliar o desempenho. Em geral, os algoritmos de aprendizado que realizam tarefas de classificação são submetidos a testes para identificar o rótulo real de observações não utilizadas no treinamento. A partir desse teste é possível extrair a matriz de confusão. Para problemas com duas classes, a matriz de confusão possui duas linhas e duas colunas (Tabela 1.1). As classes são chamadas genericamente de positivo e negativo. A partir dos resultados apresentados na matriz de confusão é possível identificar algumas métricas para mensurar a qualidade do modelo. A primeira métrica é a quantidade de verdadeiros positivos (VP) que o algoritmo gerou. Para gerar um VP é necessário que o sistema rotule uma observação como pertencente a classe positiva e o rótulo real também seja positivo. A segunda métrica é o verdadeiro negativo (VN). Similar ao VP, o VN acontece quando o modelo classifica corretamente uma observação da classe negativa. VP e VN são os dois casos de acertos, porém a matriz de confusão também apresenta os erros. O falso positivo (FP) ocorre quando o modelo de AM rotula uma observação com o rótulo positivo, mas o rótulo real é negativo. O falso negativo (FN) ocorre quando o rótulo real é positivo, mas o algoritmo de AM o rotula como negativo.

Tabela 1.1: Composição da Matriz de Confusão

Matriz de confusão		Classe real	
		Positivo	Negativo
Classe hipotética	Positivo	VP	FP
	Negativo	FN	VN

Uma das métricas de avaliação mais utilizadas é a acurácia. A acurácia divide o

total de acertos positivos (VP) e negativos (VN) pelo total de observações da base (Fórmula 6). A acurácia é uma métrica adequada para quantificar a qualidade dos modelos de AM, mas em alguns casos pode ser interpretada equivocadamente. Alguns problemas, especialmente em cibersegurança, são naturalmente desbalanceados. Por exemplo, antes do lançamento dos ataques DDoS, a quantidade de *bots* é menor que a quantidade de dispositivos normais. Depois do início do ataque, a quantidade de tráfego gerado pelos *bots* supera a quantidade de tráfego gerado pelos usuários normais. Assim, apresentar acurácias próximas a 100% não garante que o modelo de AM é adequado. Pois ele pode ter errado a rotulação de todas as observações da classe minoritária. A Tabela 1.2 apresenta um exemplo de um modelo de AM que atingiu acurácia de 99% em um problema desbalanceado. No exemplo da figura existem 100 observações, 99 da classe positiva e uma da classe negativa. O modelo de AM rotulou todas as observações na classe positiva. Seguindo a Fórmula 6 esse modelo tem 99% de acurácia $(99+0/(99+1+0+0) = 99\%)$. Porém, a única observação da classe minoritária, classe negativa, foi incorretamente rotulada.

$$Acurácia = \frac{VP+VN}{VP+FP+FN+VN} \tag{6}$$

Tabela 1.2: Matriz de Confusão para um Exemplo com Classes Desbalanceadas

Matriz de confusão		Classe real	
		Positivo	Negativo
Classe hipotética	Positivo	99	1
	Negativo	0	0

Deste modo, é oportuno complementar a análise dos algoritmos de AM com outras métricas. A Fórmula 7 apresenta o modo para calcular a precisão de um modelo de AM. A precisão é obtida a partir da divisão dos VP com a soma dos VP com os FP. O valor de 100% para a precisão indica que todas as observações rotuladas pelo modelo de AM como sendo da classe positiva realmente eram da classe positiva. Assim, a precisão avalia o quanto o modelo de AM é preciso quanto a classificação das observações da classe positiva. É possível obter a precisão para a classe negativa, para isso basta usar substituir os termos VP e FP por VN e FN respectivamente.

Outra métrica comumente utilizada é o *recall*. Algumas traduções apresentam o *recall* como revocação ou sensibilidade. Para evitar interpretações incorretas, este trabalho utiliza o termo em inglês. O *recall* complementa a precisão analisando a relação entre todas as observações do tipo positivo e quantas observações do tipo positivo o modelo de AM rotulou corretamente. Com o *recall* é possível verificar o quão sensível às observações da classe positiva o modelo é. O valor de 100% no *recall* indica que o modelo acertou todas as rotulações para a classe positiva. A Fórmula 8 apresenta o modo para calcular o *recall*. Para obter o *recall* para a classe negativa basta substituir os termos VP e o FN por VN e FP respectivamente. Em geral, obter altas taxas de precisão e *recall* é o objetivo dos desenvolvedores, porém pode não ser uma tarefa trivial.

$$Precisão = \frac{VP}{VP+FP} \tag{7}$$

$$Recall = \frac{VP}{VP+FN} \tag{8}$$

A métrica *F1-score* e a área sob a curva (do inglês *area under the curve* - AUC) *receiver operating characteristic* são outras métricas usadas para avaliar os modelos de AM. A *F1-score* foi desenvolvida para facilitar a visualização da relação entre precisão e *recall*. O *F1-score* é a média harmônica entre precisão e *recall* (Fórmula 9). Assim, o *F1-score* apresenta em uma única métrica um bom indicativo sobre qualidade do modelo. A AUC também pode complementar a análise dos resultados. O valor de AUC igual a 1 significa que o modelo de aprendizado classificou todas as amostras corretamente. É necessário criar a curva ROC para calcular a métrica AUC. A curva ROC é baseada em diferentes valores de limite, taxas de verdadeiros positivos e falsos positivos. Portanto, a AUC condensa a relação entre limiares, taxa de verdadeiros positivos e taxa de falsos positivos em apenas uma medida.

$$F1 - score = 2. \frac{Precisão \,.\, recall}{Precisão + recall} \tag{9}$$

A métrica *kappa* (κ), também conhecida como *kappa* de Cohen é uma métrica que indica a concordância entre dois tomadores de decisão [Cohen 1960]. A Fórmula 10 define o modo de cálculo da *kappa*. Onde o termo P_0 indica a proporção de concordância dos tomadores de decisão observada e o P_e indica a proporção de concordância dos tomadores de decisão esperada [Cohen 1960]. Landis e Koch 1977 propuseram um guia para interpretar os resultados da métrica *kappa* (Tabela 1.3). Onde, valores menores que 0 indicam um nível de concordância pobre e valores entre $[0,81$ e $1,00]$ indicam um nível de concordância quase perfeito [Landis and Koch 1977]. Em AM a métrica *kappa* pode ser utilizada para comparar o resultado (matriz de confusão) de dois modelos (tomadores de decisão). Assim, é possível estimar o nível de concordância dos modelos. Contudo, a literatura indica que a métrica *Kappa* pode apresentar comportamentos indesejados em bases de dados desbalanceadas [Delgado and Tibau 2019]. Portanto, é importante usá-la em conjunto com outras métricas. Por fim, a biblioteca Scikit-learn implementa o cálculo da métrica *kappa*[7].

$$Kappa(\kappa) = \frac{P_0 - P_e}{1 - P_e} \tag{10}$$

Valor *Kappa*	Força do Acordo
< 0,00	Pobre
0,00-0,20	Pouco
0,21-0,40	Justo
0,41-0,60	Moderado
0,61-0,80	Substancial
0,81-1,00	Quase perfeito

Tabela 1.3: Guia de interpretação do valor da métrica *kappa* [Landis and Koch 1977]

As métricas de avaliação citadas anteriormente demandam dados rotulados durante os testes para ser possível quantificá-las (calculá-las). Contudo, ao usar o aprendizado de máquina não supervisionado, é plausível que os rótulos originais não estejam disponíveis para que essas métricas possam ser calculadas. Assim, a literatura propõe

[7] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.cohen_kappa_score.html

métricas para avaliar a qualidade da clusterização realizada pelo aprendizado de máquina não supervisionado. O índice de silhueta (do inglês, *silhouette index*), o índice de Calinski–Harabasz e o índice de Davies–Bouldin são métricas clássicas para avaliar o resultado do aprendizado de máquina não supervisionado. A literatura evoluiu esses índices clássicos propondo novos índices como o S_Dbw e o CDbw [Liu et al. 2010].

1.3. Estado da Arte da Ciência de Dados em Cibersegurança

A interrelação entre "cibersegurança" e a "ciência de dados" representa um avanço significativo na proteção de sistemas e redes. À medida que a quantidade de dados gerados e coletados aumenta exponencialmente, são necessárias novas ferramentas e metodologias para extrair informações desses dados e se beneficiar. A ciência de dados oferece o caminho, permitindo uma análise mais profunda e precisa dos dados aplicada para encontrar padrões, reconhecer ou prever comportamentos relacionados às ameaças cibernéticas. Essa seção apresenta como a literatura utiliza a ciência de dados na cibersegurança. São detalhadas as soluções essenciais que formam a base da ciência de dados para a cibersegurança, incluindo aprendizado de máquina, aprendizado estatístico, coleta e pré-processamento de dados, técnicas a para engenharia de *features*, análise de dados e visualização. Na coleta e pré-processamento de dados, são abordadas as etapas fundamentais para a obtenção e organização de informações cruciais em cibersegurança. Além disso, serão discutidas as fontes de dados disponíveis na área, assim como, as ferramentas e técnicas especializadas para coletar esses dados. As próximas subseções seguem as etapas da ciência de dados: **aquisição de dados**, **preparação dos dados**, **pré-processamento de características**, **visualização dos dados** e **análise dos dados**. Por fim, apresentamos a aplicação de técnicas de aprendizado de máquina em cibersegurança.

1.3.1. Aquisição de dados

Usar os dados ideais auxilia na utilização de ciência de dados na cibersegurança. A aquisição de dados do tráfego de rede é bem difundida na literatura e é um dos tipos de dados de entrada usados na aplicação de ciência de dados em cibersegurança. Porém, existem outros tipos de dados para tal propósito como logs de sistemas e até mesmo textos provenientes de mídias sociais. Em Wang e Zhang 2017, por exemplo, os autores propõem uma solução que monitora textos relevantes em redes sociais, como o Twitter, para prever a probabilidade de ataques acontecer no futuro. O estudo de Jog et al. 2015 propõe uma solução para coletar dados do tráfego de rede distribuidamente para detectar ataques DDoS. A solução é instalada em pontos estratégicos da infraestrutura da vítima, analisando o tráfego e prevendo quando pode ocorrer uma sobrecarga. A abordagem ótima identifica todos os caminhos possíveis que o tráfego de rede toma para chegar ao servidor vítima. Com essas informações, o algoritmo seleciona os dispositivos com a maior cobertura de rede possível. Embora esta abordagem identifique a melhor combinação de dispositivos para instalar a solução, ela não é recomendada para redes maiores, por requerer muito processamento computacional para identificar todos os caminhos possíveis. Os autores propuseram duas outras abordagens, a abordagem *Maximum-Coverage-Node-First* (MCNF) e a abordagem *Weak-Path-First* (WPF) para selecionar os melhores nós utilizando menos processamento. Ambas as abordagens utilizam menos processamento para definir os ideais para instalar a solução proposta.

O estudo de Liu et al. 2015 utiliza dados coletados em listas de reputação e eventos de segurança para prever ocorrências de ataques cibernéticos. Os autores coletaram dados de 11 listas de reputação entre janeiro de 2013 e fevereiro de 2014. Usando os endereços IP de cada dia, os autores identificaram os Sistemas Autônomos (ASs) e o prefixo do *Border Gateway Protocol* (BGP) relacionado aos endereços IP. Os autores coletaram os eventos de segurança relatados no site https://www.hackmageddon.com/ e identificaram os nomes de domínio das vítimas dos ataques e encontraram os prefixos BGP relacionados aos nomes de domínio da maioria dos eventos. Os autores selecionaram os eventos de segurança em outubro de 2013 e identificaram os prefixos BGP relacionados a esses eventos de segurança para definir a base de treinamento. Os autores pesquisaram todo o histórico entre janeiro e setembro de 2013 para o prefixo BGP e extraíram a duração e a frequência no qual o prefixo BGP contém endereços IP nas listas de reputação. Os resultados indicam prever a ocorrência de ataques com uma média de VP de 69%.

O estudo de Sapienza et al. 2018 propõe uma solução para prever eventos relacionados à cibersegurança. A coleta de dados consiste em buscar informações em três fontes de dados diferentes. Os autores coletaram os tweets de 69 especialistas em segurança cibernética usando a API oficial do Twitter. Além do Twitter, os autores selecionaram 290 blogs de segurança para coletar informações sobre vulnerabilidades, explorações e outros problemas em segurança cibernética para enriquecer os dados coletados no Twitter. A última fonte de dados são os fóruns da *dark web*, onde os autores selecionaram 263 sites diferentes. Os autores utilizam os dados coletados no Twitter e blogs de segurança como entrada para geração de alertas. A solução então analisa os dados coletados para remover termos duplicados. Quando encontra um novo termo, a solução emite um aviso que pode representar um ataque futuro. A solução mostrou 81% de acurácia na detecção de eventos relacionados à segurança cibernética.

1.3.2. Preparação de dados

Além de coletar os dados, preparar os dados otimiza e potencializa os resultados da aplicação de ciência de dados. O Auto-Sklearn [Feurer et al. 2015] é *framework* AutoML (Subseção 1.2.2.2) preocupado com a preparação dos dados. Auto-Sklearn é baseado no Scikit-learn e possui 14 métodos de preparação de dados, quatro técnicas de pré-processamento de características e 15 algoritmos de classificação. O trabalho de Araujo et al. 2023 propõe o sistema ANTE que utiliza o Auto-Sklearn para pré-processar o tráfego de rede e detectar diferentes tipos de *botnets*. A proposta de Araujo et al. 2023 seleciona autonomamente o pipeline de AM mais apropriado para cada *botnet*. O pipeline de AM do ANTE envolve três estratégias. A primeira estratégia resolve o problema de dados ausentes, sendo chamada de estratégia de imputação. A segunda estratégia para melhorar os dados de treinamento é o redimensionamento. Alguns algoritmos de AM mostram melhores resultados se os dados forem representados em certas escalas, como representar os valores dos atributos no intervalo entre zero e um. A terceira estratégia para melhorar os dados de treinamento é o pré-processamento de recursos. A seleção de atributos visa remover atributos de baixa discriminação e melhorar o funcionamento geral do ANTE. Por fim, o modelo consome os dados preparados.

Olabelurin et al. 2015 propõem a análise de alertas criados por *Intrusion detection systems* (IDSs) para antecipar ciberataques. A proposta possui três fases: pré-

processamento, construção do modelo e detecção de ataques. No pré-processamento, a solução transforma alertas de diferentes IDSs em objetos padronizados na fase de pré-processamento. A solução coleta a descrição dos alertas, nível de prioridade, protocolo, informações do sensor, IP e porta de origem/destino, hora e tipo. Durante a fase de construção do modelo, a solução calcula a entropia dos dados pré-processados para medir a uniformidade dos dados. Os autores escolheram K-means para compor a solução, pois o K-means pode encontrar clusters de formato esférico e convergir rapidamente. A fase de construção do modelo termina quando o K-means agrupa os alertas do IDS com base na entropia. Na fase de detecção, a solução verifica se os clusters definidos pelo K-means são normais ou maliciosos; para isso, a solução calcula a entropia média de cada cluster. Se a entropia média for próxima de zero, o cluster é malicioso. Se a entropia média estiver próxima dos maiores valores da base, o cluster é normal.

Big data em cibersegurança é uma realidade. A literatura endereça trabalhos tendo como entrada grandes volumes de dados para detectar intrusão e anomalias, detectar *spam* e *spoofing*, detectar *malware* e *ransomware*, analisar a segurança de códigos e a segurança na nuvem. Em AlMahmoud et al. 2019, os autores propuseram uma plataforma colaborativa de detecção de spam baseado em big data. A proposta possui três componentes principais: o ofuscador, o classificador e o detector de anomalias. O ofuscador possibilita o processamento paralelo dos e-mails sem ser necessário que a plataforma analise o conteúdo da mensagem. Assim, a plataforma proposta evita ferir a privacidade do e-mail original. O classificador analisa o resultado da mensagem ofuscada e agrupa dados similares. O detector de anomalias verifica o tamanho e a taxa de crescimento dos grupos de e-mails para identificar spam. Em De Paola et al. 2018, os autores apresentam um sistema de detecção de *malware* baseado em nuvem e *big data* para classificação rápida de arquivos executáveis. Os usuários enviam os arquivos executáveis para serem analisados pelo sistema. O sistema aplica filtros de hash para identificar novos arquivos e cópias dos arquivos recebidos anteriormente. A detecção de *malware* é baseada em uma rede profunda que utiliza apenas uma parte do arquivo. O objetivo desta análise é, por meio de um processo leve, obter altas acurácias na identificação dos *malwares*. Caso esse processo estenda-se por muito tempo ou gere uma identificação pouco confiável, o sistema começa uma análise mais robusta que utiliza mais memória e processamento. Assim, o sistema resolve rapidamente a identificação dos *malwares* mais simples e provê atenção aos *malwares* que precisam de mais atenção.

1.3.3. Pré-processamento de Características

Para melhorar os resultados nas tarefas relacionadas com cibersegurança, soluções baseadas na ciência de dados precisam pré-processar as características após adquirir (Subseção 1.3.1) e preparar os dados (Subseção 1.3.2). A criação de novas características e a seleção das características são linhas de pesquisa difundidas para realizar o pré-processamento dos dados. Em Neira et al. 2023b, os autores aplicam a teoria dos sinais precoces de alerta sobre o tráfego de rede para gerar novas características. Essas novas características realçaram os sinais da preparação dos ataques DDoS e proporcionaram a detecção antecipada dos ataques DDoS. Similarmente, os trabalhos de Albano et al. 2023 criam novas características baseadas no tráfego de rede usando a teoria dos padrões ordinais para identificar *botnets* e predizer ataques DDoS, respectivamente.

A seleção de características é uma linha de pesquisa mais difundida que a criação de novas características. Existem revisões focadas em analisar as várias técnicas de seleção de características em cibersegurança [Maldonado et al. 2022]. A computação bio inspirada é uma fonte de soluções para a seleção de características em cibersegurança. Najafi Mohsenabad e Tut 2024 comparam algoritmos de seleção de características baseados em Colônia Artificial de Abelhas, Otimização da Colônia de Formigas e o Algoritmo de Polinização de Flores visando a detecção de diferentes ciberataques. O algoritmo baseado na Otimização de Colônia de Formigas proporcionou a maior acurácia, atingindo 98,8%.

Em Borges et al. 2024 a seleção de atributos para a predição de ataques considerou atributos multifacetados para oferecer uma visão mais robusta da variabilidade dos dados. Dessa forma, os autores consideram informações de diferentes camadas do protocolo TCP/IP. O estudo de Muhammad et al. 2020 propõe uma solução focada na seleção de características para detectar *botnets* durante o estágio inicial de comunicação C&C. Para selecionar as características, os autores utilizaram o Principal Component Analysis (PCA) e o Information Gain. No final do processo de seleção de características, a solução obteve as 40 características mais representativas. Usando 37 das 40 características e o *random forest*, os autores obtiveram 97,8% de acurácia para detectar *botnets* e usando todas as 40 características, a solução proposta atingiu uma acurácia de 99%.

1.3.4. Visualização de dados

A visualização dos dados é uma preocupação cada vez mais recorrente na literatura [Noel et al. 2016, Raynor et al. 2023, de Neira et al. 2023]. A visualização correta dos dados auxilia na construção das soluções baseadas na ciência de dados e na tomada de decisão realizada pelos administradores de rede e equipes de segurança frente a ameaças cibernéticas. O Matplotlib[8], Seaborn[9], D3[10], Tableau[11], e o Bokeh[12] são exemplos de ferramentas genéricas que auxiliam na visualização dos dados por meio da geração de gráficos e figuras.

Em Noel et al. 2016, os autores apresentam o CyGraph, uma ferramenta para análise, visualização e gerenciamento de conhecimento sobre ciberataques. Durante a execução, o CyGraph coleta alertas de intrusão e os correlaciona com caminhos de vulnerabilidade conhecidos. O CyGraph então processa os eventos de rede e outras saídas de sensores, incluindo a captura de pacotes. Isso inclui quaisquer atributos de rede que potencialmente contribuem para o sucesso do ataque, como topologia de rede, regras de firewall, configurações de host e vulnerabilidades. O CyGraph funde os dados coletados para produzir um modelo unificado baseado em grafo. Com o grafo, o CyGraph sugere os melhores cursos de ação para responder a ciberataques, ajuda a priorizar vulnerabilidades expostas e auxilia em análises após o ataque.

Neira et al. 2023a propuseram uma solução focada na explicabilidade para a iden-

[8] https://matplotlib.org/
[9] https://seaborn.pydata.org/
[10] https://d3js.org/
[11] https://www.tableau.com/
[12] https://bokeh.org/

tificação da preparação de ataques DDoS. A solução apresenta o tráfego de rede em figuras de três dimensões, referentes a três atributos coletados no tráfego de rede. Utilizando k-means, a proposta de Neira et al. 2023a divide o tráfego de rede em dois grupos. O Grupo 1 é composto pelo tráfego de rede originado por usuários normais. O Grupo 2 é composto especialmente pelo tráfego de rede gerado por *bots*. Esse tipo de visualização dos dados proporciona aos administradores de rede e as equipes de segurança o entendimento da evolução das ameaças cibernéticas. Esse entendimento auxilia na tomada das decisões necessárias para evitar danos causados pelos ciberataques.

A visualização dos dados é um fator importante na segurança dos BGPs [Raynor et al. 2023]. O BGPlay[13] [Di Battista et al. 2004] é uma das ferramentas de visualização de dados relacionados com BGPs mais difundidas na literatura [Raynor et al. 2023]. O BGPlay apresenta um grafo com as conexões entre os ASs. O BGPlay coleta as informações do roteamento em fontes de informações de roteamento bem conhecidas, constantemente atualizadas e disponíveis na Internet [Raynor et al. 2023]. Em Papadopoulos et al. 2013, os autores propuseram o BGPfuse, uma ferramenta para a visualização e análise de anomalias de mudança de caminho do BGP baseado em grafos. O BGPfuse usa características dos BGPs capazes de mensurar o grau de anomalia de cada evento de mudança de caminho. O BGPfuse usa diferentes formas para apresentar os grafos e representar em profundidade as relações entre os ASes envolvidos. Além disso, BGPfuse combina diferentes grafos para realçar semelhanças estruturais entre todos os grafos de recursos individuais.

1.3.5. Análise de dados

A análise de dados pode ser feita de diferentes formas. Aprendizado de máquina, deep learning, modelos estatísticos e soluções baseadas em cadeias de Markov são alguns exemplos presentes na literatura. O estudo de Pelloso et al. 2018 utiliza a teoria da metaestabilidade (modelos estatísticos) para identificar sinais antes do início do ataque. Em Leros e Andreatos 2019, os autores propõem uma solução para prever o tráfego de rede próximo ao real para detectar ataques DDoS. A solução possui um módulo que utiliza observações anteriores para treinar o modelo autorregressivo incrementado com o filtro de Kalman.

No estudo de Ali and Al-Shaer 2013, os autores propõem uma solução para utilizar cadeias de Markov para prever eventos de segurança cibernética. A solução realiza a modelagem da cadeia de Markov com base em logs de aplicação. Assim, a solução identifica a possibilidade de existir variações de estado representando mudanças para um estado de ataque. O estudo de Abaid et al. 2016 visa identificar e modelar o comportamento típico de *botnets* em uma cadeia de Markov. Os autores propuseram uma metodologia para prever ataques com base na probabilidade de evolução do estado atual para um estado de ataque em breve. Por fim, o estudo de Holgado et al. 2020 propõe a utilização de alertas produzidos pelo IDS para prever ataques utilizando o Modelo Oculto de Markov (HMM).

1.3.6. Aprendizado de máquina em cibersegurança

As ferramentas de automação e orquestração são usadas para melhorar a resposta a incidentes [Serpeloni et al. 2024], e a inteligência artificial para facilitar as investigações fo-

[13] https://stat.ripe.net/widget/bgplay

renses [Dunsin et al. 2024]. A combinação de métodos de análise estática e dinâmica tem se mostrado extremamente eficaz na detecção de *malwares*. A análise estática envolve a inspeção do código do *malware* sem executá-lo, utilizando técnicas como descompilação e análise de assinatura para identificar padrões conhecidos de comportamento malicioso. Esta abordagem permite uma identificação rápida e eficiente de *malwares* conhecidos. Por outro lado, a análise dinâmica executa o *malware* em um ambiente controlado para observar seu comportamento em tempo real, detectando atividades maliciosas que podem não ser evidentes na análise estática, como a exploração de vulnerabilidades ou a comunicação com servidores de comando e controle. Redes Generativas Adversariais (GANs) complementam essas técnicas ao gerar exemplos sintéticos de *malware* que são utilizados para treinar e melhorar os sistemas de detecção. As GANs consistem em duas redes neurais competindo entre si: uma gera novos exemplos (gerador) e a outra tenta distinguir entre exemplos reais e gerados (discriminador). Esse processo resulta em exemplos de *malware* altamente variados e realistas, que ajudam a fortalecer os sistemas de detecção contra novos e sofisticados ataques.

No estudo de Ali and Al-Shaer 2013, os autores utilizaram dados de logs para predizer eventos de segurança cibernética. A solução realiza a modelagem da cadeia de Markov com base nos *logs* de aplicação gerados no sensor. Os autores coletaram os *logs* de eventos durante o período de duas semanas e os processaram centralizadamente. O diferencial do estudo é que os autores identificaram que, por meio da modelagem da cadeia de Markov, eles predizem a mudança entre os estados com um erro menor que 2%.

Em Zargar et al. 2013, os autores classificaram os mecanismos de defesa DDoS em três categorias: prevenção, detecção e mitigação (resposta). Ambientes como Cloud, IoT e redes definidas por software (SDN) possuem mecanismos de defesa especializados para suportar características específicas de cada ambiente. O objetivo final do combate a um ataque DDoS é evitá-lo [Zargar et al. 2013]. A prevenção deve ocorrer antes do ataque ser lançado para evitar ou reduzir os efeitos negativos [Somani et al. 2017, de Neira et al. 2023, Zargar et al. 2013]. Em Somani et al. 2017, os autores apresentam abordagens adicionais para prevenir ataques DDoS, como resposta a desafios, servidores/portas ocultos, acesso restritivo e limites de recursos [de Neira et al. 2023].

Os estudos que consideram a predição do ataque podem ser classificados em camadas como mostra a Figura 1.12. A primeira camada classifica os estudos quanto ao aspecto temporal. A segunda camada utiliza arquiteturas centralizadas versus arquiteturas distribuídas como critério de classificação. Os estudos classificados na categoria de curto prazo seguem ambas as abordagens, enquanto os da categoria de longo prazo, até agora, seguem apenas uma abordagem centralizada. Na terceira camada, os estudos classificados na categoria de curto prazo e centralizados seguem quatro possíveis aspectos metodológicos: Aprendizado de Máquina/Aprendizado Profundo, Baseado em Markov, Modelo puramente Estatístico e Híbrido. O único estudo classificado como categoria de curto prazo e distribuído utiliza Modelos Estatísticos como aspecto metodológico. Os estudos classificados na categoria de arquitetura centralizada e de longo prazo utilizam Machine Learning/Deep Learning, Modelos Estatísticos ou uma abordagem híbrida baseada em Machine Learning e Modelos Estatísticos. Por fim, os estudos utilizam tráfego de rede, alerta do Sistema de Detecção de Intrusões (IDS), logs de aplicativos e dados coletados em fontes de dados externas na camada de aspecto de dados.

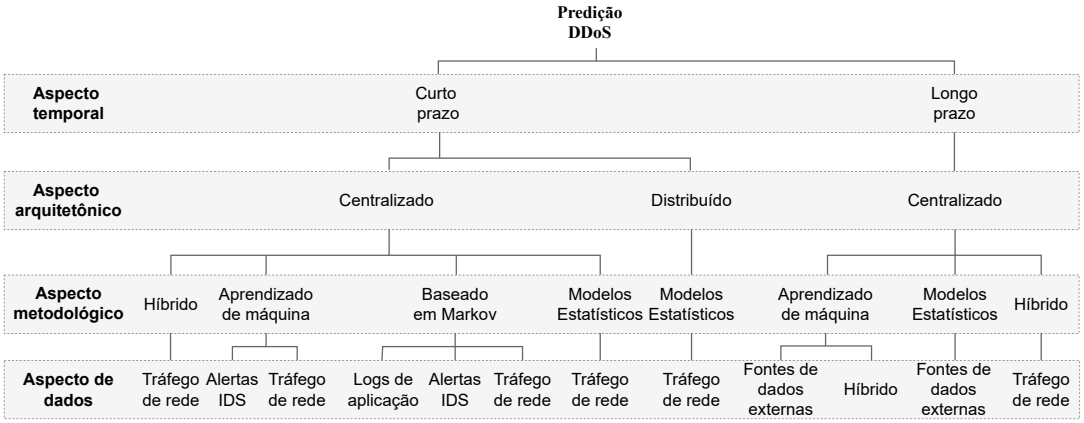


Figura 1.12: Classificação das Soluções para a Predição de Ataque DDoS

1.4. Ambientes Experimentais, Ferramentas e Datasets

Os ambientes experimentais e ferramentas são amplamente utilizados como plataformas de suporte ao desenvolvimento e/ou avaliação de propostas em diversas áreas de pesquisa, como, por exemplo, prevenção, predição, detecção e mitigação de ataques cibernéticos. Além disso, estes ambientes experimentais e ferramentas, podem ser utilizados em ambientes educacionais e de treinamentos para fornecer cursos práticos [Prates Jr et al. 2021]. Projetar ferramentas é uma tarefa desafiadora e com alto custo devido às respectivas especificidades. O que contribui para a escassez e caráter privado das mesmas. O projeto MCTI/CGI.br/FAPESP MENTORED vem buscando desenvolver um ambiente experimental para estudos relacionados a ataques DDoS considerando a existência de dispositivos representantes da Internet das Coisas[14].

Cyber Ranges são ambientes de simulação sofisticados e de alta fidelidade projetados para facilitar o treinamento, os testes e as pesquisas avançadas em cibersegurança. Dentre os vários *Cyber Ranges* existentes na literatura, a equipe de autores deste capítulo teve acesso ao longo da preparação desse documento ao *Airbus CyberRange*. Desenvolvido pela *Airbus*, esta plataforma oferece um ambiente realista e controlado onde os profissionais de cibersegurança aprimoraram suas habilidades, avaliam tecnologias de segurança e realizam avaliações de segurança abrangentes[15]. O *CyberRange* está equipado com ferramentas e tecnologias de última geração, permitindo a replicação de infraestruturas cibernéticas complexas e a simulação de uma ampla gama de ameaças cibernéticas e cenários de ataque [Grimaldi et al. 2023]. Os principais objetivos do *Airbus CyberRange* são apoiar pesquisas de ponta em cibersegurança, fornecendo um ambiente versátil para simular e analisar novas ameaças, vetores de ataque e mecanismos de defesa. Além disso, ele oferece uma plataforma de testes e avaliações para diversas soluções de cibersegurança sob condições controladas e realistas. A avaliação e validação de segurança permite que as organizações conduzam avaliações de segurança completas e validem suas estratégias e arquiteturas de defesa cibernética contra ataques simulados.

[14]Site do projeto MCTI/CGI.br/FAPESP MENTORED: http://mentored.dcc.ufmg.br
[15]https://www.cyber.airbus.com/products/cyberrange

O simulador *Cyber Range* da Cyberbit[16] é considerado o mais avançado simulador de ataques cibernéticos do mundo, sendo desenvolvido nas Forças de Defesa de Israel, com direitos exclusivos no Brasil da CECyber. O simulador reúne as principais ferramentas de mercado e oferece um ambiente controlado onde os usuários podem experimentar e responder a ameaças e incidentes cibernéticos do mundo real. As simulações realistas compreendem ataques cibernéticos, incluindo *ransomware*, DDoS, *phishing* e ameaças persistentes avançadas. O ambiente de rede replica infraestruturas de TI reais, incluindo redes corporativas e serviços de nuvem. Portanto, o *Cyber Range* da Cyberbit se destaca como uma plataforma de treinamento abrangente e eficaz que não apenas aprimora as habilidades individuais, mas também fortalece a prontidão organizacional contra ameaças.

Outra plataforma projetada para treinamento e educação em cibersegurança é o *TryHackMe*[17]. Este é uma plataforma *online* interativa a qual fornece diversos laboratórios virtuais e desafios sobre vários aspectos da cibersegurança. Lançado em 2018, esta ferramenta se tornou um recurso popular tanto para iniciantes quanto para profissionais. A plataforma apresenta uma interface amigável, tutoriais guiados e um conjunto diversificado de cenários reais que simulam ataques e defesas cibernéticas, tornando-a uma ferramenta eficaz para o aprendizado prático. O *TryHackMe* proporciona uma aprendizagem acessível, democratizando o acesso à educação em cibersegurança de alta qualidade para usuários de qualquer nível de habilidade. Ele promove o desenvolvimento de habilidades ao oferecer treinamento prático por meio de cenários e desafios reais. Estes treinamentos são fundamentais para que usuários possam avançar profissionalmente obtendo certificações como, *CompTIA Security+*, *Certified Ethical Hacker* (CEH) e *Offensive Security Certified Professional* (OSCP) e avanço em suas carreiras.

No quesito emuladores, o *Common Open Research Emulator*[18] (CORE) é uma ferramenta de emulação de rede versátil e poderosa projetada para facilitar o desenvolvimento, teste e avaliação de protocolos e aplicações de rede. CORE fornece um ambiente virtual onde os usuários podem criar e experimentar topologias de rede complexas de maneira flexível e controlada, sem a necessidade de hardware físico. Ele suporta simulação e emulação de rede em tempo real, tornando-o uma importante ferramenta para pesquisadores, educadores e engenheiros de rede. O CORE apoia o desenvolvimento e teste de novos protocolos de rede, onde os pesquisadores implementam, modificam e avaliam comportamentos de protocolo simulando diferentes topologias e configurações sob diversas condições de rede. Ele ainda permite avaliar o desempenho de protocolos e aplicações em um ambiente controlado, identificando possíveis pontos fortes e fracos. Portanto, o CORE aprimora a compreensão e o avanço das tecnologias de rede.

O *Measurement Lab* (M-Lab)[19] é uma plataforma de servidor aberta e distribuída que fornece um ecossistema para a medição aberta e verificável do desempenho da rede de Internet global. Os dados coletados pelo M-lab são disponibilizados abertamente de modo a promover a pesquisa na Internet, melhorar a transparência, além de facilitar a compreensão dos problemas de desempenho e conectividade da Internet em todo o mundo. Além

[16] https://cecyber.com/plataforma-de-simulacao/

[17] https://tryhackme.com/

[18] https://www.nrl.navy.mil/Our-Work/Areas-of-Research/Information-Technology/NCS/CORE/

[19] https://www.measurementlab.net/

dos dados, todas as ferramentas de medição hospedadas pelo M-Lab são de código aberto. Ao fornecer dados e ferramentas de acesso aberto, o M-Lab permite que pesquisadores estudem vários aspectos da conectividade da Internet, como velocidade, latência e confiabilidade, sem restrições proprietárias. O M-Lab desempenha um papel importante no avanço da pesquisa da Internet, promovendo a transparência e apoiando o desenvolvimento de políticas. Além dos dados disponibilizados pelo M-lab, a literatura indica outros repositórios de código aberto para compartilhamento, publicação e arquivamento de dados de pesquisa, tal como o Harvard Dataverse gerenciado pelo *Institute for Quantitative Social Science* (IQSS) da Universidade Harvard[20]. Ele promove o compartilhamento de dados entre pesquisadores, fomentando a colaboração e permitindo a reutilização de dados. No escopo deste capítulo, este é um ambiente útil para coleta de dados a serem analisados pelas técnicas aqui descritas.

Outras importantes bases de dados sobre análise de tráfego de rede com foco explícito em cibersegurança são relevantes. Dentre elas, o KDD Cup 99 é uma das bases mais antigas e consequentemente mais usadas em pesquisa de cibersegurança. Embora focado em vários cenários de detecção de intrusão, o conjunto de dados KDD Cup 99 inclui registros de ataques DDoS. Esta base foi melhorada e disponibilizada como NSL-KDD, esta nova versão reduz a redundância dos dados para fornecer um conjunto mais balanceado [Protić 2018]. Existem, portanto, muitas críticas ao uso da base KDD Cup 99 em termos de representatividade atual dos ataques presentes nessa base. O conjunto de dados CTU-13 é uma coleção de 13 cenários de diferentes tipos de tráfego de *botnet*, proporcionando diversos padrões de ataque [21]. A base foi criada pela Universidade CTU na República Tcheca. Os dados foram coletados em um ambiente real, tornando-os altamente realistas. Ele contém vários tipos de *botnet*, como IRC, HTTP e P2P e o tráfego é rotulado como normal, *botnet* ou tráfego de segundo plano.

Outro conjunto de dados projetado para fornecer um conjunto de dados realista e rotulado para pesquisa de ataques DDoS é o UNB ISCX, da Universidade de New Brunswick. Os principais tipos de ataque DDoS, como HTTP, inundação SYN e inundação UDP. Todo o tráfego é rotulado como normal e de ataque, além disso, os dados contém informações como registro de data e hora, IP de origem, IP de destino e tamanho do pacote. Similarmente, o *Canadian Institute for Cybersecurity*, criou a base CIC-DDoS2019, a qual inclui ataques DDoS, inundação HTTP, inundação UDP e inundação SYN. O tráfego também é coletado em um ambiente realista [22]. O Centro de Análise Aplicada de Dados da Internet (do inglês, *Center for Applied Internet Data Analysis* - CAIDA)[23] fornece vários conjuntos de dados relacionados ao tráfego de rede em larga escala, incluindo ataques DDoS. Os dados são fornecidos no formato de séries temporais, garantindo a privacidade e proporcionando a análise de padrões ao longo do tempo. Contudo, os conjuntos de dados são de acesso exclusivo a pesquisadores de universidades americanas ou parceiros das mesmas.

Monitorar o tráfego de rede específico de dispositivos de Internet das Coisas, atualmente é muito valioso devido à popularização da tecnologia. Portanto, a base IoT-23,

[20] https://dataverse.harvard.edu/
[21] https://www.stratosphereips.org/datasets-ctu13
[22] https://www.unb.ca/cic/datasets/ddos-2019.html
[23] https://www.caida.org/

fornece 23 capturas de tráfego de rede de dispositivos de Internet das Coisas, incluindo tráfego malicioso. Ela é especialmente projetada para capturar ataques DDoS direcionados a dispositivos IoT [24]. Todas essas bases de dados acima descritas, são fundamentais para o avanço da pesquisa na detecção e mitigação de ataques DDoS, fornecendo aos pesquisadores dados realistas e abrangentes para testar e validar suas metodologias.

Para auxiliar no desenvolvimento de soluções próprias em cibersegurança, o Instituto Nacional de Padrões e Tecnologia (NIST) propôs o *NIST Cybersecurity Framework* (NIST CSF) [NIST 2018]. Este é um *framework* que visa melhorar a postura de cibersegurança das organizações, fornecendo uma abordagem estruturada para identificar, proteger, detectar, responder e recuperar-se de ameaças e incidentes de cibersegurança alinhados aos requisitos regulamentares. É, também, um caminho seguro e estratégico para mapear constantemente o nível de maturidade do negócio no quesito cibersegurança. O NIST CSF fornece uma linguagem comum e um conjunto de termos que ajudam a preencher a lacuna entre públicos técnicos e não técnicos. Ademais, o NIST CSF promove a melhoria contínua das práticas de cibersegurança, incentivando as organizações a avaliar e atualizar regularmente as suas estratégias e controles de cibersegurança.

É desejável que o desenvolvimento de ambientes experimentais e ferramentas forneçam aos usuários uma interface amigável e transparente. Eles devem possibilitar o monitoramento em tempo real e não intrusivo do tráfego de rede e dos recursos computacionais. Além disso, espera-se que disponibilizem visualizações de recursos tanto de forma gráfica quanto pela linha de comandos de forma clara e objetiva. Bibliotecas desenvolvidas em linguagem de programação Python fornecem um conjunto de ferramentas robusto para que um usuário qualquer possa desenvolver suas próprias soluções e/ou automatizar tarefas de cibersegurança, desde a análise de rede, interações Web e até operações criptográficas com análise de vulnerabilidades.

Dentre algumas das principais bibliotecas *Python* usadas para desenvolver ferramentas na área de cibersegurança destacam-se o *Scapy*[25] é utilizado para manipulação de pacotes e análise de tráfego de rede. Ele permite criar, enviar, receber e examinar pacotes de rede, portanto é uma ferramenta essencial para descoberta de rede, injeção de pacotes e teste de protocolos de rede. O *Python-Nmap*[26] também é uma interface para o *scanner* de rede, comumente utilizado para descoberta de rede, varredura de portas e avaliação de vulnerabilidades. Outra biblioteca importante para o desenvolvimento de ferramentas de exploração de rede e testes de penetração é o *Impacket*[27]. Ela trabalha com protocolos de rede, fornecendo acesso programático de baixo nível a protocolos como SMB, MSRPC e outros. Já o *Requests*[28], é uma biblioteca HTTP amigável, comumente usada para fazer solicitações HTTP. É útil para interagir com serviços da *web*, APIs e realizar *scraping* ou testes de penetração de aplicativos da *web*. A *BeautifulSoup*[29] também é uma biblioteca útil para análise de documentos HTML e XML, sendo particularmente útil em web

[24] https://paperswithcode.com/dataset/iot-23
[25] https://scapy.net/
[26] https://nmap.org/
[27] https://pypi.org/project/impacket/
[28] https://pypi.org/project/requests
[29] https://pypi.org/project/beautifulsoup4/

scraping. Para operações criptográficas, destaca-se a *PyCryptoDome*[30]. Esta biblioteca fornece recursos para implementar criptografia, descriptografia, hash e vários protocolos criptográficos, tornando-o vital para o desenvolvimento de aplicativos seguros e para a condução de análises criptográficas.

Ainda existem algumas das bibliotecas Python que são amplamente usadas para visualização de dados, as quais podem ser particularmente úteis para o desenvolvimento de ferramentas de cibersegurança. Dentre elas, elencam-se *Matplotlib*[31], o qual fornece um conjunto abrangente de funções para criar visualizações estáticas, animadas e interativas, suportando diversos de tipos de gráficos, O *Seaborn*[32] é construído sobre *Matplotlib* e simplifica o processo de criação de visualizações complexas além de ser eficaz para visualizar relações estatísticas. Já o *Plotly*[33] produz gráficos interativos com qualidade de publicação *online*, oferecendo suporte a vários tipos de gráficos, recursos de interatividade, sendo ideal para a criação de visualizações dinâmicas para *dashboards* de cibersegurança. Adicionalmente, tem-se o *Dash*[34], focado na construção de aplicativos web interativos combinando o poder do *Plotly* para visualizações com a flexibilidade do desenvolvimento[35]. Além destas, a *Bokeh*[36] é analogamente útil para criar visualizações interativas e em tempo real.

1.5. Descrição do Estudo de Caso Prático

Os estudos de caso buscam fornecer uma compreensão aprofundada tanto das técnicas de ataque quanto das estratégias defensivas, permitindo uma visão holística dos desafios e soluções na segurança de redes IoT. Nesta seção, os estudos de casos serão divididos em dois contextos: o do atacante e o da defesa de uma rede ilustrativa. No contexto do atacante, serão demonstrados os passos comumente utilizados para infectar dispositivos em uma rede, particularmente em ambientes de IoT. Serão abordadas as técnicas e as ferramentas empregadas para comprometer a segurança desses dispositivos e transformá-los em partes de uma *botnet*. Além disso, será demonstrado como esses dispositivos infectados são utilizados para realizar ataques de negação de serviço (DDoS) em grande escala, destacando as consequências para a rede e os serviços afetados.

No contexto da defesa, será abordado um fluxo de passos da Ciência de Dados, apresentadas nas seções anteriores, para treinar, testar e utilizar modelos preditivos que detectam ataques aos dispositivos na rede. Isso será feito por meio da análise de tráfego de rede. Serão discutidas as abordagens para coleta de dados, técnicas de pré-processamento, análise exploratória de dados e a seleção de características relevantes para a construção de modelos de aprendizado de máquina. Além disso, será avaliada a eficácia desses modelos em cenários simulados e reais, fornecendo uma visão prática sobre como essas técnicas são implementadas e ajustadas para diferentes ambientes de rede.

Na infraestrutura do estudo de caso, dez dispositivos do tipo *Raspberry Pi* são

[30] https://pypi.org/project/pycryptodome/
[31] https://matplotlib.org/
[32] https://seaborn.pydata.org/
[33] https://plotly.com/
[34] https://dash.plotly.com/
[35] https://flask.palletsprojects.com/en/3.0.x/
[36] https://bokeh.org/

utilizados para simular o funcionamento de dispositivos IoT vulneráveis. Todos esses dispositivos estarão conectados a uma mesma rede, utilizando um *switch* ou roteador, juntamente com um servidor DHCP para distribuir endereços IP automaticamente aos dispositivos. A rede incluirá outros componentes importantes, como um servidor web, que atuará como alvo principal dos ataques após a infecção dos dispositivos, e uma máquina *sniffer*, que será usada para monitorar e analisar o tráfego de rede gerado.

Para explorar a construção de defesas contra ataques de rede, serão utilizadas capturas de tráfego geradas em um ambiente similar, combinadas com técnicas avançadas de ciência de dados. O processo começará com o pré-processamento dos dados, seguido por uma análise exploratória para identificar padrões e características relevantes. Posteriormente, serão treinados e testados os modelos de aprendizado de máquina utilizando algoritmos apropriados, entre eles, *One Class SVM*, *XGBoost*, *Random Forest*, *Isolation Forest*, e até redes neurais CNN e *Autoencoders*. A eficácia dos modelos será avaliada com base em métricas de desempenho, como Acurácia, Precisão, Revocação e F1-score.

1.5.1. Ataques a Rede

Neste cenário, serão exploradas as vulnerabilidades comuns em dispositivos IoT, representados por 10 Raspberry Pi com a porta Telnet 23 aberta e mantendo senhas padrão. Este ambiente simula um dos riscos mais comuns no ambiente IoT, a falta de segurança básica nos dispositivos, que é facilmente explorada por atacantes. Inicialmente, é necessário identificar os dispositivos conectados à rede e as respectivas vulnerabilidades. Uma ferramenta *port scanner* auxilia a mapear as portas e identificar o status (aberto ou fechado) de cada uma delas. Esse passo é fundamental, pois, em um ambiente real, a identificação dos alvos e suas vulnerabilidades é a primeira fase de um ataque. A topologia da rede e a identificação dos dispositivos com a porta Telnet aberta é realizada por meio de ferramentas comuns de varredura de rede, como o Nmap. Esta etapa também permitirá entender a disposição da rede e os pontos mais vulneráveis, proporcionando uma visão clara dos possíveis alvos.

Após a identificação dos dispositivos vulneráveis, o próximo passo é realizar um ataque de força bruta para descobrir as credenciais do Telnet ou SSH, seguindo a metodologia aplicada na construção da botnet MIRAI. Pode-se utilizar uma lista de senhas padrão frequentemente usadas em dispositivos IoT e fornecidas pelos respectivos fabricantes. Isto revela importância do uso de senhas fortes e a falha de segurança crítica que é usar senhas padrão em dispositivos. Ferramentas como Hydra, Medusa ou simples scripts podem ser empregadas para automatizar esse processo e demonstrar como atacantes podem rapidamente obter acesso a dispositivos desprotegidos. Ao conseguir acesso através do Telnet ou SSH, conseguimos compreender como essa vulnerabilidade pode ser explorada em um cenário real.

Com o acesso ao dispositivo comprometido, serão instalados *malwares* nos Raspberry Pi. Essa etapa demonstra como, após obter acesso a um dispositivo, um atacante pode facilmente carregar e executar código malicioso, transformando o dispositivo em um agente de ataque. Serão utilizados *scripts* e ferramentas de *malware* comumente encontrados em ambientes de *botnet* para ilustrar este processo. Uma vez que o *malware* esteja instalado, ele pode ser programado para se comunicar com um servidor de comando e

controle (C&C), permitindo que o atacante controle remotamente o dispositivo.

Após comprometer múltiplos dispositivos, neste cenário com dez Raspberry pi, será demonstrado como esses dispositivos podem ser utilizados para lançar um ataque coordenado de negação de serviço ao servidor web alvo. Este ataque será executado utilizando ferramentas como *socket* e *requests*, que realizam o envio maciço de pacotes para sobrecarregar o servidor alvo. Essa ação ilustra como dispositivos IoT mal configurados podem ser usados em conjunto para lançar ataques significativos e disruptivos a alvos na rede. Além disso, serão discutidas as possíveis mitigações e defesas contra tais ataques, como a implementação de sistemas de detecção e prevenção de intrusões (IDS/IPS) e a importância de boas práticas de segurança na configuração de dispositivos IoT.

1.5.2. Defesas contra Ataques de Rede

Este cenário é direcionado para a análise de tráfego de rede e na detecção de ataques utilizando técnicas de ciência de dados e aprendizado de máquina. Será utilizada uma captura de tráfego de rede de um ambiente similar ao do primeiro cenário, que contém interações entre dispositivos IoT, a infecção destes por *malware* e o subsequente ataque DDoS a uma máquina alvo. O primeiro passo é transformar o arquivo de captura de tráfego de rede (.pcap) em um formato que possa ser facilmente manipulado e analisado. Para isso, o arquivo .pcap é convertido em um arquivo .csv, utilizando ferramentas como o Wireshark ou o Tshark. Essa transformação facilita a aplicação de técnicas de análise de dados, uma vez que o formato .csv é amplamente suportado por ferramentas e bibliotecas de ciência de dados.

Com os dados em formato .csv, a linguagem de programação Python e as respectivas bibliotecas, tais como, Numpy e Pandas auxiliam na extração métricas estatísticas relevantes e preparar os dados para análise. Durante esta fase, outra etapa crucial para o treinamento e teste de modelos de aprendizado de máquina é a rotulagem dos dados. Os rótulos possibilitam diferenciar o tráfego benigno do malicioso, permitindo que os modelos aprendam a reconhecer padrões associados a ataques. Além disso, os dados serão filtrados para remover entradas inconsistentes ou irrelevantes e normalizar os valores, facilitando a análise subsequente.

Em seguida, análises exploratórias nos dados procuram obter *insights* iniciais e visualizar padrões que possam indicar comportamentos anômalos ou indícios de ataque. As bibliotecas de visualização como Matplotlib e Seaborn possibilitam gerar gráficos e diagramas que revelam tendências e correlações nos dados. Esta análise ajuda a compreender melhor o conjunto de dados e a identificar quais características são mais relevantes para a detecção de ataques, como a frequência de pacotes, a distribuição de protocolos e os padrões de comunicação entre dispositivos.

Após a análise exploratória, os algoritmos de seleção de características refinam o conjunto de dados. A seleção de características é uma etapa importante para melhorar o desempenho dos modelos de aprendizado de máquina por ajudar a eliminar dados irrelevantes ou redundantes, concentrando-se nas características mais significativas para a detecção de ataques. Serão utilizadas técnicas como a análise de componentes principais (PCA) e métodos baseados em árvore, como *Random Forests*, para identificar e selecionar as características mais informativas.

Com as características selecionadas, ocorre a separação dos dados em conjuntos de treino e teste, utilizando uma abordagem de validação cruzada para garantir a robustez dos modelos. Diferentes algoritmos de aprendizado de máquina, entre eles, *One Class SVM*, *XGBoost*, *Random Forest*, *Isolation Forest*, e até redes neurais CNN e *Autoencoders*, são usados para treinar modelos que possam identificar tráfego malicioso com base nos padrões detectados. Por fim, os modelos são avaliados com base nas métricas de desempenho Acurácia, Precisão, Revocação e F1-score. Além disso, os modelos serão testados quanto à sua precisão e capacidade de generalização, garantindo que possam efetivamente detectar ataques em cenários reais.

1.6. Principais Desafios e Limitações

Um dos problemas popularmente explorados quando se refere à aplicação de Inteligência Artificial em Cibersegurança é a **detecção de anomalias**. Entretanto, apesar da diversidade de trabalhos e contribuições existentes, a detecção em tempo real com altas taxas de acertos é complexa, especialmente em sistemas cibernéticos multifacetados. É necessário desenvolver métodos que consigam distinguir entre desvios causados por *outliers* e aqueles provocados por atacantes. Isso envolve a modelagem estocástica e a aplicação de conceitos de controle estatístico, além de lidar com a raridade dos dados anômalos [Hero et al. 2023]. **Antever problemas causados pelos ciberataques** também é uma linha de pesquisa importante e em aberto. Predizer ataques e gerenciar os riscos são desafios enfrentados pela cibersegurança que a ciência de dados pode ajudar a resolver [de Neira et al. 2023].

Em Gupta and Badve 2017, os autores destacam a **necessidade de cooperação** entre diferentes entidades de rede para criar estratégias para melhorar a defesa contra ataques DDoS. Estudos como [Neira et al. 2023b] propõem a solução cooperativa para predição de ataques. As arquiteturas distribuídas, correlação de alertas, privacidade do usuário e alta precisão são desafios que soluções colaborativas devem abordar [de Neira et al. 2023]. Assim, é importante que soluções cooperativas usem a ciência de dados para lidar com os diferentes ciberataques, atingindo altas taxas de acerto.

Outro desafio é a necessidade de **soluções que se adaptem a diferentes cenários**. A distribuição estatística dos dados de entrada do sistema pode variar ao longo da execução do sistema. Assim, é essencial avaliar o comportamento das soluções diante de mudanças e criar mecanismos que evitem a degradação dos resultados caso a dinâmica dos dados mude. Para identificar mudanças de conceito, as soluções podem utilizar técnicas como *Adaptive Windowing*, *Concept Drift Detection* e *Early Drift Detection Method*. A literatura ilustra a utilização de técnicas de detecção de mudança de conceito em segurança cibernética, como a detecção de *botnets* [de Araújo et al. 2022]. Contudo, com a constante evolução dos ataques, esse tipo de solução será cada vez mais necessária.

A **explicabilidade dos resultados** é uma questão crítica em soluções baseadas na ciência de dados aplicada para a cibersegurança. A explicabilidade dos resultados previne perdas causadas por potenciais erros de previsão, pois essas soluções focam na transparência e interpretabilidade dos resultados [de Neira et al. 2023]. Assim, os administradores de rede podem interpretar como o modelo realizou a previsão e tomar as melhores decisões. Outro desafio significativo é a **coleta e análise de dados**. A coleta

de dados de rede é crítica para uma boa postura de segurança cibernética, mas a falta de padrões industriais para anotação e registro de dados, juntamente com a contaminação e heterogeneidade dos dados, dificulta o treinamento de detectores de anomalias com poucos exemplos rotulados [Hero et al. 2023]. Por fim, a implementação de mecanismos de **privacidade e desempenho** em ciência de dados aplicada às redes é desafiadora. Existe um trade-off entre a privacidade dos dados dos clientes e o desempenho dos algoritmos de aprendizado de máquina, especialmente em detecção de anomalias em larga escala [Hero et al. 2023]. A necessidade de equilibrar esses aspectos exige soluções inovadoras que protejam a privacidade sem comprometer a eficiência e a precisão das análises.

1.7. Considerações Finais

A crescente interconexão digital e a popularização dos dispositivos computacionais e tecnologias de comunicação evidenciam novas ameaças de segurança. A ampliação da superfície de ataque com a implantação da Internet das Coisas (IoT) expõe vulnerabilidades intrínsecas, facilitando a geração e exploração de ameaças. Nesse contexto, a ciência de dados e a inteligência artificial (IA) emergem como ferramentas poderosas na cibersegurança, oferecendo novas possibilidades para a análise de grandes volumes de dados, a identificação e predição de vulnerabilidades e a detecção de intrusões.

Este capítulo extrapola a visão limitada à IA aplicada à cibersegurança e vai além. Ele apresenta os conceitos, a metodologia e as técnicas da ciência de dados para cibersegurança. Os principais objetivos deste capítulo foram disseminar a cultura e os conceitos da ciência de dados na cibersegurança; demonstrar o potencial das técnicas de IA e AM para essa área; incentivar colaborações entre outros grupos de pesquisa no Brasil e demonstrar resultados alcançados no projeto MCTI/FAPESP MENTORED relacionados ao tema. Neste capítulo, foi enfatizada a importância da ciência de dados, da inteligência artificial e do aprendizado de máquina na proteção dos sistemas digitais, associando com trabalhos existentes na literatura. Ele apresenta de forma didática os principais conceitos, a metodologia e as técnicas de ciência de dados aplicadas à cibersegurança. O capítulo segue uma organização apoiada nas etapas da Ciência de Dados: coleta de dados, preparação dos dados, pré-processamento de características, visualização dos dados e análise dos dados, associando essas etapas com o estado da arte no tema.

É importante enfatizar que este capítulo complementa o minicurso de mesmo título, ministrado em conjunto com o Simpósio Brasileiro de Sistemas de Computadores e de Sistemas Computacionais (SBSeg) 2024. No minicurso, será demonstrado como a ciência de dados automatiza processos e aprimora a detecção e resposta a ameaças, passando pela análise do comportamento de usuários e sistemas para identificar atividades suspeitas ou anomalias indicativas de ataques e identificação de vulnerabilidades.

É importante enfatizar a importância e os benefícios do uso de técnicas de Ciência de Dados em Cibersegurança. Porém, são igualmente claras as limitações e os desafios no uso dessas técnicas. Um desses desafios se remete à qualidade dos dados de entrada usados, além do grande desbalanceamento dos dados quando se compara o número de instâncias em um cenário sem ataques e em um cenário de ataques. Outro ponto é o sobre o desafio em ajustar e calibrar os modelos utilizados. As dificuldades de generalização dos resultados alcançados com esses modelos e técnicas são conhecidas, assim

como os vários dilemas na aplicação de técnicas de aprendizado de máquinas, como o dilema de Occam, onde infelizmente a acurácia e a simplicidade (interpretabilidade) em ML conflitam. Além disso, a maleabilidade e a facilidade com que se leva a modelos a oferecerem resultados que se deseja (nem sempre os corretos) são grandes. Por exemplo, para quaisquer dois algoritmos de aprendizado de máquina, existem tantas situações (apropriadamente ponderadas) em que se pode demonstrar que um algoritmo é superior a outro e vice-versa, de acordo com qualquer das métricas aplicadas. Essa fluidez é um ponto de atenção na análise e credibilidade dos resultados divulgados na literatura. Por fim, o dilema denominado de maldição da dimensionalidade que pode comprometer e enviesar visões e resultados. Esses são aspectos importantes que precisam ser considerados na análise dos dados e explicabilidade das análises.

Agradecimentos

Os autores agradecem o apoio da UFPR e da UFMG e o auxílio financeiro da FAPESP, bolsas #2018/23098-0, #2022/06840-0 e #2024/04923-0 CNPq, bolsas #309129/2017-6 e #432204/2018-0, CAPES, bolsas #88887.501287/2020-00.

Referências

[Abaid et al. 2016] Abaid, Z., Sarkar, D., Kaafar, M. A., and Jha, S. (2016). The early bird gets the botnet: A Markov chain based early warning system for botnet attacks. In *LCN*, pages 61–68, UAE. IEEE.

[Alani 2021] Alani, M. M. (2021). Big data in cybersecurity: a survey of applications and future trends. *Journal of Reliable Intelligent Environments*, 7(2):85–114.

[Albano et al. 2023] Albano, L., Borges, L., Neira, A., and Nogueira, M. (2023). Predição de ataques DDoS pela correlação de séries temporais via padrões ordinais. In *Anais do XXIII SBSeg*, pages 69–82, Brasil. SBC.

[Ali and Al-Shaer 2013] Ali, M. Q. and Al-Shaer, E. (2013). Configuration-based IDS for advanced metering infrastructure. In *SIGSAC*, page 451–462, USA. ACM.

[AlMahmoud et al. 2019] AlMahmoud, A., Damiani, E., Otrok, H., and Al-Hammadi, Y. (2019). Spamdoop: A privacy-preserving big data platform for collaborative spam detection. *IEEE TBD*, 5(3):293–304.

[Alsharif et al. 2022] Alsharif, M., Mishra, S., and AlShehri, M. (2022). Impact of human vulnerabilities on cybersecurity. *Computer Systems Science & Engineering*, 40(3).

[Anderson et al. 1972] Anderson, J. P. et al. (1972). Computer security technology planning study. Technical report, Citeseer.

[Araujo et al. 2023] Araujo, A. M., Bergamini de Neira, A., and Nogueira, M. (2023). Autonomous machine learning for early bot detection in the Internet of things. *Digital Comm. and Net.*, 9(6):1301–1309.

[Azevedo 2016] Azevedo, P. R. M. d. (2016). *Introdução à estatística*. EDUFRN, RN, 3 edition.

[Bellare and Rogaway 1993] Bellare, M. and Rogaway, P. (1993). Random oracles are practical: A paradigm for designing efficient protocols. In *CCS*, pages 66–75. ACM.

[Bhatia et al. 2018] Bhatia, S., Behal, S., and Ahmed, I. (2018). *Distributed Denial of Service Attacks and Defense Mechanisms: Current Landscape and Future Directions*, pages 55–97. Springer, Cham.

[Borges et al. 2024] Borges, L., de Neira, A. B., Albano, L., and Nogueira, M. (2024). Multifaceted DDoS attack prediction by multivariate time series and ordinal patterns. In *2024 IEEE ICC (WS18)*, USA.

[Brito et al. 2023] Brito, D., de Neira, A. B., Borges, L. F., and Nogueira, M. (2023). An autonomous system for predicting DDoS attacks on local area networks and the Internet. In *2023 IEEE LATINCOM*, pages 1–6, Panama. IEEE.

[Bruijne et al. 2017] Bruijne, M. d., Eeten, M. v., Ganan, C. H., and Pieters, W. (2017). *Towards a new cyber threat actor typology*. TU Delft.

[Cohen 1960] Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

[Crevier 1993] Crevier, D. (1993). *AI: The Tumultuous History of the Search for Artificial Intelligence*. Basic Books, Inc., New York, NY, USA.

[de Araújo et al. 2022] de Araújo, A. M., de Neira, A. B., and Nogueira, M. (2022). Lifelong autonomous botnet detection. In *GLOBECOM*, pages 1–6, Brazil. IEEE.

[de Neira et al. 2020] de Neira, A. B., Araujo, A. M., and Nogueira, M. (2020). Early botnet detection for the Internet and the Internet of Things by autonomous machine learning. In *MSN*, pages 516–523, Japan.

[de Neira et al. 2023] de Neira, A. B., Kantarci, B., and Nogueira, M. (2023). Distributed denial of service attack prediction: Challenges, open issues and opportunities. *ComNet*, 222:109553.

[de Padrões e Tecnologia (NIST) 2024] de Padrões e Tecnologia (NIST), I. N. (2024). Nist cybersecurity framework (NIST CSF).

[De Paola et al. 2018] De Paola, A., Gaglio, S., Re, G. L., and Morana, M. (2018). A hybrid system for malware detection on big data. In *IEEE INFOCOM (WKSHPS)*, pages 45–50.

[Delgado and Tibau 2019] Delgado, R. and Tibau, X.-A. (2019). Why cohen's kappa should be avoided as performance measure in classification. *PLOS ONE*, 14(9):e0222916.

[Di Battista et al. 2004] Di Battista, G., Mariani, F., Patrignani, M., and Pizzonia, M. (2004). *BGPlay: A System for Visualizing the Interdomain Routing Evolution*, page 295–306. Springer Berlin Heidelberg.

[Do et al. 2019] Do, Q., Martini, B., and Choo, K.-K. R. (2019). The role of the adversary model in applied security research. *Computers & Security*, 81:156–181.

[Dolev and Yao 1983] Dolev, D. and Yao, A. C. (1983). On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208.

[Douligeris and Mitrokotsa 2004] Douligeris, C. and Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms: Classification and state-of-the-art. *Comput. Netw.*, 44(5):643–666.

[Dunsin et al. 2024] Dunsin, D., Ghanem, M. C., Ouazzane, K., and Vassilev, V. (2024). A comprehensive analysis of the role of artificial intelligence and machine learning in modern digital forensics and incident response. *FSI Digital Investigation*, 48:301675.

[Feurer et al. 2015] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J. T., Blum, M., and Hutter, F. (2015). Efficient and robust automated machine learning. In *NeurIPS*, NIPS'15, page 2755–2763, Cambridge, MA, USA. MIT Press.

[Feurer et al. 2019] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J. T., Blum, M., and Hutter, F. (2019). Auto-sklearn: efficient and robust automated machine learning. In *Automated Machine Learning*, page 21. Springer.

[Grimaldi et al. 2023] Grimaldi, A., Ribiollet, J., Nespoli, P., and Garcia-Alfaro, J. (2023). Toward next-generation cyber range: A comparative study of training platforms. In *ESORICS*, pages 271–290. Springer.

[Gupta and Badve 2017] Gupta, B. B. and Badve, O. P. (2017). Taxonomy of DoS and DDoS attacks and desirable defense mechanism in a cloud computing environment. *Neural. Comput. Appl.*, 28(12):3655–3682.

[Gupta and Dahiya 2021] Gupta, B. B. and Dahiya, A. (2021). *Distributed Denial of Service (DDoS) Attacks: Classification, Attacks, Challenges, and Countermeasures*. CRC Press, USA.

[Ham 2021] Ham, J. V. D. (2021). Toward a better understanding of "cybersecurity". *Digital Threats: Research and Practice*, 2(3):1–3.

[Hastie et al. 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *Unsupervised Learning*, pages 485–585. Springer New York, New York, NY.

[Hero et al. 2023] Hero, A., Kar, S., Moura, J., Neil, J., Poor, H. V., Turcotte, M., and Xi, B. (2023). Statistics and Data Science for Cybersecurity. *Harvard Data Science Review*, 5(1). https://hdsr.mitpress.mit.edu/pub/koyzu1te.

[Holgado et al. 2020] Holgado, P., Villagrá, V. A., and Vázquez, L. (2020). Real-time multistep attack prediction based on hidden markov models. *IEEE TDSC*, 17(1):134–147.

[Horsanali et al. 2021] Horsanali, E., Yigit, Y., Secinti, G., Karameseoglu, A., and Canberk, B. (2021). Network-aware AutoML framework for software-defined sensor networks. In *DCOSS*, pages 451–457. IEEE.

[Ibitoye et al. 2020] Ibitoye, O., Abou-Khamis, R., Matrawy, A., and Shafiq, M. O. (2020). The threat of adversarial attacks on machine learning in network security – a survey.

[Imran et al. 2021] Imran, Jamil, F., and Kim, D. (2021). An ensemble of prediction and learning mechanism for improving accuracy of anomaly detection in network intrusion environments. *Sustainability*, 13(18):10057.

[Janos 2020] Janos, M. (2020). Deep learning – conceitos e aplicações. Acessado em: 12/2021. https://www.3dimensoes.com.br/post/deep-learning-conceitos-e-aplica%C3%A7%C3%B5es.

[Jog et al. 2015] Jog, M., Natu, M., and Shelke, S. (2015). Distributed and predictive-preventive defense against DDoS attacks. In *ICDCN*, USA. ACM.

[Jurgens and Cin 2024] Jurgens, J. and Cin, P. D. (2024). Global cybersecurity outlook 2024. Online. Fórum Econômico Mundial.

[Kaluarachchi et al. 2021] Kaluarachchi, T., Reis, A., and Nanayakkara, S. (2021). A review of recent deep learning approaches in human-centered machine learning. *Sensors*, 21(7).

[Kawaguchi 2016] Kawaguchi, K. (2016). Deep learning without poor local minima. *Advances in Neural Information Processing Systems*, 29:586–594.

[Kour and Gondhi 2020] Kour, H. and Gondhi, N. (2020). Machine learning techniques: A survey. In *IDCTA*, pages 266–275, Cham. Springer International Publishing.

[Kumbhare and Chobe 2014] Kumbhare, T. A. and Chobe, S. V. (2014). An overview of association rule mining algorithms. *IJCSIT*, 5(1):927–930.

[Lam and Abbas 2020] Lam, J. and Abbas, R. (2020). Machine learning based anomaly detection for 5G networks.

[Landis and Koch 1977] Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159.

[Laprie et al. 2004] Laprie, J.-C., Randell, B., and Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE TDSC*, 1(1):11–33.

[Leros and Andreatos 2019] Leros, A. P. and Andreatos, A. S. (2019). *Network Traffic Analytics for Internet Service Providers—Application in Early Prediction of DDoS Attacks*, pages 233–267. Springer, Cham.

[Lipner and Anderson 2018] Lipner, S. and Anderson, R. (2018). CIA history. *Personal commun.*

[Liu and Lang 2019] Liu, H. and Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences*, 9(20).

[Liu et al. 2010] Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. (2010). Understanding of internal clustering validation measures. In *2010 IEEE ICDM*, pages 911–916.

[Liu et al. 2015] Liu, Y., Zhang, J., Sarabi, A., Liu, M., Karir, M., and Bailey, M. (2015). Predicting cyber security incidents using feature-based characterization of network-level malicious activities. In *IWSPA*, page 3–9, USA. ACM.

[Maldonado et al. 2022] Maldonado, J., Riff, M. C., and Neveu, B. (2022). A review of recent approaches on wrapper feature selection for intrusion detection. *ESWA*, 198:116822.

[Martin et al. 2018] Martin, G., Ghafur, S., Kinross, J., Hankin, C., and Darzi, A. (2018). Wannacry—a year on.

[Monroe 2021] Monroe, D. (2021). Trouble at the source. *CACM*, 64(12):17–19.

[Montagner and Westphall 2022] Montagner, A. S. and Westphall, C. M. (2022). Uma breve análise sobre phishing. *ComInG*, 6(1):46–56.

[Muhammad et al. 2020] Muhammad, A., Asad, M., and Javed, A. R. (2020). Robust early stage botnet detection using machine learning. In *ICCWS*, pages 1–6, Pakistan. IEEE.

[Muhammad and Yan 2015] Muhammad, I. and Yan, Z. (2015). Supervised machine learning approaches: A survey. *ICTACT Journal on Soft Computing*, 5(3).

[Nair 2024] Nair, S. S. (2024). Securing against advanced cyber threats: A comprehensive guide to phishing, xss, and sql injection defense. *JCSTS*, 6(1):76–93.

[Najafi Mohsenabad and Tut 2024] Najafi Mohsenabad, H. and Tut, M. A. (2024). Optimizing cybersecurity attack detection in computer networks: A comparative analysis of bio-inspired optimization algorithms using the CSE-CIC-IDS 2018 dataset. *Applied Sciences*, 14(3).

[Neira et al. 2023a] Neira, A., Borges, L., Araújo, A., and Nogueira, M. (2023a). Unsupervised feature engineering approach to predict DDoS attacks. In *IEEE Globecom*, Malaysia. IEEE.

[Neira et al. 2023b] Neira, A. B. d., Araujo, A. M. d., and Nogueira, M. (2023b). An intelligent system for DDoS attack prediction based on early warning signals. *TNSM*, 20(2):1254–1266.

[Ngo et al. 2020] Ngo, F. T., Agarwal, A., Govindu, R., and MacDonald, C. (2020). *Malicious Software Threats*, pages 793–813. Springer, Cham.

[Nilashi et al. 2020] Nilashi, M., Ahmadi, H., Manaf, A. A., Rashid, T. A., Samad, S., Shahmoradi, L., Aljojo, N., and Akbari, E. (2020). Coronary heart disease diagnosis through self-organizing map and fuzzy support vector machine with incremental updates. *IJFS*, 22(4).

[NIST 2018] NIST (2018). Framework for improving critical infrastructure cybersecurity. Technical report, U.S. Department of Commerce.

[Noel et al. 2016] Noel, S., Harley, E., Tam, K., Limiero, M., and Share, M. (2016). Chapter 4 - cygraph: Graph-based analytics and visualization for cybersecurity. In Gudivada, V. N., Raghavan, V. V., Govindaraju, V., and Rao, C., editors, *Cognitive Computing: Theory and Applications*, volume 35 of *Handbook of Statistics*, pages 117–167. Elsevier.

[Nogueira et al. 2021] Nogueira, M., Borges, L. F., and Nakayama, F. (2021). Das redes vestíveis aos sistemas ciber-humanos: Uma perspectiva na comunicação e privacidade dos dados. *SBRC, Sociedade Brasileira de Computação*.

[Noorbehbahani and Saberi 2020] Noorbehbahani, F. and Saberi, M. (2020). Ransomware detection with semi-supervised learning. In *2020 ICCKE*, pages 024–029, Irã. IEEE.

[Olabelurin et al. 2015] Olabelurin, A., Veluru, S., Healing, A., and Rajarajan, M. (2015). Entropy clustering approach for improving forecasting in DDoS attacks. In *ICNSC*, pages 315–320, Taiwan. IEEE.

[O'Reilly 2021] O'Reilly (2021). Chapter 1. introduction to tensorflow: Acessado em: 12/2021. https://www.oreilly.com/library/view/ai-and-machine/9781492078180/ch01.html#introduction_to_tensorflow.

[Papadopoulos et al. 2013] Papadopoulos, S., Theodoridis, G., and Tzovaras, D. (2013). Bgpfuse: using visual feature fusion for the detection and attribution of bgp anomalies. In *VizSec*, VizSec '13, page 57–64, New York, NY, USA. Association for Computing Machinery.

[Pelloso et al. 2018] Pelloso, M., Vergutz, A., Santos, A., and Nogueira, M. (2018). A self-adaptable system for DDoS attack prediction based on the metastability theory. In *GLOBECOM*, pages 1–6, UAE. IEEE.

[Pise and Kulkarni 2008] Pise, N. N. and Kulkarni, P. (2008). A survey of semi-supervised learning methods. In *2008 CIS*, volume 2, pages 30–34.

[Portella et al. 2015] Portella, A. C. F., do Nascimento, I. R., Alves, A. F., and Scheidt, G. N. (2015). *Estatística básica para os cursos de ciências exatas e tecnológicas*. EDUFT, Palmas, TO, 1. ed. edition.

[Prates Jr et al. 2021] Prates Jr, N. G., Andrade, A. M., de Mello, E. R., Wangham, M. S., and Nogueira, M. (2021). Um ambiente de experimentaçao em cibersegurança para Internet das coisas. In *Anais do VI Workshop do testbed FIBRE*, pages 68–79. SBC.

[Protić 2018] Protić, D. D. (2018). Review of kdd cup '99, nsl-kdd and kyoto 2006+ datasets. *Vojnotehnički glasnik/Military Technical Courier*, 66(3):580–596.

[Raschka 2020] Raschka, S. (2020). Chapter 1: Introduction to machine learning and deep learning. Acessado em: 12/2021. https://sebastianraschka.com/blog/2020/intro-to-dl-ch01.html.

[Raynor et al. 2023] Raynor, J., Crnovrsanin, T., Di Bartolomeo, S., South, L., Saffo, D., and Dunne, C. (2023). The state of the art in bgp visualization tools: A mapping of visualization techniques to cyberattack types. *IEEE TVCG*, 29(1):1059–1069.

[Ren et al. 2021] Ren, P., Xiao, Y., Chang, X., Huang, P.-y., Li, Z., Chen, X., and Wang, X. (2021). A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Computing Surveys*, 54(4).

[Salim et al. 2020] Salim, M. M., Rathore, S., and Park, J. H. (2020). Distributed denial of service attacks and its defenses in IoT: a survey. *The Journal of Supercomputing*, 76(7):5320–5363.

[Sapienza et al. 2018] Sapienza, A., Ernala, S. K., Bessi, A., Lerman, K., and Ferrara, E. (2018). DISCOVER: Mining online chatter for emerging cyber threats. In *WWW '18*, page 983–990, France. WWW.

[Serpeloni et al. 2024] Serpeloni, C. V. C., Malta, E. B. S., Alencar, J. O., and Lobo, R. L. (2024). Uma abordagem sobre a gestão e tratamento de eventos e incidentes utilizando o microsoft sentinel. *JTnI*, 4(2):22–22.

[Sewak 2019] Sewak, M. (2019). *Deep reinforcement learning*. Springer.

[Shin et al. 2011] Shin, S., Gu, G., Reddy, N., and Lee, C. P. (2011). A large-scale empirical study of conficker. *IEEE Transactions on Information Forensics and Security*, 7(2):676–690.

[Silva et al. 2015] Silva, J. L. d. C. e., Fernandes, M. W., and de Almeida, R. L. F. (2015). *Estatística e Probabilidade*. EdUECE, Fortaleza, CE, 3. ed. edition.

[Singh 2019] Singh, P. (2019). *Supervised Machine Learning*, pages 117–159. Apress, CA.

[Somani et al. 2017] Somani, G., Gaur, M. S., Sanghi, D., Conti, M., and Buyya, R. (2017). DDoS attacks in cloud computing: Issues, taxonomy, and future directions. *Comput. Commun.*, 107:30–48.

[Sutton and Barto 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[Tiwari and Dwivedi 2016] Tiwari, V. K. and Dwivedi, R. (2016). Analysis of cyber attack vectors. In *2016 ICCCA*, pages 600–604. IEEE.

[Vegesna 2023] Vegesna, V. V. (2023). Adopting a conceptual architecture to mitigate an iot zero-day threat that might result in a zero-day attack with regard to operational costs and communication overheads. *IJCESR*, 10:9–17.

[Wang et al. 2020] Wang, Z., Hong, T., and Piette, M. A. (2020). Building thermal load prediction through shallow machine learning and deep learning. *Applied Energy*, 263:114683.

[Wang and Zhang 2017] Wang, Z. and Zhang, Y. (2017). DDoS event forecasting using Twitter data. In *IJCAI*, page 4151–4157, Australia. AAAI Press.

[Wlosinski 2019] Wlosinski, L. G. (2019). Cybersecurity takedowns. *ISACA JOURNAL*, 6.

[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 Commun. Surv. Tutor.*, 15(4):2046–2069.

[Zhang et al. 2018] Zhang, W., Yang, G., Lin, Y., Ji, C., and Gupta, M. M. (2018). On definition of deep learning. In *2018 World Automation Congress (WAC)*, pages 1–5.

[Zhou and Belkin 2014] Zhou, X. and Belkin, M. (2014). Chapter 22 - semi-supervised learning. In *Academic Press Library in Signal Processing: Volume 1*, volume 1 of *Academic Press Library in Signal Processing*, pages 1239–1269. Elsevier.