Capítulo 4

Segurança em Redes 5G: Oportunidades e Desafios em Detecção de Anomalias e Predição de Tráfego baseadas em Aprendizado de Máquina

Guilherme N. N. Barbosa (UFF), Govinda Mohini G. Bezerra (UFF),
Dianne S. V. de Medeiros (UFF), Martin Andreoni Lopez (TII),
Diogo M. F. Mattos (UFF)

Abstract

This chapter focuses on approaching and contextualizing the security of the fifth-generation (5G) mobile networks, discussing network anomaly detection techniques through hybrid tools. Classical techniques for prediction, such as time series regression analysis and the Hidden Markov Model, are revisited. New anomaly detection and traffic prediction techniques based on deep learning are presented, such as recurrent neural networks, neural networks with long short-term memory, and convolutional neural networks. Finally, the challenges and new paradigms of the next-generation networks (6G) are presented. We also present a case study with a practical exercise to develop an example of anomaly detection and traffic prediction through open source and free tools.

Resumo

Este capítulo aborda e contextualiza a segurança das redes móveis de quinta geração (5G), discutindo técnicas de predição de tráfego e detecção de anomalias em redes através de ferramentas híbridas. Revisitam-se técnicas clássicas para predição de tráfego, como análise de regressão de séries temporais e Modelo Oculto de Markov. Novas técnicas de detecção de anomalia e predição de tráfego baseadas em aprendizado profundo são apresentadas, tais como redes neurais recorrentes, redes neurais com memória longa de curto prazo e redes neurais convolucionais. Por fim, são apresentados os desafios da próxima geração de rede móveis (6G), novos paradigmas e um estudo de caso com exercício prático de desenvolvimento de um exemplo de detecção de anomalia e predição de tráfego através de ferramentas livres de código aberto.

Este capítulo foi realizado com recursos do CNPq, CAPES, RNP, FAPERJ, FAPESP (2018/23062-5) e Prefeitura de Niterói/FEC/UFF (Edital PDPA 2020).

4.1. Introdução

A quinta geração (5G) de sistemas de comunicações moveis é mais do que uma nova geração de tecnologias, mas denota uma nova era em que a conectividade se tornará cada vez mais fluida e flexível. Em 2025, as redes 5G provavelmente cobrirão um terço da população mundial[2], interconectando pessoas, máquinas e dispositivos inteligentes, com mais de 41 bilhões de dispositivos interconectados [Wasicek, 2020]. As redes 5G se adaptam aos aplicativos e o seu desempenho se ajusta precisamente às necessidades do usuário, permitindo a execução de diferentes aplicações como carros autônomos, enxames de drones [Andreoni Lopez et al., 2021], cirurgias remotas, comunicação máquina-a-máquina (*Machine-to-Machine* - M2M), entre outros. Além disso, a rede 5G visa melhorar o desempenho atual das redes móveis, aprimorando a experiencia dos usuários com picos de vazão de 10 Gb/s e latências menores que 1 milissegundo [Wazid et al., 2020]. O maior desempenho e a flexibilidade das redes 5G são devido à implementação do paradigma das redes sem fio definidas por software (*Wireless Software Defined Networs* - WSDN). Algumas soluções implementam as WSDN como o SoftAir, CRAN e CONTENT. Com as WSDNs, é possível realizar fatiamento (*slicing*) das redes [Popovski et al., 2018, Cunha et al., 2019]. Assim, a rede 5G é definida como uma rede orientadas a serviços, que permitem a implantação de novas aplicações com suporte a diferentes requisitos de desempenho. Atualmente, as redes móveis de quinta geração já são uma realidade para 176 redes comerciais no mundo e são foco de investimento de mais de 461 operadoras em 137 países[3]. Para usar essas redes já em operação, são catalogados mais de 600 dispositivos comercialmente disponíveis, mostrando que a tecnologia 5G é uma realidade comercializada e em rápido crescimento. Contudo, a tecnologia 5G impõe desafios para a garantia de privacidade e segurança. Assim, soluções de privacidade e segurança devem ser implantadas em vários níveis, incluindo dispositivos, equipamentos de interface aérea, infraestrutura de rede de acesso de rádio na nuvem (*Cloud - Radio Access Network* - C-RAN), instalações de *backhaul* móveis, entre outros. Para garantir a adequação do nível correto de segurança e privacidade, o 3GPP define a Especificação #: 33.501 para os requisitos de segurança dos sistemas 5G[4]. A organização foca a segurança 5G em autenticação de assinatura, autorização do equipamento do usuário, autorização de acesso e serviço de rede, mas também inclui o usuário e a integridade dos dados de sinalização para garantir a uniformidade e a interoperabilidade entre os elementos da rede.

Os maiores desafios de segurança 5G surgem na camada de aplicação, devido à multiplicidade de aplicações suportadas e à flexibilidade da tecnologia para acomodar novos aplicações. Com largura de banda substancialmente maior e latência ultrabaixa, a rede 5G oferece suporte a muitas aplicações e serviços novos e aprimorados, como realidade virtual não vinculada e telepresença a qualquer hora e em qualquer lugar. Aplicações disruptivos ou tradicionais, como Voz sobre 5G (*Voice over 5G*), são susceptíveis a problemas de segurança comuns, como confidencialidade e privacidade de dados, e a alguns problemas completamente novos, como roubo de identidade virtual ou a extrapolação do consentimento do usuário através da realização de aprendizado sobre seus dados.

[2]Disponível em https://www.gsma.com/futurenetworks/ip_services/understanding-5g/5g-innovation/.

[3]Disponível em https://gsacom.com/paper/5g-market-update-executive-summary-august-2021/.

[4]Disponível em https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3169.

4.1.1. As Gerações de Redes Móveis

A primeira geração de comunicações móveis (1G), introduzida em 1979, tinha transmissões com picos de 2,4 Kb/s. A segunda geração (2G), já na década de 1990, permitia o envio de mensagens curtas através da tecnologia *General Packet Radio Service* (GPRS), com velocidades entre 50 Kb/s e 1 Mb/s. Tanto a primeira como a segunda geração careciam de segurança no projeto. A terceira geração (3G), introduzida em 1998, fornecia serviços como internet móvel, navegação web, descarga de imagens com taxas de navegação de até 2 Mb/s. Nas redes 3G, foi introduzida a característica de autenticação mútua, ou de duas vias, para evitar a conexão com estações bases falsas. As redes referenciadas como 3.5G consistem em uma evolução da terceira geração, com a inserção da tecnologia de pacotes de acesso de alta velocidade (*High Speed Packet Access* - HSPA+) com velocidades teóricas de até 22 Mb/s. Em relação às redes 3G, a tecnologia da rede 3.5G provia uma nova rede de acesso, pois o núcleo da rede 3G já contava com comunicação sobre IP. As redes de quarta geração (4G) fornecem maiores velocidades e segurança introduzindo novos serviços sobre IP, como telefonia e TV. Além disso, a inserção de novas tecnologias, como múltiplas entradas e múltiplas saídas (*Multiple-input and multiple-output* - MIMO) para aumentar a capacidade de transmissão, e a multiplexação por divisão de frequências ortogonais (*Orthogonal Frequency Division Multiplexing* - OFDM) para manter altas taxas de transferências, fazem as redes 4G atingirem velocidades de transmissão de dados de até 1 Gb/s. As redes 4G usam protocolos criptográficos avançados para autenticação do usuário e oferecem proteção contra ataques físicos, como adulteração física de estações base, que podem ser instaladas em instalações públicas ou do usuário. No entanto, todas as gerações mencionadas carecem de suporte à manutenção da conexão confiável em mobilidade. A infraestrutura das redes 4G adicionam a banda larga móvel aprimorada (*enhanced Mobile BroadBand* - eMBB), a qual suporta conexões estáveis com taxas de dados de pico altas, bem como taxas moderadas para usuários de celular servindo como entrada para as redes 5G. As principais categorias de casos de uso das redes 5G são a comunicação massiva de tipo de máquina (*massive Machine Type Communication* - mMTC) [Bockelmann et al., 2016], com aplicações tais como o monitoramento de ambientes com grande quantidade de dispositivos e baixas ta-

Tabela 4.1. Resumo das ameaças nas diferentes tecnologias de comunicações móveis da primeira a quarta geração. Adaptado de [Ahmad et al., 2019].

Geração	Mecanismos de Segurança	Desafios de Segurança
1G	Sem medidas explícitas de segurança e privacidade.	Bisbilhotamento, interceptação de chamadas e nenhum mecanismo de privacidade.
2G	Proteção baseada em autenticação, anonimato e criptografia.	Estação base falsa, segurança de link de rádio, autenticação unilateral e spamming.
3G	Adotou a segurança 2G, acesso seguro à rede, Autenticação e Acordo de Chave (AKA) e autenticação bidirecional.	Vulnerabilidades de segurança de tráfego IP, segurança de chaves de criptografia, segurança de roaming.
4G	Nova criptografia (EPS-AKA) e mecanismos de confiança, segurança de chaves de criptografia, segurança de acesso do 3GPP e proteção de integridade.	Maior segurança induzida por tráfego de IP, integridade de dados, segurança de Base Transceiver Stations (BTS) e interceptação de chaves de longo prazo.

xas de transmissão, e as comunicações ultraconfiáveis de baixa latência (*Ultra-Reliable Low-Latency Communications* - URLLCs) [Popovski et al., 2018] que suportam transmissões de baixa latência, como a direção de veículos autônomos que precisam de alta confiabilidade e reação imediata.

4.1.2. As Ameaças de Segurança às Redes 5G

As redes 5G foram projetadas para solucionar muitas das falhas que suas predecessoras tinham, tais como limitações na autenticação de dispositivos, falhas à privacidade do usuário e a vulnerabilidade da interface de rádio. Como muitos operadores estendem a infraestrutura das redes 4G, quase todas as ameaças e requisitos de segurança relacionados às gerações móveis pré-5G ainda são aplicáveis no 5G. Além disso, o 5G terá um novo conjunto de desafios de segurança devido principalmente aos seguintes fatores: maior número de usuários, heterogeneidade de dispositivos conectados, novos serviços de rede, questões de privacidade do usuário e suporte a dispositivos da Internet das Coisas (*Internet of Things* – IoT) e a aplicativos de missão crítica. O software de rede e a utilização de novas tecnologias como redes definidas por software (*Software Defined Networking* – SDN) [Andreoni Lopez et al., 2016], virtualização de funções de rede (*Network Function Virtualization* – NFV) [Andreoni Lopez et al., 2019], computação de borda móvel (*Mobile Edge Computing* - MEC) e fatiamento da rede (*Network Slicing*), apresentarão desafios adicionais a segurança e privacidade [Khan et al., 2019].

Alguns desafios de segurança identificados na literatura [Ahmad et al., 2017a, Ahmad et al., 2018, Wazid et al., 2020] são: o **tráfego de rede repentino** (*flash crowd*), em que existe um grande número de dispositivos de usuário final acessando a infraestrutura ao mesmo tempo; **segurança de interfaces de rádio**, na qual as chaves de criptografia de interface de rádio devem ser enviadas por canais não seguros; **integridade do plano do usuário**, relacionada à proteção de dados do usuário para evitar o vazamento de informações confidencias; **segurança obrigatória na rede**, segurança fim-a-fim de todos os serviços na rede; **segurança de roaming**, pois os parâmetros de segurança do usuário não são atualizados com roaming de uma rede de operadora para outra, levando a compromissos de segurança em redes visitantes; **ataques de negação de serviço (DoS) na infraestrutura**, controle de pacotes para evitar o descontinuidade dos serviços; **tempestades de sinalização**, relacionadas a sistemas de controle distribuído que requerem coordenação, por exemplo o protocolo de *Non-Access Stratum* - (NAS) das redes 3G e 4G ; **Ataques DoS em dispositivos do usuário final**, já que não há medidas de segurança para sistemas operacionais, aplicativos e configuração de dispositivo de usuário dados.

Nas redes 5G, em virtude da conexão à internet dos objetos, ameaças tradicionais também podem ser executadas [Wazid et al., 2020]. **Bisbilhotamento** é uma ameaça passiva que ocorre quando o atacante escuta as mensagens trocadas entre participantes da rede. Outro ataque passivo é a **análise de tráfego**, no qual o atacante intercepta e examina o tráfego da rede para determinar o seu comportamento. O **ataque de repetição** ou *replay attack* é uma forma de ataque em que uma transmissão é repetida de forma maliciosa por um atacante que a interceptou. Esse ataque é semelhante ao **ataque do homen-no-meio**, no qual depois de extrair as mensagens como no ataque de repetição, o atacante modifica as mensagens antes de enviá-las ao destinatário. O ataque do homen-no-meio normalmente é associado ao **ataque de falsificação de identidade**, na qual o atacante modifica

algumas características como endereços IP ou MAC para fingir ser outro membro da rede e assim evitar o princípio de não repúdio. O **ataque de negação de serviço** (*Denial of Service* - DoS) é ainda uma das piores ameaças das rede. No ataque de DoS, o adversário realiza algumas tarefas, como explorar vulnerabilidades de protocolos, para impedir que partes legítimas acessem os recursos da rede ou sistema. Além disso, existe uma variante distribuída (*Distributed Denial of Service* - DDoS), na qual múltiplos atacantes atuam simultaneamente. Esse ataque pode ser realizado em aplicações, como a inundação por HTTP, TCP SYN e UDP, com o objetivo de consumir a maior largura de banda possível, ou na infraestrutura, como os ataques de negação de serviço nos rádios definidos por software [Li et al., 2011]. **Ataques às bases de dados** também são uma ameaça às rede 5G, pois na arquitetura das redes 5G existem diferentes servidores, seja na nuvem, na névoa ou na borda da rede. Múltiplos ataques podem ser executados contra as bases de dados com o objetivo de o atacante extrair informações. Ataques dentro desse grupo são a injeção de SQL, *Cross-Site Scripting* (XSS) e *Cross-Site Request Forgery* (CSRF). **Ataques de malware** permanecem como uma ameaça às novas redes, pois o atacante executa um Software Malicioso (*Malicious Software – Malware*) em um sistema remoto para realizar atividades não autorizadas, como roubo, exclusão, atualização e criptografia de informações importantes. Alguns tipos de *malware* são os cavalos de troia, software espião, *ransomware*, *keylogger* e botnets. As botnets são normalmente utilizadas para o espalhamento do *malware*. Botnets populares que ainda estão em funcionamento são Mirai, Reaper, Echobot e Necurs.

Considerando especificamente as redes 5G, é possível identificar desafios em relação aos casos de uso [Zhang et al., 2019b]. Para os milhares de dispositivos *mMTC* de baixo custo, como sensores, é essencial o uso de algoritmos criptográficos leves e protocolos de gerenciamento de chaves de baixo consumo, de forma a otimizar o uso das baterias de baixa capacidade desse tipo de dispositivo. Os serviços *URLCC*, que requerem muito baixa latência, requerem protocolos rápidos e leves de autenticação forte, assim como algoritmos criptográficos de alta velocidade para atender os requerimentos de baixa latência e alta confiabilidade. Contudo, a restrição de processamento desses dispositivos limita o nível de segurança alcançado pelos protocolos usados.

Uma vez que a redes 5G utilizam tecnologias como a Virtualização de Função de Rede (*Network Function Virtualization* - NFV) e Redes Definidas por Software *Software Defined Network* - SDN, as redes 5G herdam ameaças dessas tecnologias como mostra a Figura 4.1.2. A funções de rede virtualizadas (*Virtual Network Functions* - VNF) são introduzidas no 5G para consolidar várias funções de rede em dispositivos de software, que são executados em uma variedade de hardware padrão da indústria. A dissociação do software e do hardware permite a redução de despesas de capital e operacionais, aumentando a escalabilidade e a resiliência do serviço de rede. Contudo, desafios de segurança do NFV podem vir da infraestrutura de implantação da virtualização (*Network Function Virtualization Infrastructure* - NFVI), do gerenciamento e orquestração do ambiente NFV (*Network Function Virtualization Orchestration* - NFVO) e das interfaces entre o as funções de rede virtualizadas [Lal et al., 2017]. Como ameaças na infraestrutura, o atacante pode criar máquinas virtuais que contêm *malware* para ganhar acesso a outras VMs ou mesmo ao *hypervisor*. Mesmo o *hypervisor* pode ser atacado através da injeção de código malicioso que pode comprometer o controle, assim como as VMs hóspedes. As funções

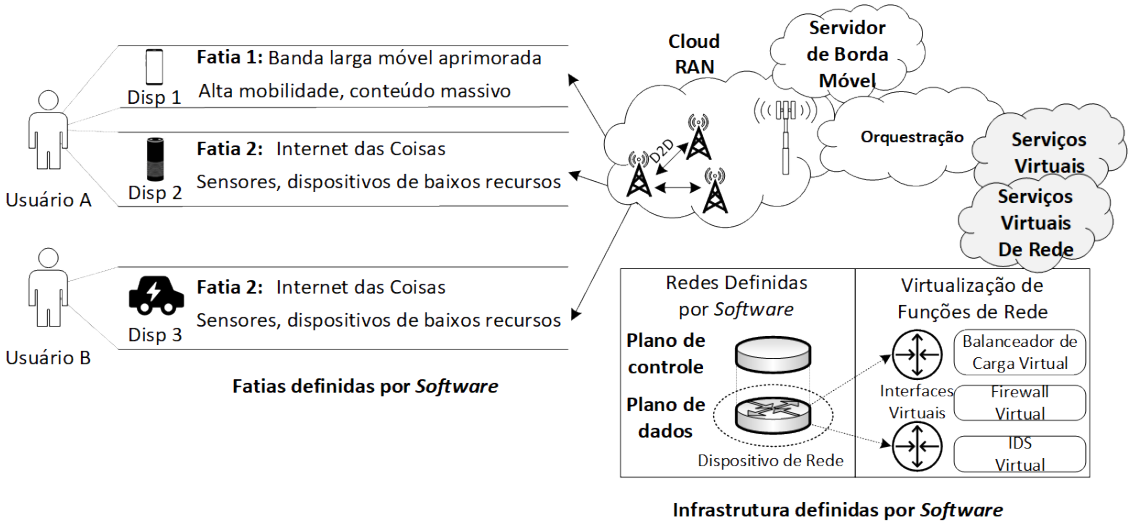

Figura 4.1. Exemplo de arquitetura da rede 5G. Cada usuário tem diferentes equipamentos (*User Equipment* – EU). Para cada UE é possível obter uma fatia de rede com os requerimentos e qualidade de serviço específico. As estações bases são controladas pela rede de acesso de rádio na nuvem (*Cloud Radio Access Network*. Entre os dispositivos pode existir comunicação direta sem passar pelas estações bases. O servidor de borda móvel trará a nuvem para mais perto da rede dos dispositivos. O gerenciamento é realizado por redes definidas por *software* que permitem a criação de serviços virtuais e funções de rede virtualizadas. Adaptado de [Nieto et al., 2019].

de rede virtualizadas (VNFs) são vulneráveis a todo tipo de ameaças de software. Além disso, o atacante pode executar um ataque de negação de serviço contra a VNF para inundar a comunicação. O gerenciamento e a orquestração (NFVO) são um alvo importante já que pode ser considerado como um ponto único de falha. Todas as regras de orquestração devem ser revisadas para ter uma consistência no sistema. As interfaces do NFV devem ser protegidas para evitar a execução de código malicioso, eliminação de portas dos fundos (*backdoors*), prevenção de vazamento de informação. O conceito do fatiamento da rede é introduzido nas redes 5G para fornecer serviços personalizados, permitindo o compartilhamento de recursos entre múltiplos inquilinos sobre uma mesma infraestrutura. No entanto, o isolamento é fundamental entre as fatias de rede [Cunha et al., 2019]. O atacante pode abusar da elasticidade de uma fatia para consumir os recursos de outra fatia.

As redes definidas por *software* (*Software Defined Network* - SDN) simplificam o gerenciamento das redes fornecendo programabilidade por meio do desacoplamento das funções de controle do plano de encaminhamento de dados. O plano de controle é logicamente centralizado para criar políticas de encaminhamento de dados e o plano de dados é distribuído para lidar com o tráfego com base nas políticas de encaminhamento. A centralização lógica das SDN apresenta múltiplas vulnerabilidades [Yao et al., 2019]. As interfaces entre os planos, conhecidas como *northbound*, entre o controlador e as aplicações de rede, e *southbound*, entre os elementos comutadores e o controlador, podem ser utilizadas para atacar outro plano. O plano de controle é especialmente atraente para ataques como negação de serviço devido à sua característica de centralidade como ponto único de falhas. Além disso, outras possíveis ameaças ressaltadas na literatura [Scott-Hayward et al., 2015] são: *acesso não autorizado*, seja ao controlador da

redes ou às aplicações; *vazamento da informação*, como a descoberta de regras de fluxos ou políticas de encaminhamento, credenciais como chaves ou certificados para cada rede lógica; *modificação de dados*, alterar regras de fluxos para modificar pacotes, ataque homem-no-meio; *aplicações maliciosas*, execução de aplicações que permitem a inserção de regras fraudulentas; *negação de serviço*, seja através da inundação de comunicação do controlador-comutadores ou inundação da tabela de fluxo em cada comutador; *segurança do sistema SDN*, os comutadores OpenFlow podem operar no modo a prova de falhas ou falha autônoma, quando o comutador é desconectado do controlador, o atacante pode usar esses modos para atacar o controlador.

Paralelamente, a comunicação dispositivo-a-dispositivo *Device-to-Device*(D2D) apresenta desafios de privacidade em relação a localização. A comunicação D2D exige uma proximidade relativa entre os nós. Isso permite que usuários em conluio executem técnicas para localizar nós móveis próximos [De Ree et al., 2019]. A privacidade do local pode ser garantida usando técnicas de preservação de identidade de autenticação mútua anônima. A introdução de pequenas células móveis define os dispositivos móveis e a infraestrutura de rede. O principal problema das pequenas células móveis reside na falta de uma entidade segura e confiável para estabelecer a segurança durante a implantação da rede. Essa falta de uma entidade confiável apresenta problemas quando se trata de gerenciamento de chaves. Os esquemas de gerenciamento de chaves determinam como as chaves criptográficas são geradas, distribuídas aos nós da rede, autenticadas, atualizadas e revogadas.

4.1.3. Organização do Capítulo

Este capítulo foca nos desafios de segurança relacionados ao grande volume de dados que as redes 5G propiciam. O capítulo aborda técnicas para a previsão de tráfego e detecção de anomalias nas redes. São elencadas técnicas baseadas em inferência estatísticas e técnicas baseadas em aprendizado de máquina. Essas técnicas são essenciais para garantir a segurança das redes 5G, pois com o grande aumento previsto para as comunicações nessa nova geração de rede, o controle e a orquestração das infraestruturas de rede deverão ser mais ágeis e precisos.

O restante do capítulo está organizado da seguinte forma. A Seção 4.2 discute a privacidade dos dados pessoais nas redes 5G. As ferramentas para a predição de tráfego baseadas em modelos estatísticos são apresentadas na Seção 4.3. Por sua vez, as ferramentas baseadas em algoritmos de aprendizado de máquinas são elencadas na Seção 4.4. Os desafios da detecção de anomalias e predição de tráfego para as redes 5G são abordados na Seção 4.5. A Seção 4.6 discorre sobre os desafios futuros para a próxima geração de redes móveis, as redes 6G. Um exemplo prático para detecção de anomalias utilizando redes neurais é mostrado na Seção 4.7. As considerações finais e perspectivas estão na Seção 4.8.

4.2. A Privacidade dos Dados nas Redes 5G

Na era das redes 5G, o desempenho das aplicações está intimamente ligado com a exploração das capacidades dessas redes. O uso ótimo dos recursos disponíveis é alcançado garantindo os requisitos estritos de qualidade de serviço (*Quality of Service* –

QoS), como altas taxas de transmissão, latência ultra baixa e mínima variação de atraso *jitter*. Para tanto é necessária a criação de perfis verticais acurados em termos de uso de recursos, eficiência elástica e capacidade de se adaptar dinamicamente às condições da rede. O uso dos perfis é fundamental para automatizar os processos de produção e desenvolvimento de *software* de automação sobre uma infraestrutura 5G [Zafeiropoulos et al., 2020]. Novas tecnologias precisam ser testadas e combinadas com validações práticas. Zafeiropoulos *et al.* propõem uma metodologia integrada de *benchmarking* e de criação de perfis para aplicações industriais em 5G de forma a facilitar a extração de informações relevantes sobre o sistema testado para, assim, realizar o dimensionamento adequado das aplicações e determinar políticas de operação eficientes [Zafeiropoulos et al., 2020].

As redes 5G possuem grandes dimensões e incluem diversas partes interessadas, como os usuários finais, operadoras, provedores de serviços verticais, empresas e novas tecnologias em conjunto com novos modelos de negócios. Os serviços oferecidos nessas redes contêm informações primárias sobre seus usuários, como identidade, localização ou posição, e outros dados privados. É comum o uso de computação em nuvem por uma parte dos interessados para armazenar, usar e processar informações pessoais dos usuários finais. Os dados pessoais dos usuários finais são processados e compartilhados por diferentes partes interessadas de acordo com os objetivos de cada parte. Como essas informações são armazenadas e podem estar disponíveis às partes interessadas, as redes 5G evocam problemas significativos no vazamento de dados privados, podendo ser uma fonte crítica de violações de privacidade [Khan et al., 2019]. Atender às questões de privacidade para cada parte interessada é uma tarefa complexa devido à natureza paradoxal da tarefa nesse cenário, uma vez que as partes possuem interesses particulares contrastantes envolvidos. A análise do perfil de tráfego de aplicações que executam sobre as redes 5G pode conter diversas informações pessoais sobre seus usuários. A proposta FLOWR (*Flow Recognition*) é um sistema de autoaprendizado que requer um treinamento supervisionado mínimo e detecta automaticamente novas assinaturas de aplicativos contidos no fluxo de rede [Xu et al., 2015]. Para tanto, a proposta foca em aplicações *Web* e define como características do aplicativo a concatenação do nome do serviço *Web* e uma chave-valor do cabeçalho HTTP. Como assinatura do aplicativo, extrai características que identificam o aplicativo com uma boa margem de confiança. A proposta tem como premissa que fluxos em intervalos de tempo próximos e com uma alta probabilidade de ocorrer concomitantemente são oriundos de um mesmo aplicativo. Assim, uma característica que ocorre em um fluxo com um intervalo de tempo T, junto à assinatura de um aplicativo, e que possui uma probabilidade de ocorrência concomitante com a assinatura maior que um limiar p, é promovida a assinatura daquele aplicativo. Para o funcionamento do sistema, é necessário um conhecimento inicial que é extraído de um conjunto de dados.

Soluções de Internet das Coisas (*Internet of Things* - IoT) são uma das principais aplicações das redes 5G. Contudo, os operadores desses ambientes inteligentes não têm o completo conhecimento dos seu inventário de dispositivos IoT. Sivanathan *et al.* focam em identificar dispositivos IoT em uma rede, através da assinatura de cada dispositivo e desenvolvem um arcabouço para classificação de dispositivos IoT usando características de tráfego obtidas a nível de rede [Sivanathan et al., 2019]. Para tanto, são utilizados 28 dispositivos IoT, englobando câmeras, luzes, tomadas, sensores de movimento, eletrodomésticos e monitores de saúde. Os traços (*traces*) de tráfego de todos os dispositivos são

coletados por um período de seis meses. A análise utiliza características estatísticas, aplicando as seguintes métricas para a caracterização dos dispositivos: volume, duração e taxa média do fluxo, tempo de hibernação, número das portas, endereços de consultas DNS, intervalo das consultas NTP (*Network Time Protocol*) e conjuntos de cifras do *handshaking* TLS (*Transport Layer Secutrity*). É realizada uma medida de custo de obtenção de cada uma das variáveis, de acordo com necessidade de processamento dessas medidas. O custo é, então, classificado em custo baixo, médio ou alto. O trabalho propõe também uma métrica de mérito dos atributos, ou seja, o impacto de cada variável no resultado da classificação. Dessa forma, é possível realizar uma escolha de quais variáveis utilizar, de forma a otimizar a implementação em linha (*online*), sem comprometer o desempenho da classificação. A proposta IoTArgos implementa um sistema de monitoramento de segurança multi-camadas, que coleta, analisa e caracteriza dados de comunicação de dispositivos IoT heterogêneos através de roteadores domésticos programáveis [Wan et al., 2020]. O sistema IoTArgos executa em 22 redes domésticas de três países diferentes, constituídas de 20 dispositivos IoT com diversas aplicações. Esses dispositivos coletam dados continuamente por seis meses utilizando roteadores domésticos programáveis que executam o *software* OpenWrt, e usando *dongles* USB para coleta de pacotes ZigBee e Bluetooth. Ao total, são coletados 6 milhões de fluxos considerados normais. Em relação aos dados de ataques, são simulados 19 diferentes ataques a dispositivos IoT em diferentes camadas, gerando 300 mil fluxos atacantes. O sistema extrai dois tipos de características multi-camadas. O primeiro contém informações consideradas com características brutas: endereço IP, nome do domínio do terminal de destino, tempo de chegada entre pacotes, tamanho do pacote, duração do fluxo, portas utilizadas e portas. O segundo é considerado como características avançadas: quantidade de terminais remotos e quantidade de aplicações dominantes. A classificação dos ataques e a detecção de intrusão são realizadas utilizando um modelo de aprendizado de máquina com múltiplos estágios. O primeiro estágio consiste na utilização de algoritmos clássicos de aprendizado de máquinas supervisionado, como K-Vizinhos Mais Próximos (*K-Nearest Neighbors* - KNN), Regressão Logística, Naïve Bayes, Floresta Aleatória (*Random Forest* - RF) e Máquina de Vetor de Suporte (*Support Vector Machine* - SVM), para classificar ataques conhecidos. Os fluxos considerados normais pelo primeiro estágio, ou seja, fluxos legítimos, são inseridos no segundo estágio que utiliza algoritmos de aprendizado não supervisionado para descobrir comportamentos suspeitos ou não usuais, sendo capaz de evitar, assim, ataques desconhecidos e de dia zero (*zero-day attacks*). Os autores propõem o conceito de um módulo de defesa em tempo real, que consiste em um módulo seguinte ao de detecção de intrusão com o propósito de alertar os usuários sobre o ataque detectado, além de desabilitar ou desconectar os dispositivos comprometidos e o seu respectivo concentrador (*hub*), caso necessário. A avaliação experimental demostra que o sistema IoTArgos é capaz de detectar atividades anômalas que visam dispositivos IoT em casas inteligentes com alta precisão.

Li *et al.* demonstram que o aumento da abrangência das câmeras e a integração na vida cotidiana podem resultar em padrões de comportamento e problemas de privacidade [Li et al., 2020]. Os autores realizaram um estudo detalhado, utilizando um grande provedor de câmeras de segurança domésticas (*Home Security Camera* - HSC), cobrindo 15,4 milhões de fluxos e 211 mil usuários. As análises são realizadas através de duas

abordagens: comportamento por usuário e comprometimento de privacidade. Os serviços oferecidos pelos provedores de HSC possuem basicamente dois modos: o modo ao vivo, em que o usuário assiste as imagens captadas pelas câmeras em tempo real, e o modo de reprodução, no qual são realizadas gravações das imagens no servidor a partir de uma detecção de movimento e então os usuários podem assistir as gravações posteriormente. O modo ao vivo está disponível a todos os usuários gratuitamente enquanto o modo de reprodução somente usuários que pagam uma assinatura. Do conjunto total de usuários, 59% dos usuários pagam pela assinatura e correspondem a 95% do total de tráfego, cuja predominância está no tráfego de *upload* de reprodução. Segundo a análise, 60% desse tráfego não é assistido, o que caracteriza um desperdício de recursos de rede e de armazenamento. É ressaltado que os usuários tendem a assistir os vídeos das câmeras em uma ou duas localidades que, normalmente, são diferentes da localização da câmera. Foram identificados três riscos à privacidade dos usuários. O primeiro é o risco de pico de tráfego devido a um aumento vertiginoso no tráfego da câmera que indica que o usuário começou a assistir ao vídeo ao vivo ou houve alguma detecção de movimento que iniciou a gravação das imagens. O trabalho mostra que a aplicação de um classificador simples é capaz de distinguir entre esses dois estados com 100% de acurácia, o que leva ao conhecimento da presença ou não dos usuários nas casas. O segundo risco está relacionado à regularidade de tráfego. O padrão de tráfego das câmeras pode representar o padrão de comportamento dos usuários, revelando as suas rotinas. Os usuários mais suscetíveis a este tipo de ataque são os usuários que pagam pela assinatura e têm altas taxas de *upload*. O terceiro risco está relacionado à mudança da taxa de tráfego, pois indica mudanças nas atividades realizadas pelos usuários. Experimentos com diversas atividades identificaram as diferenças de taxas de tráfego de acordo com as mudanças das atividades.

4.3. As Ferramentas baseadas em Modelos Estatísticos

Nas redes móveis de quinta geração (5G), um dos maiores desafios será gerenciamento de rede devido à sua complexidade. Assim, ITU (*International Telecommunication*) trabalha na atualização das recomendações relativas à qualidade de serviço (QoS) e qualidade de experiência dos usuários (QoE). A recomendação ITU-T Y.3172 prevê a introdução de mecanismos de aprendizado de máquina para o gerenciamento e a orquestração funcionalidades nas próximas gerações de rede[5]. Nesse sentido, modelos estatísticos baseados em séries temporais são os métodos clássicos com bom desempenho sempre escolhidos para realizar predições [Boukerche et al., 2020], sobretudo em fluxos de rede, por possuírem uma boa capacidade analítica e uma implementação com baixo custo computacional. A previsão de séries é um campo essencial do aprendizado de máquina aplicado a redes 5G [Chakraborty et al., 2020]. A modelagem de séries temporais é uma ampla área de pesquisa e vários modelos de previsões de séries temporais evoluíram ao longo do tempo. Nesta seção, são abordados os principais modelos estatísticos para análise de séries temporais utilizando técnicas de regressão como ARIMA (*Auto-Regressive Integrated Moving Average*) e SARIMA (*Seasonal Auto-Regressive Integrated Moving Average*), e o Modelo Oculto de Markov (*Hidden Markov Model* - HMM) que têm como característica a análise de probabilidades entre eventos. As séries temporais são representações matemáticas de fenômenos que ocorrem continuamente durante um intervalo

[5]Disponível em https://www.itu.int/rec/T-REC-Y.3172-201906-I/en.

de tempo. Podem ser divididas em três componentes: tendência, sazonalidade e irregularidade. A tendência relaciona-se a uma perspectiva de longo prazo, a sazonalidade diz respeito a eventos sistemáticos associados ao calendário e, por último, as irregularidades são flutuações não sistemáticas em curta duração [Medeiros et al., 2019]. Existem objetivos basilares para realizar a análise de uma série, sendo eles a cognição do mecanismo gerador da série e a predição de pontos futuros. No que diz respeito ao mecanismo gerador da série, é fundamental identificar o comportamento, isto é, descrever se existem ciclos, tendências ou sazonalidade e pontos de periodicidade relevantes. Com base nisso, a predição do comportamento é possível. Cabe ressaltar que a escolha do melhor método e seus respectivos parâmetros para uma dada série, tem como objetivo reduzir os erros de predição, pois estimar o futuro envolve incertezas.

4.3.1. *Auto-Regressive Integrated Moving Average* – ARIMA

O modelo ARIMA foi inicialmente proposto por George Box e Gwilym Jenkins, sendo também conhecido como método Box-Jenkins. Existem ainda variações do modelo como o VARIMA(*Vector Auto-Regressive Integrated Moving Average*), que é utilizado para múltiplas séries temporais, e o SARIMA (*Seasonal Auto-Regressive Integrated Moving Average*), empregado em casos em que existe uma possível sazonalidade nos pontos da série. Todos esses modelos possuem um ótimo desempenho para análises de curto prazo, enquanto o modelo SARIMA é o que possui melhor capacidade para análises a longo prazo. A estrutura do ARIMA é composta por três coeficientes sendo o primeiro denominado auto-regressivo p, seguido do coeficiente de diferenciação d e por último o coeficiente de médias móveis da série q. O modelo ARIMA [Yang et al., 2021] é dado por:

$$y'_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i y'_{t-i} + \varepsilon_t + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i}, \tag{1}$$

em que o coeficiente α_i refere-se ao termo auto-regressivo da série, β_i é relacionado à média móvel e ε_t diz respeito à parte residual do modelo. As principais etapas para utilização do modelo ARIMA podem ser realizadas em três etapas [Yang et al., 2021]:

1. **Pré-processamento na série**. O pré-processamento pode ser feito através do teste *Augmented Dickey–Fuller* (ADF) para identificar se a série é estacionária. Em caso negativo, são realizadas diferenciações da série, quantas vezes forem necessárias, até obter uma série estacionária. O número de diferenciações é caracterizado através do parâmetro d;

2. **Cálculo dos valores da função de autocorrelação amostral (ACF) e autocorrelação parcial (PACF)**. O cálculo dos valores das funções ACF e PACF é feito para a série estacionária obtida, determinando os parâmetros p e q respectivamente. Para fins de desempenho, esses parâmetros podem ser obtidos através da análise da métrica *Akaike Information Criterion* (AIC), que tem como objetivo mensurar a qualidade relativa de um modelo estatístico;

3. **Teste do modelo e realizar predições**. Por fim, são realizados testes no modelo que apresenta o melhor desempenho e as predições da série são realizadas.

4.3.2. *Seasonal Auto-Regressive Integrated Moving Average* – SARIMA

Uma das principais variações do ARIMA é o modelo SARIMA. Esse modelo tem como objetivo realizar uma análise mais profunda em séries com características predominantes de sazonalidade e periodicidade, podendo ser útil para predições de tráfego de redes sem fio [Sone et al., 2020] e detecção de anomalias em redes [Kromkowski et al., 2019]. Por ser uma variação do modelo ARIMA, o SARIMA pode ser representado por SARIMA$(p,d,q)(P,D,Q)_s$. A primeira parte do modelo, representada pelos parâmetros p, d e q é não sazonal, enquanto a segunda parte é sazonal e constitui o fator de sazonalidade. Os parâmetros P, D e Q representam respectivamente o número dos termos de sazonalidade da parte auto-regressiva, o número de diferenciações sazonais e a parte sazonal de médias móveis. O fator de sazonalidade contribui para analisar características como uso de banda, que tendem a ter comportamento cíclico. Hanbanchong e Piromsopa utilizam o modelo SARIMA para detectar anomalias predizendo o uso de banda através da sazonalidade existente [Hanbanchong e Piromsopa, 2012]. Em diversos outros campos de estudo, em que a série temporal é utilizada como fonte de análise para estabelecer pontos futuros, o modelo SARIMA é amplamente utilizado. Análise de condições climáticas, predição de carga energética e propagação de doenças infecciosas são temas de estudo que frequentemente usam esse modelo.

4.3.3. Modelo Oculto de Markov (*Hidden Markov Model* – HMM)

Processos estocásticos são definidos através de variáveis aleatórias, as quais representam características determinísticas em um intervalo de tempo t. Os processos estocásticos são utilizados para analisar o comportamento de sistemas em que o grau de incerteza é consideravelmente alto. Esses processos podem ser classificados em dois cenários, em relação ao estado e ao tempo, com característica discretas e contínuas para cada um dos cenários. Um processo estocástico é considerado *markoviano* se a probabilidade condicional de um estado futuro depende apenas do estado presente e não dos estados anteriores. Essa probabilidade pode ser descrita como probabilidade de transição e é expressa matematicamente por

$$P(q_{t+i} = S_j | q_t = S_i). \tag{2}$$

A equação 2 representa a probabilidade do estado q_{t+1} ser S_j no momento $t+1$ dado que o estado q_t é igual a S_i no instante t.

Um processo markoviano é classificado como **Cadeia de Markov** se as variáveis aleatórias são definidas em um espaço de estados discreto. A Cadeia de Markov representa sistemas que podem a qualquer instante de tempo t estar em um dado estado S. A mudança entre um estado e outro ocorre através de uma matriz de transição, que descreve as probabilidades de o sistema mudar do estado S_0 para o estado S_n. A Figura 4.3.3 mostra uma Cadeia de Markov com 3 estados e as probabilidades de transição entre esses estados, representadas por $\alpha_{i,j}$.

O conjunto de probabilidades da matriz de transição de uma Cadeia de Markov é caracterizado pela Equação 3 e deve obedecer às propriedades das Equações 4 e 5.

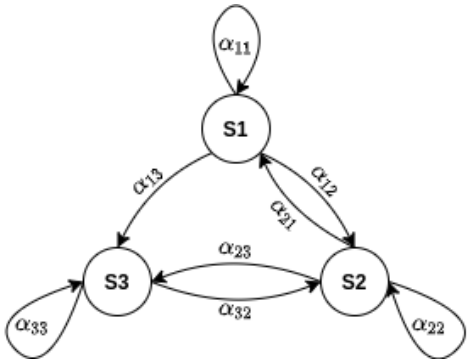

Figura 4.2. Exemplo de representação de uma cadeia de Markov com probabilidades de transição entre três estados. Os estados são representados pelas variáveis s_i e as probabilidades de transição entre estados, pelas variáveis α_{ij}.

$$a_{ij} = P(s_{t+i} = S_j | s_t = S_i) \qquad (3)$$

$$a_{ij} \geq 0 \qquad (4) \qquad \sum_{i=1}^{N} a_{ij} = 1 \qquad (5)$$

O **Modelo Oculto de Markov** (*Hidden Markov Model* - HMM), em sua essência, é a variação de um processo estocástico Markoviano. O HMM é caracterizado por duas componentes, uma não observável e outra observável. A primeira componente representa o estado de um sistema previamente modelado, enquanto a segunda representa as observações já realizadas. Os processos não observáveis representam um conjunto de estados interligados através da matriz de probabilidades, enquanto os processos observáveis representam as saídas de cada estado. Como exemplo, alertas de um sistema de detecção de intrusão (*Intrusion Detection System* - IDS) [Chadza et al., 2020], podem ser caracterizados como um processo estocástico observável em um modelo oculto de Markov, no qual a sequência de observações representam os alertas e a sequência de estados ocultos representam o estado do evento de segurança [Zhan et al., 2020]. O modelo é representado de forma reduzida através de uma tupla com três elementos, (A, B, π), em que A representa a matriz de transição, B a distribuição de probabilidades das observações e π o vetor de probabilidade inicial. Di Bernardino e Brogi mostram que o modelo também pode ser representado com parâmetros adicionais [Di Bernardino e Brogi, 2019], da seguinte forma:

1. Sendo N o número de estados do sistema, o conjunto de estados descritos individualmente é dado por
$$S = \{S_1, S_2, ..., S_N\}; \qquad (6)$$
2. Existe um número M de observações realizadas, cujo conjunto é dado por
$$O = \{O_1, O_2, ..., O_M\}; \qquad (7)$$
3. A transição entre estados é dada pela matriz de transição de probabilidades A que possui dimensão NxN e é definida por $A = [a_{ij}]$, cujos elementos são dados por
$$a_{ij} = P(s_{t+i} = S_j | s_t = S_i), \qquad 1 \leq i, \quad j \leq N; \qquad (8)$$

4. A matriz de probabilidades de observações $B = [b_{ij}]$ possui dimensão NxM e os elementos são descritos através de

$$b_{ij} = P(o_t = O_j | s_t = Si), \qquad 1 \leq i \leq N, \quad 1 \leq j \leq M; \tag{9}$$

5. O vetor de probabilidade inicial é definido por

$$\pi_i = P(s_1 = S_i), \qquad 1 \leq i \leq N. \tag{10}$$

No HMM existem dois tipos principais de estruturas, classificadas como ergóticas, ou sem restrições, e esquerda-direita (*left-right*). No modelo com estrutura ergótica, cada estado pode transitar entre quaisquer outros, sendo esse modelo completamente conectado. O modelo com estrutura esquerda-direita não permite transições entre um estado e estados anteriores [Chadza et al., 2020], sendo mais relevante para detecção de ataques, principalmente os que possuem diversas etapas antes de atingirem o objetivo.

4.3.4. Classificador Bayesiano

A classificação Bayesiana é fundamentada no Teorema de Bayes, no qual as probabilidades de um dado evento estão condicionadas à probabilidade de hipótese com resultados já conhecidos. Seja um conjunto de dados $X = (x_1, y_1)$,...,(X_N, y_n) com x amostras e y classes correlatas para um problema de classificação, sendo $x \in \mathbb{R}$ e $y \in [1, K]$ [Medeiros et al., 2019], o Teorema de Bayes é descrito por:

$$P(y = i|x) = \frac{P(i) * P(x|i)}{P(x)}, \tag{11}$$

em que $p(i)$ a probabilidade de uma hipótese ser verdadeira a partir da amostra de uma classe e $p(y|x)$ a distribuição de probabilidades desconhecidas no espaço amostral x.

O classificador Naïve Bayes tem sua origem no **Teorema de Bayes** e tem como premissa desconsiderar a correlação entre variáveis. É comumente utilizado para dados com alta dimensionalidade [Kumar Dwivedi et al., 2018]. A probabilidade condicional é utilizada para prever ataques e tráfegos regulares, podendo ser utilizada na detecção de ataques em redes definidas por software [Ahmad et al., 2020], por exemplo. Em redes *Ad-Hoc*, o classificador Naïve Bayes pode ser usado como parte de um arcabouço para detecção de ataques de Negação de Serviço Distribuído (*Distributed Denial-of-Service* - DDoS) [Reddy e Thilagam, 2020]. Nesse caso, Reddy e Thilagam utilizam o classificador para dividir o tráfego de rede em dois padrões, normal e ataques DDoS. Para isso, consideram cinco características do tráfego para determinar a qual padrão um fluxo pertence, sendo eles o tamanho do pacote, a porta, o IP de origem, o IP de destino e a variação do atraso (*jitter*). O classificador é utilizado amplamente para classificação de textos, sendo possível detectar cargas úteis anômalas no tráfego de rede [Swarnkar e Hubballi, 2016]. Essa detecção é importante porque uma das principais formas de ataque HTTP ocorre através da modificação da carga útil do pacote.

A tabela 4.2 descreve os modelos estatísticos descritos nesta seção, apresentando as principais características, finalidades, pontos positivos e negativos.

Tabela 4.2. Comparativo entre os principais modelos estatísticos usados para a predição de tráfego em redes 5G.

Modelo	Finalidade	Aplicação	Prós	Contras
ARIMA	Análise de séries temporais	Predição de Tráfego	Análise de curto prazo	Custo computacional
SARIMA	Análise de séries temporais com sazonalidade	Predição de Tráfego	Captura dependência entre dados consecutivos	Custo computacional
HMM	Predição de estados	Detecção de anomalias	Capacidade de capturar dependência temporal	Tempo de treinamento
Classificador Bayesiano	Classificador	Detecção de anomalias	Eficaz para múltiplas classes	Presume independência das características

4.4. As Ferramentas baseadas em Algoritmos de Aprendizado de Máquinas

A rede 5G demanda a implantação de uma infraestrutura mais automatizada e independente da ação humana. Técnicas de aprendizado de máquina têm se tornado amplamente utilizadas em predições de tráfego e detecção de anomalias, principalmente pelo alto desempenho, que compensa o custo computacional exigido por alguns algoritmos. As técnicas de aprendizado recebem dados com o objetivo de obter modelos que descrevam as observações realizadas, permitindo a descoberta de comportamentos até então então desconhecidos. A partir dos modelos, podem ser tomadas decisões sobre tarefas específicas de forma acurada [Zhu et al., 2019]. O núcleo da rede 5G deve ser escalável e para isso implementam-se novas instâncias sob demanda utilizando redes definidas por software (*Software-Defined Networking* - SDN) e virtualização de funções de rede (*Network Function Virtualization* - NFV). Para que isso seja realizado de forma automatizada e utilizando os recursos de rede de forma eficiente, é necessário prever a carga de uso da rede a todo instante. Dessa forma, torna-se possível provisionar corretamente os recursos. Nesse sentido, a utilização de técnicas de aprendizado de máquina é fundamental [Alawe et al., 2018]. Por outro lado, com o aumento de tráfego e da quantidade de dispositivos, a segurança também exige esforços para detectar anomalias e ameaças antecipadamente na rede 5G, principalmente em ambientes com utilização de canal compartilhado e computação na borda, uma vez que brechas de segurança são decorrentes da comunicação com redes abertas, facilitadas pela virtualização de funções de rede [Sedjelmaci, 2021]. A detecção de ataques provenientes de *botnets*, por exemplo, pode ser realizada através de aprendizado de máquina utilizando um sistema de detecção de intrusão (*Intrusion Detection System* - IDS) baseado em redes neurais profundas (*Deep Neural Network* - DNN) [Fernandez Maimo et al., 2018, Lobato et al., 2021].

O aprendizado de máquina pode ser classificado em diversas categorias, sendo as principais as listadas a seguir [Medeiros et al., 2019]:

- **Aprendizado supervisionado**, em que as observações são fornecidas como pares de entrada-saída e o objetivo do algoritmo é identificar uma função que relacione as entradas com as saídas. O algoritmo realiza predições baseadas em amostras com padrões previamente rotulados. O treinamento é mantido até que se encontre um modelo ótimo de precisão e acurácia. Como exemplo, podem-se citar os modelos de máquina de vetor de suporte (*Support Vector Machine* - SVM), Redes Neurais, árvores de decisão, entre outros;
- **Aprendizado não supervisionado**, em que as observações fornecidas para os algoritmos são referentes somente aos dados de entrada e sem rotulação, sendo o objetivo do algoritmo agrupar as entradas em grupos similares denominados agrupamentos (*clusters*). K-Médias (*K-Means*), Floresta de Isolamento (*Isolation Forest*) e Rede de Crença Profunda (*Deep Belief Network*) são exemplos de algoritmos de aprendizado não supervisionado;
- **Aprendizado semi-supervisionado** caracteriza-se como uma interseção entre os modelos supervisionado e não-supervisionado. Os algoritmos são capazes de aprender a partir de um conjunto de dados parcialmente rotulado e generalizam o aprendizado para os demais dados, não rotulados;
- **Aprendizado por reforço**, em que os algoritmos se baseiam em um modelo de recompensas e punições que são oferecidas a partir da interação do modelo com o ambiente. Não há mapeamento direto entre entradas e saídas e os resultados são obtidos a partir da retroalimentação (*feedback loop*) entre o sistema de aprendizado e o ambiente. A cada iteração, as ações disponíveis são apresentadas ao modelo no seu estado atual e, após a mudança de estado, recebe um sinal de reforço que tem o objetivo de instigar um comportamento desejado, ou seja, ações que maximizam a recompensa a longo prazo [Kaelbling et al., 1996]. Exemplos de algoritmos dessa categoria são *Q-Learning*, *Q-Learning* Profundo (*Deep Q-Learning* - DQL) e Estado-Ação-Recompensa-Estado-Ação (*State–Action–Reward–State–Action* - SARSA).

Nas redes 5G, o aumento de tráfego é consequência do número de dispositivos conectados, aumentando a complexidade das redes e gerando uma quantidade significativa de informações para análise em tempo real. Tal efeito induz uma maior dificuldade no processamento e detecção de anomalias. Nesse contexto, o aprendizado profundo (*deep learning*) torna-se uma ferramenta valiosa. O aprendizado profundo é uma área do aprendizado de máquina que tem o objetivo de reconhecer padrões complexos em estrutura de dados a partir de de representações mais simples dessas estruturas. As soluções mais simples são organizadas em uma hierarquia cujos diferentes níveis se complementam para a composição de informações complexas [Goodfellow et al., 2016]. Trinh *et al.*, por exemplo, utilizam um modelo semi-supervisionado com uso do algoritmo memória longa de curto prazo (*Long Short-Term Memory* - LSTM) para detectar atividades legítimas [Trinh et al., 2019]. A detecção é importante sobretudo em áreas metropolitanas, onde podem ocorrer anomalias causadas por aglomerações inesperadas e degradação no serviço de maneira involuntária.

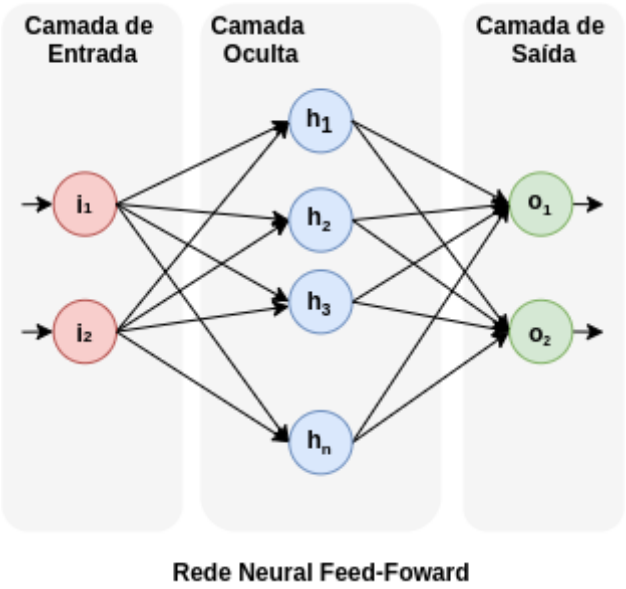

Figura 4.3. Camadas de uma Rede Neural *Feed-Forward* e interação entre os neurônios. A rede neural é formada por três camadas, entrada, oculta e saída. O fluxo de informação percorre a rede da entrada para a saída. Não há retroalimentação entre neurônios.

4.4.1. Redes Neurais Recorrentes (*Recurrent Neural Network* – RNN)

As Redes Neurais *Feed-Forward*, também conhecidas apenas como redes neurais, fazem parte dos primeiros algoritmos que impulsionaram a inteligência artificial. Simulando o comportamento do cérebro humano, tais redes foram utilizadas primordialmente para problemas de classificação e, com o avanço do poder computacional, tornaram-se capazes de trabalhar em diversos campos da ciência. As Redes Neurais *Feed-Forward* possuem a característica de processamento de dados em sequência. A estrutura dessas redes é composta por três camadas, pelas quais o fluxo de informações transita de maneira unidirecional. A informação se move da camada de entrada para a camada oculta e termina nas camadas de saída, conforme ilustrado na Figura 4.4.1. A camada de entrada recebe as informações, representadas na figura por i_1 e i_2, repassando-as para a camada oculta, que aplica uma função matemática específica nos dados da camada anterior para produzir uma saída. Essa função é conhecida como função de transferência, representada na figura por h_k. Por fim, a camada de saída representa o resultado do treinamento da rede neural, representado na figura pelas saídas o_1 e o_2. Por considerarem apenas a entrada atual, essas redes não possuem memória, não sendo viáveis para realizar predições de séries temporais.

As redes neurais recorrentes (*Recurrent Neural Networks* - RNNs) são uma variação de rede neural *feed-forward* que adiciona a capacidade de memorizar os estados passados para processar as próximas sequências de dados, possuindo grande potencial para realizar predições em séries temporais [Jiang e Schotten, 2019]. A Figura 4.4.1 mostra um exemplo genérico de RNN. Observa-se que, diferentemente das redes neurais *feed-forward*, há um relaxamento no sentido de fluxo da informação, que passa a poder fluir através de conexões cíclicas, representadas na figura pelas setas tracejadas. Essas

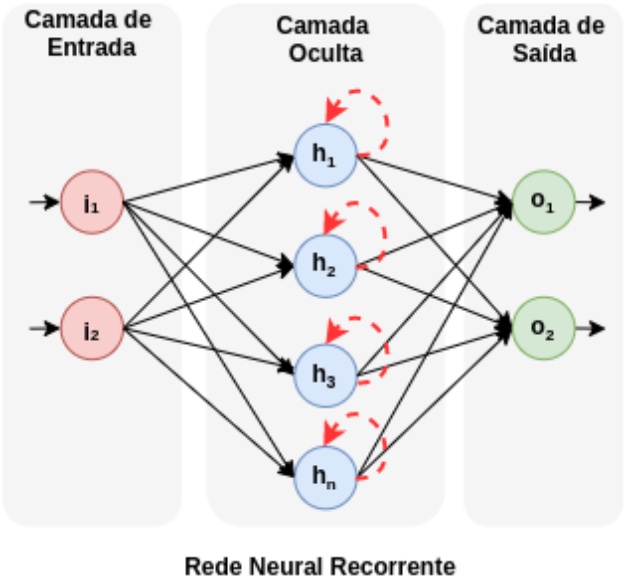

Figura 4.4. Interação entre os neurônios de uma Rede Neural Recorrente. A rede apresenta a capacidade memorizar estados passados e usá-los no processamento dos próximos dados. Há a retroalimentação de informação nos neurônios da camada oculta.

conexões permitem o acesso a estados anteriores, agregando a capacidade de memória à rede. Diversos estudos utilizam as redes neurais recorrentes para predição de tráfego, em virtude de capturar comportamentos mais complexos e não lineares, comparadas aos modelos estatísticos tradicionais, podendo exibir dependências de longo prazo [Ramakrishnan e Soni, 2018]. Ramakrishnan e Soni comparam alguns modelos de redes neurais recorrentes, como o modelo de memória longa de curto prazo (*Long Short-Term Memory* - LSTM) e unidades recorrentes fechada (*Gated Recurrent Units* - GRU) com modelos estatísticos tradicionais, para mensurar o desempenho de cada um na predição do volume de tráfego, de pacotes por protocolo e distribuição de pacotes. A análise mostra que a LSTM possui o melhor desempenho dentre os modelos. As redes móveis 5G produzem dados sequenciais em larga escala [Zhang et al., 2019a], tais como fluxos de tráfego de dados e latência de aplicativos. Dessa forma, é interessante utilizar a RNN para aprimorar a análise de dados de séries temporais em redes móveis.

4.4.2. *Long Short-Term Memory* – LSTM

A rede neural LSTM é uma variação de RNN, porém os nós da rede possuem um estado interno de memória, que pode ser utilizado para armazenar e recuperar informações durante várias iterações. Esse modelo vem sendo amplamente utilizado para modelagem de dados contínuos como processamento de linguagem e predição de séries temporais através de reconhecimento de padrões. Uma célula básica do modelo LSTM é apresentada na Figura 4.5. A estrutura da célula é composta por três portas lógicas denominadas Portão de Esquecimento (*Forget Gate*, Portão de Entrada (*Input Gate*) e Portão de Saída (*Output Gate*). O ***Forget Gate*** é responsável por remover os valores que não são mais elementares no estado da célula. Possui como entrada, dois valores sendo o primeiro h_{t-1} que diz respeito ao valor da célula anterior e x_t, que representa uma entrada em um

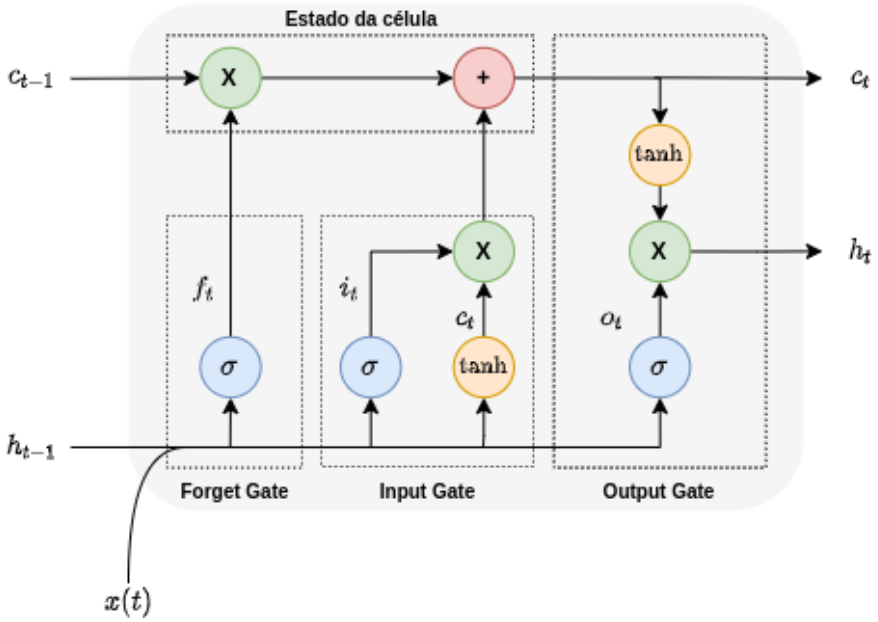

Figura 4.5. Célula básica de uma rede neural de memória longa de curto prazo (LSTM). Os portões de entrada (*input gate*), esquecimento (*forget gate*) e saída (*output gate*) controlam o esquecimento ou o aproveitamento de estados anteriores. A memória interna e novos dados são ativados por funções *sigmoides* e de tangente hiperbólica.

dado instante. Ambos os valores são inseridos em uma função de ativação denominada **sigmoide**, fornecendo uma saída binária. Em seguida, o valor é multiplicado por uma matriz de peso (W_f), e adiciona-se o enviesamento (*bias*) b_f. O enviesamento é um fator de correção para ajustar o modelo. O modelo é representado pela Equação 12.

$$f^{(t)} = \sigma(W_f[h^{(t-1)}, x^{(t)}] + b_f). \tag{12}$$

O ***input gate*** tem como objetivo inserir novas informações relevantes para atualizar o estado da célula. Inicialmente, o estado x_t e o estado oculto h_{t-1} são passados pela função *sigmoide*, tendo como valores de saída 0 e 1, sendo que 0 representa uma informação não relevante, enquanto 1 possui relevância para a memória. Em seguida, x_t e h_{t-1} passam por uma segunda função, a tangente hiperbólica $tanh$, que irá criar um vetor fornecendo valores -1 a 1. Em seguida, o resultado de cada uma das funções é inserido em uma função de multiplicação. O ***output gate*** tem como finalidade definir qual será o próximo estado oculto h_t, responsável por realizar as predições. Inicialmente, os valores h_{t-1} e x_t são passados por uma terceira função *sigmoide* resultando em um valor o_t. Então, o estado da célula é passado por uma função $tanh$. Por fim, são multiplicados ambos os valores para se obter o novo estado oculto h_t, determinando assim os próximos valores com relevância que devem ser transmitidos para a próxima etapa.

Dada a sua capacidade de memória, as redes neurais LSTM são utilizadas para predizer séries temporais. Soluções híbridas com outras redes neurais são utilizadas para otimizar o processamento e predição de tráfego de redes. Huang *et al.* utilizam redes

neurais convolucionais e LSTM para extrair, respectivamente, características geográficas e temporais do tráfego de rede [Huang et al., 2017]. Em redes sem fio, o conceito de Informação do Estado do Canal (*Channel State Information* - CSI) é fundamental para garantir qualidade nas transmissões, em função de características do meio, como interferências e perdas no espaço livre. Como a rede 5G propicia um aumento de dispositivos sem fio, torna-se fundamental realizar predições de tráfego voltadas para a otimização do uso dos canais. Luo *et al.* utilizam redes neurais convolucionais e LSTM para extrair as relações espaciais e temporais para predizer o CSI [Luo et al., 2020].

4.4.3. Redes Neurais Convolucionais

Dentro do contexto de inteligência artificial e aprendizado de máquina, as redes neurais convolucionais (*Convolutional Neural Networks* - CNNs) são um tipo de rede neural profunda (*Deep Neural Network* - DNN) utilizadas de forma mais eficiente com dados de entrada com características multidimensionais, por exemplo, imagens. As CNNs podem ser utilizadas para classificar ou agrupar os dados de saída, de acordo com um grau de similaridade atribuído pelo algoritmo. A influência para criação da rede neural convolucional é originária da estrutura do córtex visual do cérebro humano, que tem como objetivo processar informações visuais. O termo visão computacional sintetiza essas características biológica através de processos e modelagens utilizando, sobretudo, algoritmos capazes de analisar imagens e classificá-las, semelhante ao cérebro humano. A estrutura clássica de uma CNN é composta por cinco camadas, conforme mostra a Figura 4.4.3. As principais camadas são as camadas convolucionais, camadas de agrupamento e camadas densas [Andreoni Lopez e Mattos, 2021]. A **camada convolucional** contém filtros de tamanhos específicos, responsáveis por realizar a operação de convolução dos dados originados na camada de entrada, sendo imagens ou mapa de características, resultando em um novo mapa de características que irá alimentar a próxima camada. Matematicamente, uma imagem ou mapa de características é representado por uma matriz, tendo como componentes n_A, n_L e n_C representando altura, largura e número de canais respectivamente. Para o caso de uma imagem RGB, considera-se $n_C = 3$. Por convenção, considera-se que o filtro K é quadrado com dimensão ímpar, denominado por f, permitindo que cada pixel da imagem seja centralizado no filtro e, assim, considere todos os elementos em sua vizinhança. O produto convolucional entre a imagem e o filtro é uma matriz bidimensional, resultado de uma operação de multiplicação elementar entre o filtro e uma parte da imagem, conforme mostra a Figura 4.4.3 e expressa como:

$$conv(I,K)_{x,y} = \sum_{i=1}^{n_A} \sum_{j=1}^{n_L} \sum_{k=1}^{n_C} K_{i,j,k} I_{x+i-1,y+j-1,k}, \tag{13}$$

em que K é a matriz representando o filtro aplicado à matriz de entrada I para a operação de convolução *conv*.

A **camada de agrupamento** é utilizada após a camada de convolução, com o objetivo de reduzir as amostras das características extraídas da camada de entrada, sem impacto no número de canais, reduzindo dessa forma a redundância de dados [Andreoni Lopez e Mattos, 2021]. Por fim, a **camada densa** tem como finalidade descrever de forma mais detalhada as características extraídas da camada anterior. Uma função de ativação é utilizada para resultar na probabilidade de cada amostra na camada de saída.

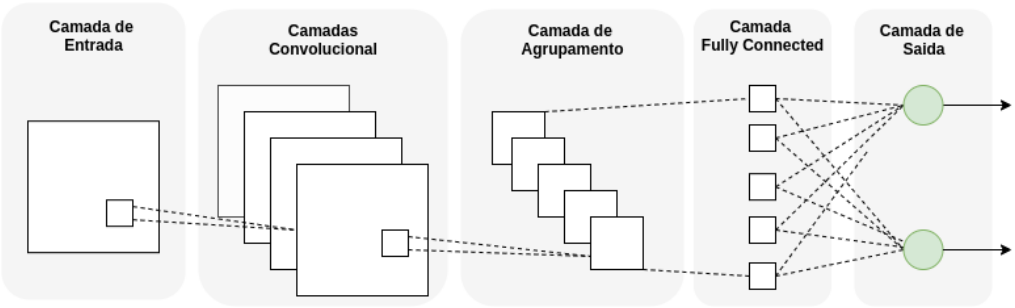

Figura 4.6. Estrutura simplificada de uma rede neural convolucional (*Convolutional Neural Network* - CNN). O aprendizado profundo com CNN é caracterizado pela repetição de camadas convolucional e de agrupamento. A cada par de camadas de convolução e agrupamento são extraídas características de mais alto nível. A camada densa (*Fully Connected*) realiza a classificação através das características extraídas. A consolidação do resultado ocorre camada de saída.

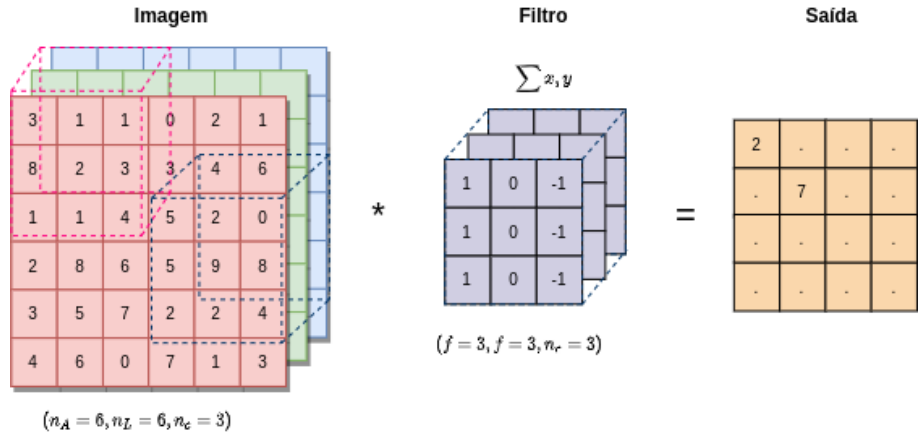

Figura 4.7. Operação convolucional entre imagem e filtro. O resultado da convolução é proveniente da multiplicação matricial entre segmentos da imagem de entrada e os filtros usados.

4.4.4. Codificadores Automáticos (*Autoencoders*)

Os codificadores automáticos, *autoencoders*, são um tipo de algoritmo de aprendizado de máquina não supervisionado utilizado para identificar a codificação dos dados de entrada. Na maioria dos casos, é utilizado como um pré-processamento de outras redes neurais com objetivo de reduzir a dimensionalidade dos dados e consequentemente ignorar ruídos existentes no sinal, podendo aprimorar a entrada de dados de algoritmos supervisionados, como a CNN por exemplo. A estrutura do codificador automático é composta por três camadas: uma camada de entrada, uma camada oculta e uma cada de saída, sendo a camada oculta utilizada como **codificador** e a camada de saída como **decodificador** [Bochie et al., 2020]. O codificador é representado por uma função $f(x)$ que transforma os dados de entrada em uma função h. O decodificador é uma função $g(x)$ que transforma a representação h para um valor reconstruído $\bar{x}$. Os codificadores automáticos são prioritariamente ferramentas para a comprimir dados. Atualmente, duas aplicações

práticas comuns dos codificadores automáticos são a eliminação de ruído de dados, já que levam os dados a uma versão mais compacta, codificada com perdas, e a redução de dimensionalidade para visualização de dados. Com as restrições de dimensionalidade e esparsidade apropriadas, os codificadores automáticos podem aprender projeções de dados que são mais interessantes do que a análise de componentes principais (*Principal Component Analysis* – PCA) ou outras técnicas simples.

Wu, Nekovee e Wang propõem um método de inferência da interferência dinâmica em um canal gaussiano multiusuário baseado em aprendizado profundo e codificadores automáticos [Wu et al., 2020]. A proposta é um mecanismo de codificador automático adaptativo. A intensidade da interferência é prevista por meio de um processo de aprendizado profundo, com a aprendizagem em linha (*online*) em tempo real do conhecimento do nível de interferência. Os resultados mostram que o codificador automático proposto funciona de forma mais robusta em um canal de interferência para todos os níveis de interferências. A melhoria é mais notável para os cenários de interferência forte e muito forte. A proposta estabelece uma base para permitir uma constelação adaptável para sistemas de comunicação 5G, nos quais condições de rede heterogêneas são consideradas.

4.4.5. Aprendizado Federado

O aprendizado federado (*Federated Learning* - FL) é um paradigma de aprendizado que tem como objetivo permitir que dispositivos móveis treinem de maneira colaborativa modelos preditivos compartilhados, mantendo os dados de treinamento localmente [Cunha Neto et al., 2020]. Isso garante principalmente segurança com relação aos dados, pois não é necessário o envio de informações sensíveis, como dados pessoais, para um servidor centralizado, sendo uma das principais técnicas que garantem a privacidade dos usuários. Com o aumento de dispositivos conectados na borda da rede e o aumento significativo de poder computacional por parte dos dispositivos de borda, o aprendizado federado é uma técnica promissora para utilização na rede 5G. O objetivo desse modelo é que cada dispositivo receba o modelo atual de um servidor central e, em seguida, utilize os próprios dados locais para realizar o treinamento local. Como cada cliente realiza um treinamento com dados distintos, são geradas pequenas atualizações locais que são enviadas para o servidor central. Por sua vez, o servidor central garante a agregação das atualizações originárias de todos os clientes, sendo calculada a média entre todos os participantes para melhorar o modelo compartilhado, através do algoritmo de média federada (*Federated Average* - FedAvg) [Cunha Neto et al., 2020]. Após a geração no modelo mais atual, novamente o servidor central envia para os clientes a última atualização. Assim, o aprendizado federado permite a geração de modelos mais eficientes com menor latência, pois é possível a utilização imediata do modelo no próprio dispositivo. O aprendizado federado é fortemente baseado em mecanismos de aprendizado de máquina treinados e otimizados através do método do gradiente descendente estocástico (*Stochastic Gradient Descent* - SGD). As principais implementações do aprendizado federado atuais dependem da ponderação da contribuição local dos diferentes clientes para definir a direção de otimização do modelo global através do algoritmo FedAvg.

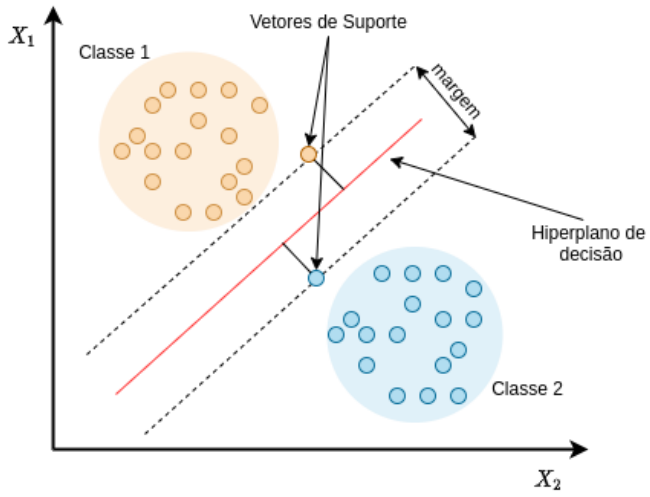

Figura 4.8. Estrutura geral do algoritmo máquina de vetor de suporte (*Support Vector Machine* - SVM). O algoritmo busca amostras de dados, vetores de suporte, que definem o hiperplano de separação entre classes. A seleção dos vetores de suporte visa maximizar a margem de separação entre classes.

4.4.6. Máquina de Vetor de Suporte de Classe Única (*One-Class Support Vector Machine* - OCSVM)

A Máquina de Vetor de Suporte (*Support Vector Machine* – SVM) é uma técnica de aprendizado de máquina supervisionada, podendo ser aplicada para classificação binária ou de múltiplas classes. Nesse último caso, aplica-se uma SVM para cada classe. A técnica é baseada na teoria de aprendizagem estatística e seu principal objetivo é classificar um conjunto de dados através de um espaço multidimensional, definindo um hiperplano de separação entre as classes de tal modo que os dados com as mesmas características estejam agrupados do mesmo lado do hiperplano. A proposta do algoritmo é encontrar os pontos mais próximos das linhas de cada uma das classes, conforme mostra a Figura 4.4.6, sendo esses pontos denominados **vetores de suporte**. Em seguida, calcula-se a distância da **margem**. O SVM tem como objetivo maximizar essa distância, pois assim será encontrado o hiperplano ideal do modelo, de forma a criar um limite de decisão entre as classes. É fundamental que a margem tenha a maior amplitude possível para garantir a maximização da distância entre as classes. O SVM utiliza uma família de funções denominada ***kernel***, que possui diversos tipos, como linear, polinomial, *sigmoide*, dentre outras. O objetivo da função *kernel* é transformar o conjunto de dados, de modo que uma superfície de decisão não linear possa ser transformada em um plano de dimensão superior, fazendo com que a separação entre as classes seja tratada de linearmente.

Uma variação do modelo tradicional é o modelo de classe única da máquina de vetor de suporte (*One-Class Support Vector Machine* - OCSVM), utilizado amplamente para detecção de anomalias. Para aplicação do modelo OCSVM utilizam-se no treinamento dados de uma única classe, ditos normais, ou dados contendo uma pequena fração de amostras anômalas. Com isso, o OCSVM é capaz de detectar amostras fora da classe alvo e amostras com novas características. As amostras que não pertencem à classe alvo estão distantes do hiperplano de decisão e são classificadas como pontos discrepantes

(*outliers*), que nesse caso representam as anomalias. Diversas aplicações utilizam esse modelo, por exemplo, o OCSVM é utilizado para aprender o comportamento normal dos sensores de veículos conectados e automatizados [Wang et al., 2021]. Para detectar variações de anomalias, os autores realizam um processamento nos dados originários dos sensores para mitigar a influência de ruídos utilizando um filtro de Kalman estendido.

A Tabela 4.3 apresenta de forma sintetizada as principais características de cada modelo apresentado nessa seção. Destaca-se que para cada modelo é necessário realizar uma ponderação entre custo computacional e desempenho.

Tabela 4.3. Comparativo entre os principais modelos de aprendizado de máquinas usados para a detecção de anomalias e predição de tráfego em redes 5G.

Modelo	Finalidade	Aplicação	Vantagens	Desvantagens
Codificadores automáticos (*Autoencoders*)	Aprendizado de representações e compactação	Detecção de anomalias	Capacidade em tratar dados sequenciais	Alto custo para grande volume de dados
CNN	Modelagem de dados espaciais	Detecção de anomalias e análise de dados espaciais	Precisão no reconhecimento de padrões	Alto custo computacional e dificuldade na definição de parâmetros
LSTM	Análise de dados sequenciais	Predição de tráfego	Modelagem de dependências de longo prazo	Maior consumo de memória
One-Class SVM	Classificador	Detecção de anomalias	Eficaz em espaços de alta dimensionalidade	Tempo de treinamento superior para grande conjunto de dados
RNN	Análise de dados sequenciais	Predição de tráfego	Capacidade de capturar dependência temporal	Necessidade de um grande conjunto de dados para treinamento

4.5. Os Principais Desafios de Gerenciamento, Segurança e Previsão de Tráfego em Redes 5G

O aumento na comunicação através da rede 5G representa uma disruptura na segurança e privacidade de dados transmitidos, principalmente no que diz respeito à criptografia. Métodos para garantir a segurança da informação, como esteganografia e criptografia caótica são modelos apontados como adequados para aplicações em tempo real [Kakkar, 2020], que serão utilizadas amplamente com a chegada da rede 5G. Wang *et. al.* propõem técnicas de aprendizado profundo para modelar a relação espaço-tempo na predição de redes móveis [Wang et al., 2017]. Para isso, utilizam uma arquitetura baseada em codificadores automáticos (*autoenconder*) e LSTM para avaliar a dependência espaço-tempo, em função da distribuição do tráfego na rede. Os autores utilizam o codificador automático para modelagem e extração das características espaciais e LSTM para modelagem

temporal. Alawe *et al.* investigam a escalabilidade dos recursos do núcleo da rede 5G, composto principalmente por SDN e NFV. O plano de controle no núcleo da rede 5G implementa a Função de Acessibilidade e Mobilidade (*Access and Mobility Function* - AMF) que atua diretamente nas requisições de conexão dos usuários. Assim, a AMF representa um gargalo no plano de controle de redes móveis [Alawe et al., 2018]. São comparados dois modelos de redes neurais, DNN e LSTM, tendo o último um desempenho superior. Nie *et al.* realizam predição de tráfego em redes de malha utilizando Rede de Crença Profunda (*Deep Belief Network* - DBN) [Nie et al., 2017]. Inicialmente, os autores utilizam a transformada discreta *wavelet* para extração das componentes de baixa frequência do tráfego de rede, que representam as dependências de longo prazo, aplicando em seguida a DBN para realizar a predição dessa componente. Já as componentes de altas frequências, representam flutuações irregulares no tráfego e os autores utilizam um modelo gaussiano para caracterizá-las, estimando os parâmetros através da Máxima Verossimilhança.

Uma das principais características da rede 5G é a escalabilidade, seja pela ótica de novos serviços ofertados pelos provedores, quanto por uma maior utilização por parte de consumidores, sensores, dispositivos inteligentes, entre outros. Dessa forma, estender a capacidade da rede torna-se fundamental para garantir essa escalabilidade. Nesse sentido, técnicas como fatiamento da rede são comumente tema de estudos para alcançar a escalabilidade, fornecendo uma maior flexibilidade na administração dos recursos. A utilização de recursos virtualizados e automatizados, fundamentais na rede 5G, são agravantes no ponto de vista de segurança, pois os provedores devem garantir que o fatiamento seja eficaz, evitando que agentes externos possam interromper o serviço e garantindo que o plano de dados continue íntegro. Nesse contexto, Benslimen *et al.* propõem um arcabouço utilizando o modelo ARIMA para predizer ataques, e o modelo LSTM para predizer anomalias e falhas [Benslimen et al., 2021].

Com o avanço das tecnologias, os meios de comunicação sem fio tornaram-se extremamente populares, fazendo com que a tecnologia associada evolua contínua e rapidamente para suportar a comunicação de dados em tempo real com qualidade, como a realização de vídeo chamadas. No entanto, na rede 5G, diversos sensores e dispositivos também são parte fundamental das comunicações, sendo imprescindível estabelecer uma robusta proteção do ponto de vista de infraestrutura, privacidade de usuários e, sobretudo, *software* para esses dispositivos [Zhang et al., 2019a]. Lopez-Martin *et al.* propõem a utilização de codificadores automáticos variacionais condicionais (*Conditional Variational Autoencoders*) para integrar os rótulos de intrusão dentro das camadas de decodificação, permitindo ser utilizado para predição de ataques e reconstrução de informações faltantes [Lopez-Martin et al., 2017]. Por ter apenas uma única etapa de treinamento, o modelo torna-se útil no que tange a otimização de recursos computacionais. Pela sua complexidade, a rede 5G exige o desenvolvimento de arquiteturas e soluções com alta resiliência. Ahmad *et al.* categorizam os principais desafios como [Ahmad et al., 2018]:

- ***flash network traffic***, que representa um aumento significativo de aparelhos e dispositivos conectados à rede, podendo ser contornado através da melhoria dos recursos existentes ou da adição de mais recursos conforme a demanda aumenta, usando redes definidas por *software* ou funções de rede virtualizadas;

- **integridade no plano de usuários**, que também é de vital importância, sendo fundamental a utilização de criptografia ponta a ponta. Aplicações específicas podem demandar a utilização de outras camadas de segurança neste plano, além da comunicação criptografada.

A comunicação móvel possibilita grandes avanços científicos ao longo de suas gerações, pois flexibiliza e universaliza a troca de informações em tempo real, tornando os usuários e dispositivos ubíquos, através de computação móvel, redes de sensores, entre outros. Entretanto, essas características também contribuem para que os ataques aumentem de forma exponencial, porque a superfície de ataque aumenta conforme mais dispositivos são conectados e há possibilidade desses mesmos dispositivos serem vetores de ataques distribuídos, causando prejuízos financeiros, roubo de informações e até mesmo uma guerra eletrônica. O tráfego dentro de uma célula, geralmente apresenta flutuações recorrentes e possui rajadas a qualquer instante, assemelhando-se com o comportamento de pessoas, que possuem características aleatórias no deslocamento ao longo do dia. A análise de segurança das redes tem ganhado foco em diversos campos de pesquisa, sobretudo na detecção de anomalias. Entretanto, a detecção em tempo real torna-se desafiadora em função da quantidade de dados gerados pelos dispositivos, pois requer um monitoramento ininterrupto de eventos, processos e mensagens na infraestrutura [Ariyaluran Habeeb et al., 2019]. As técnicas de detecção de anomalias em tempo real, em função dos desafios previamente listados, utilizam de forma majoritária ferramentas estatísticas por possuírem baixo custo computacional [Ahmad et al., 2017b]. Estudos apresentam os métodos de Holt-Winters [De Assis et al., 2017] e detecção do ponto de mudança (*change point detection*) [Tartakovsky et al., 2013] para detectar anomalias em redes de computadores. Ho Bang *et al.* propõem um modelo utilizando cadeia oculta semi-markoviana (*Hidden Semi-Markovian Model* - HSMM) para detecção de ataques de sinalização na rede LTE (*Long-Term Evolution*) [ho Bang et al., 2017]. Um processo é dito semi-markoviano quando a probabilidade de ocorrer uma mudança de um estado oculto para outro estado depende do tempo decorrido a partir do estado atual. Uma das vantagens da utilização desse modelo é a capacidade de capturar propriedades estatísticas do tráfego de rede e, uma vez utilizando os parâmetros do modelo do HSMM, é possível detectar anomalias de acordo com sua probabilidade ou entropia [Yu, 2010].

Além de técnicas estatísticas clássicas, o aprendizado de máquina também é amplamente utilizado para realizar detecção de anomalias e tráfego. No entanto, para aplicações em tempo real, tais algoritmos precisam de uma sequência ininterrupta de dados, podendo ocasionar um custo computacional superior em função do grande fluxo de informações a serem processadas, podendo limitar em alguns casos a acurácia da predição. A arquitetura da rede 5G tende a ser complexa. Fu *et al.* abordam dois desafios para o gerenciamento do tráfego de rede [Fu et al., 2018]. O primeiro é o fato de a rede ser heterogênea e a simultaneidade de diferentes redes com características distintas tornar a predição de tráfego mais complexa. O segundo é o fato de a rede ser majoritariamente implementada utilizando SDN e NFV, pois, com o fatiamento da rede, os serviços são utilizados de maneira autônoma em infraestruturas compartilhadas e todo o tráfego gerado por cenários distintos é unificado na infraestrutura, tornando o núcleo da rede praticamente imprevisível. Uma das técnicas promissoras é o aprendizado por reforço profundo (*Deep Reinforcement Learning*), que oferece ganhos significativos especialmente em situ-

ações com alta latência e congestionamento da rede, por exemplo. Os métodos baseados no aprendizado por reforço profundo podem aprender informações de roteamento e padrão de tráfego e, assim, gerenciar de forma mais eficiente os recursos da rede [Fu et al., 2018].

A rede 5G tem grande potencial para fomentar a implementação das cidades inteligentes, agregando serviços públicos através de comunicação ubíqua de sensores, dispositivos móveis, câmeras de segurança, entre outros. Além disso, a utilização de celulares inteligentes (*smartphones*) com o sistema operacional Android acrescenta um potencial risco. Sistemas baseados em Android representam 85% do mercado global e, consequentemente, se tornam alvos de ataques massivos [Lu et al., 2021]. É possível realizar a instalação de aplicativos de terceiros, aumentando consideravelmente a chance de implementar *botnets* ou vazamento de informações pessoais.

Do ponto de vista do gerenciamento da interface de transmissão aérea, a estimação das informações de estado do canal (*Channel State Information* - CSI), que representa as propriedades do canal de rádio, continua a ser um dos desafios fundamentais das redes 5G. O estado do canal, representado pelo CSI, tem um impacto significativo na alocação de recursos de rádio e gerenciamento de interferência, sendo utilizado para a determinação dos parâmetros da camada física do enlace. Devido à impossibilidade de envio frequente do valor do CSI pelo receptor, é fundamental que o transceptor seja capaz de estimar acuradamente o valor de CSI para permitir uma comunicação efetiva e otimizada no enlace. Métodos tradicionais para estimação do CSI possuem alta complexidade computacional e não são adequados para o uso em 5G devido ao emprego de tecnologias que aumentam o tráfego de dispositivos móveis, como MIMO (*Multiple Input Multiple Output*) massivo e ondas milimétricas. Luo *et al.* propõem um algoritmo de predição de CSI, denominado OCEAN, baseado em dados históricos de comunicação 5G [Luo et al., 2020]. Primeiramente, são identificadas diversas características importantes que afetam o CSI em um enlace de rádio, como faixa de frequência, localização, horário, temperatura, umidade do ar e tempo. Então, considerando a relação espaço-temporal do CSI, os autores projetam um arcabouço de aprendizado que consiste em uma combinação de duas redes neurais convolucionais (CNN) e uma memória longa de curto prazo (LSTM). A arquitetura do sistema proposto consiste em duas etapas de treinamento, uma *offline* e outra *online*. A etapa *offline* é responsável por treinar a rede com dados históricos. A etapa *online* ocorre com o sistema implementado, em funcionamento. Nessa etapa, a cada 5 minutos uma atualização do valor de CSI medido é utilizada para treinar novamente a rede. Dessa forma, utilizando a retroalimentação *online*, os valores previstos estão sempre sendo corrigidos e se adaptando às mudanças reais sofridas pelo canal, proporcionando resultados mais estáveis nas aplicações em sistemas de comunicação 5G.

4.6. Desafios Futuros para a Rede de Próxima Geração (6G)

O desenvolvimento de novas tecnologias surge para aprimorar as gerações antecessoras. A tecnologia desenvolvida para rede 5G contribui de forma significativa para diminuição da latência das redes móveis através da utilização de novas faixas de frequências com comprimento de onda milimétricas, utilização inteligente do espectro e redefinição do núcleo da rede [Giordani et al., 2020]. No entanto, mesmo antes de sua completa implementação, a rede possui limitações e a sua sucessora já vem sendo amplamente es-

tudada para resolver problemas principalmente de automação e inteligência artificial, com uma quantidade de dispositivos conectados ainda maior e novos conceitos.

A próxima geração das redes de telecomunicações deverá comportar uma quantidade crescente de terminais inteligentes, tais como celulares e sensores, disponibilizar aplicações de tempo real e prover inteligência e confiança embarcadas na infraestrutura de rede. Para atender a esses requisitos, a sexta geração das redes móveis, 6G, vislumbra o uso de novas tecnologias de inteligência artificial, de cadeia de blocos (*blockchain)* e de fornecimento de serviços para Internet das Coisas. A rede 6G está sendo desenvolvida para contornar as limitações existentes atualmente e possibilitar a utilização de novos paradigmas, como novas interações homem-homem e homem-máquina, utilização de frequências na faixa de terahertz, redes tridimensionais, comunicações quânticas, superfícies refletoras inteligentes, entre outras [Chowdhury et al., 2020]. Diversos campos de estudo ainda precisam de aprimoramento para o desenvolvimento e implementação da rede 6G. Estima-se que as taxas de transmissão serão da ordem de 1 Tb/s superando em cinquenta vezes a capacidade da rede 5G [Dogra et al., 2021]. Esta seção apresenta alguns conceitos com foco na futura geração de redes móveis e seus principais desafios no campo da segurança.

As aplicações 6G futuras apresentarão requisitos rigorosos e exigirão recursos de rede estendidos em comparação com as redes 5G desenvolvidas atualmente [Alwis et al., 2021]. Na rede 6G, todos os dispositivos de ponta são concebidos para se conectarem à Internet e os aplicativos de inteligência artificial serão amplamente usados por esses dispositivos. A maioria das aplicações de inteligência artificial são orientadas a dados, aumentando a preocupação com a segurança e privacidade dos dados coletados [Sun et al., 2020]. A privacidade dos clientes pode ser comprometida caso haja o vazamento de dados ou o comprometimento dos modelos de aprendizado [Cunha Neto et al., 2020].

A principal característica da rede 6G é uma conectividade ainda maior através do conceito de *Internet of Everything* (IoE), sendo uma integração entre sensores, dispositivos e qualquer objeto conectado. A IoE pode ser considerada uma extensão da Internet das Coisas abrangendo dados, processos, pessoas e dispositivos [Chowdhury et al., 2020]. O uso de inteligência artificial e questões relacionadas à privacidade de dados são temas em constante ascensão, sendo cada vez mais necessário que questões como a mitigação de vazamento de informações estejam presentes nas arquiteturas do núcleo das redes. As redes veiculares, por exemplo, possuem serviços que precisam de dados transmitidos em tempo real que necessitam de latência praticamente nula, o que não é oferecido com as tecnologias atuais. Com as redes 6G, latências menores que 1 ms serão possíveis e, com isso, a capacidade de controlar remotamente veículos poderá ser alcançada. A segurança de redes veiculares é foco de diversos estudos. Os ataques, como em qualquer outra infraestrutura interconectada, possuem os mesmos propósitos, como ganho de informação e degradação do serviço. No caso de redes veiculares, os ataques podem ser classificados em quatro grupos [Hasrouny et al., 2017]: i) os que representam risco para a interface de comunicação; ii) os que apresentam risco para *software* e *hardware*; iii) os que apresentam riscos para os sensores e iv) os que representam um risco para a infraestrutura. Por ser uma rede altamente dinâmica, os algoritmos de aprendizado de máquina são promissores para realizar detecção em sistemas de detecção de intrusão [Tang et al., 2020].

A utilização da rede 6G tem como objetivo aprimorar a fidelidade nas comunicações, tendo como desafios estabelecer comunicações ultra confiáveis e de baixa latência (*Ultra-Reliable Low Latency Communications* - URLLC). Esses conceitos permitem contribuir de maneira significativa em diversas áreas de missão crítica, permitindo por exemplo, que a **comunicação tátil** seja implementada para garantir que as interações físicas em tempo real sejam executadas, como a teleoperação. Outro conceito que ganhará notoriedade e demandará uma infraestrutura robusta é a holografia. A holografia é uma técnica que utiliza artifícios óticos para projetar luz e fornecer imagem em três dimensões, sendo objeto de estudos principalmente para a telemedicina, provendo atendimento médico em áreas remotas ou e realização de procedimentos cirúrgicos [Giordani et al., 2020].

Os mecanismos de inteligência artificial escaláveis e distribuídos são parte fundamental das redes 6G para proverem o conceito de Auto-X (autoconfiguração, automonitoramento, autocura e auto-otimização) sem qualquer envolvimento humano [Porambage et al., 2021]. Há esforços contínuos de especificação para integrar nativamente elementos de inteligência artificial em redes futuras, envolvendo operação em laço fechado e técnicas de automação por lógica difusa de operações de gerenciamento de rede, incluindo a segurança [Porambage et al., 2021]. Nas redes 6G, mecanismos de inteligência artificial serão instanciados mais próximos da fonte de dados de interesse para garantir a latência ultrabaixa, enquanto funções de aprendizado de máquina serão distribuídas pela rede para obter ganhos de desempenho devido a modelos otimizados e tomada de decisão em conjunto. No entanto, restrições práticas de elementos de rede, como restrições computacionais e conectividade intermitente, constituem um desafio em aberto. Existem também desafios complexos para o uso generalizado de técnicas de aprendizado de máquina aplicadas à cibersegurança, como facilitador da cibersegurança ou como uma técnica que pode levar a problemas de segurança em certas circunstâncias. Problemas de segurança relacionados ao uso massivo de técnicas de aprendizado de máquina estão relacionados:

- **a confiabilidade**, já que fomentam uma grande dependência desses mecanismos em redes futuras;
- **a visibilidade**, pois há uma necessidade de uma visão clara e inteligível dos esquemas baseados em aprendizado de máquina, mas normalmente esses esquemas operam como um esquema oculto para o utilizador;
- **a ética e a responsabilidade**, pois a justiça na aplicação de regras e ações previstas pelos mecanismos de aprendizado, assim como o gerenciamento de responsabilidades com entidades autônomas que operam em um ambiente de tecnologia da informação e da comunicação são tarefas complicadas, incluindo operações de segurança 6G;
- **a escalabilidade e a viabilidade**, já que para configurações distribuídas de aprendizado de máquina, como o aprendizado federado, as transmissões de dados devem ser protegidas e a privacidade preservada. A escalabilidade é um desafio em termos de recursos de computação, comunicação e armazenamento necessários;
- **os modelos de resiliência de dados**, que devem ser protegidos e robustos nas fases de aprendizagem e inferência. Uma das principais soluções previstas para as redes

6G é o uso de cadeia de blocos para uma estrutura de compartilhamento de dados distribuída, transparente e segura;

- **a privacidade**, pois diferentes técnicas de aprendizado podem ser aplicadas para a recuperação e correlação de dados, infringindo a privacidade dos usuários.

As tecnologias de livros-razão distribuídos (*Distributed Ledger Technology* – DLT), como a tecnologia de cadeia de blocos, são apontadas como viabilizadoras das redes 6G. As vantagens adicionais das DLTs como desintermediação, imutabilidade, não-repúdio, prova de procedência, integridade e pseudonimato são particularmente importantes para habilitar diferentes serviços em redes 6G com confiança e segurança distribuídas. Como tecnologias de análise de dados podem ser uma fonte para novos vetores de ataque, como ataques de envenenamento na fase de treinamento e ataques de evasão na fase de teste, as DLTs têm o potencial de proteger a integridade dos dados por meio de registros imutáveis e confiança distribuída entre diferentes partes interessadas, permitindo a confiança em sistemas multi-domínios. Contratos inteligentes baseados em DLT podem ser utilizados para definir Acordos de Nível de Confiança (*Trust Level Agreements* - TLAs) [Varalakshmi e Judgi, 2017] e a responsabilidade de cada parte ou entre componentes, em caso de violações do TLA. A DLT pode ser usada em gerenciamento seguro de funções de rede virtuais (*Virtual Network Functions* - VNFs), corretagem de fatia segura, gerenciamento de nível de serviço de segurança automatizado, gerenciamento de IoT escalonável, *roaming* seguro e manipulação de *offloading* [Camilo et al., 2020]. As cadeias de blocos também são candidatas-chaves para a preservação da privacidade em redes 6G centradas em conteúdo.

Questões relativas à segurança, sigilo e privacidade de dados terão grande foco nas redes 6G. Atualmente, com o aumento do poder computacional, diversos atacantes utilizam processamento gráfico, por exemplo, para otimizar as operações e realizar ataques de força bruta, invadindo serviços públicos e coletando informações sensíveis e pessoais. Atualmente, ainda não é possível identificar e mitigar de maneira antecipada tais eventos com precisão, pois as técnicas de ataques são cada vez mais sofisticadas. Como será o principal meio de comunicação entre os serviços inteligentes, a rede 6G deve garantir sobretudo a privacidade de dados. Será necessário que a inteligência artificial consiga identificar anomalias em tempo real em redes críticas, por exemplo, redes hospitalares em que milhares de dispositivos serão de alguma forma conectados enviando informações sensíveis. No entanto, ainda existem limitações pois os algoritmos possuem foco no desempenho e não em questões éticas. A criptografia é recomendada para qualquer serviço, no entanto na rede 6G será mandatória. A computação quântica eventualmente se tornará acessível a consumidores, o que trará grande impacto nos atuais algoritmos de criptografia. Com isso, estudos já estão sendo realizados para desenvolver a **criptografia pós-quântica** para evitar ataques a partir de computadores quânticos.

A computação quântica foi projetada para uso em redes de comunicação 6G para detecção, mitigação e prevenção de vulnerabilidades de segurança. A comunicação assistida por computação quântica é uma nova área de pesquisa que investiga as possibilidades de substituir os canais quânticos por canais de comunicação clássicos sem ruído para atingir uma confiabilidade extremamente alta em 6G. Com os avanços da computação quântica, prevê-se que a criptografia quântica segura seja introduzida no mundo

pós-quântico. O problema do logaritmo discreto, que é a base da criptografia assimétrica atual, pode se tornar solucionável em tempo polinomial com o desenvolvimento de algoritmos quânticos. Uma vez que a computação quântica tende a usar a natureza quântica da informação, ela pode fornecer intrinsecamente aleatoriedade absoluta e segurança para melhorar a qualidade da transmissão [Porambage et al., 2021].

4.7. Exemplo Prático para Detecção de Anomalias

Nesta seção, são abordados dois cenários práticos utilizando códigos escritos na linguagem Python. Essa linguagem é escolhida pela facilidade na implementação através de diversas bibliotecas já existentes para aprendizado de máquina, manipulação de grandes conjunto de dados, além de modelos estatísticos, mantendo o código sintetizado e eficiente. No primeiro exemplo, é realizada a predição de tráfego utilizando a rede neural de memória longa de curto prazo (LSTM). No segundo, é apresentado o algoritmo SVM de uma classe amplamente utilizado para detecção de anomalias em fluxos de redes. Por último, o código apresentado passa a focar na predição de séries temporais utilizando os modelos ARIMA e SARIMA. No primeiro exemplo, utilizando a rede neural LSTM, são utilizadas métricas coletadas a partir do tráfego da rede sem fio da Universidade Federal Fluminense. Os dados são processados e armazenados em arquivos distintos no formato `CSV` (*Comma Separated Value*) para facilitar o treinamento. Esse arquivo contém o cálculo da entropia de Shannon para as características da 5-tuplas do fluxo TCP (IP de Origem, IP de Destino, Porta de Origem, Porta de Destino e Protocolo) durante seis dias com janela de 1 hora [Reis et al., 2020]. Após isso, aplica-se a transformada discreta *wavelet* para separação das componentes lineares e não lineares, sendo a última a utilizada nesse arquivo. Para o segundo exemplo, foi utilizado um conjunto de dados reais rotulado com alguns tipos de ataques a clientes de uma rede de banda larga residencial [Andreoni Lopez e Mattos, 2021]. Como a proposta do classificador é utilizar apenas uma classe alvo, todos os fluxos que possuem qualquer tipo de ataque são considerados anormais enquanto os demais são ditos normais.

A primeira etapa para execução do código é realizar a instalação das bibliotecas necessárias. Entretanto, existem outras ferramentas que permitem realizar a programação diretamente no navegador, como Google Colab[6]. O Colab é uma plataforma da Google que permite a execução de códigos escritos em Python, assim como outras linguagens, sem a necessidade de configuração ou instalação de bibliotecas no computador. O código-fonte desse exemplo está disponível no repositório no Github[7]. Para realizar a execução, inicialmente são instaladas e importadas as seguintes bibliotecas: Sci-kit Learn, TensorFlow, Numpy e Pandas, como mostrado no Código Fonte 4.1. O TensorFlow [8] e o Scikit-learn [9] são bibliotecas que contêm diversos algoritmos de aprendizado de máquina. O NumPy [10] é uma biblioteca utilizada para funções matemáticas em geral, como álgebra linear, transformada de Fourier, entre outras.

Após carregadas todas as bibliotecas necessárias, o conjunto de dados é carregado

[6]Disponível em https://colab.research.google.com.
[7]Disponível em https://github.com/gnnbarbosa/lstmPrediction.
[8]Disponível em https://www.tensorflow.org/.
[9]Disponível em https://scikit-learn.org/.
[10]Disponível em https://numpy.org.

Código Fonte 4.1. Importação das bibliotecas para execução do exemplo.

```
1  import numpy
2  import time
3  import matplotlib.pyplot as plt
4  from pandas import read_csv
5  import math
6  from tensorflow.keras.models import Sequential
7  from tensorflow.keras.layers import Dense
8  from tensorflow.keras.layers import LSTM
9  from sklearn.preprocessing import MinMaxScaler
10 from sklearn.metrics import mean_squared_error
```

na memória utilizando a biblioteca Pandas [11]. Em seguida, são realizadas etapas de pré-processamento nos dados para otimização do treinamento. O Código Fonte 4.2 monstra a implementação dessa etapa. Dependendo da análise a ser realizada, essa etapa é essencial, principalmente para otimizar recursos computacionais através da redução de dados carregados em memória. O modelo LSTM, em particular, possui certa sensibilidade na escala dos dados, sendo uma boa prática realizar a normalização sem que haja perda na informação. Na maioria dos casos, na normalização, é utilizado o intervalo entre 0 e 1. A função **MinMaxScaler** da biblioteca **Sci-kit Learn** executa a etapa de normalização dos dados. Na próxima etapa, o conjunto de dados é dividido entre treinamento e validação. Nesse exemplo, é utilizado 70% do conjunto de dados para realizar o treinamento e os 30% restantes são utilizados para validação. Uma função *create_dataset* é utilizada para criação de um novo conjunto de dados. Essa função possui dois parâmetros, sendo o primeiro referente aos valores originais da matriz e o segundo, *look_back*, representa o número de entradas anteriores. Assim, será formado um novo conjunto de dados no qual X representa o valor da entropia em um determinado tempo t e Y o valor da entropia no instante $t+1$.

Realizado o pré-processamento do conjunto de dados, são configurados os parâmetros da rede neural LSTM. Nesse exemplo, o modelo possui uma camada oculta de entrada, quatro neurônios e a camada de saída que realiza a predição de apenas um único valor. A rede é treinada em 100 épocas, tamanho do lote igual a 1 e utilização da função de ativação padrão (***sigmoide***). O Código Fonte 4.3 apresenta os parâmetros utilizados no treinamento. Nesse caso, a **função de perda** utilizada é o erro médio quadrático (***mean squared error***). Essa função é fundamental para calcular o custo que o modelo deve minimizar durante o treinamento.

Após o treinamento do modelo através da função ***model.fit*** é feita a predição, conforme apresenta o Código Fonte 4.4. Nessa etapa, é necessário mensurar o desempenho da rede neural treinada. É aplicada uma transformação inversa às predições, antes de calcular as taxas de erros, para recuperar o conjunto de dados original. Destaca-se que até o momento anterior à aplicação da transformada inversa, os dados estão transformados pelos métodos de pré-processamento.

[11] Disponível em https://pandas.pydata.org/.

Código Fonte 4.2. Processamento do conjunto de dados.

```
# Convertendo os valores em uma matriz
def create_dataset(dataset, look_back=1):
        dataX, dataY = [], []
        for i in range(len(dataset)-look_back-1):
                a = dataset[i:(i+look_back), 0]
                dataX.append(a)
                dataY.append(dataset[i + look_back, 0])
        return numpy.array(dataX), numpy.array(dataY)

# Carregando o Dataset
start_time = time.time()
df = read_csv('__.csv', usecols=[5], \
        engine='python')
dataset = df.values

# Normalização do dataset
scaler = MinMaxScaler(feature_range=(0, 1))
dataset = scaler.fit_transform(dataset)

# Separando o dataset para treinamento e teste
train_size = int(len(dataset) * 0.70)
test_size = len(dataset) - train_size
train, test = dataset[0:train_size,:], dataset[train_size:len(dataset),:]

# formatacao para X=t and Y=t+1
look_back = 1
trainX, trainY = create_dataset(train, look_back)
testX, testY = create_dataset(test, look_back)

# formatação dos dados de entrada para [samples, time steps, features]
trainX = numpy.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))
testX = numpy.reshape(testX, (testX.shape[0], 1, testX.shape[1]))
```

Código Fonte 4.3. Parametrização do modelo LSTM.

```
# Criação e treinamento do modelo LSTM
model = Sequential()
model.add(LSTM(4, input_shape=(1, look_back)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(trainX, trainY, epochs=100, batch_size=1, verbose=2)
```

Por fim, para validar as métricas de desempenho e qualidade do modelo, é calculada a raiz do erro quadrático médio (***Root Mean-Square Error*** - RMSE). Essa métrica é

Código Fonte 4.4. Predição dos pontos da série.

```
# Realizando as predições
trainPredict = model.predict(trainX)
testPredict = model.predict(testX)
trainPredict = scaler.inverse_transform(trainPredict)
trainY = scaler.inverse_transform([trainY])
testPredict = scaler.inverse_transform(testPredict)
testY = scaler.inverse_transform([testY])
```

frequentemente utilizada para avaliar a diferença entre os valores previstos e valores observados em modelos obtidos a partir de algoritmos de aprendizado de máquina. Também é verificada ao término da execução do código, a quantidade de tempo necessária para finalizar a execução. Os resultados obtidos são apresentados pelo Código Fonte 4.5. Para exibir de maneira gráfica o resultado do modelo, o Código Fonte 4.6 é usado para gerar um gráfico da predição contendo os valores originais da série e a predição realizada. Cabe destacar que, dependendo do estudo a ser realizado, é interessante armazenar os valores preditos em listas.

Código Fonte 4.5. Resultados do modelo e tempo de execução.

```
# Resultados
trainScore = math.sqrt(mean_squared_error(trainY[0], trainPredict[:,0]))
testScore = math.sqrt(mean_squared_error(testY[0], testPredict[:,0]))
print ("\n")
print("=====================================================")
print("Resultados do algoritmo")
print("=====================================================")
print('Valor RMSE do treinamento: %.2f' % (trainScore))
print('Valor RMSE do teste: %.2f' % (testScore))
print("Tempo de treinamento e predição: %.2f segundos" % \
        (time.time() - start_time))
print ("\n")
```

No segundo exemplo, é utilizado o classificador *One-Class* SVM para detectar fluxos anormais no tráfego de uma rede. Para isso, utiliza-se um conjunto de dados com informações reais, rotulado com diversos tipos de ataque. Como a premissa desse classificador é identificar apenas uma classe alvo, é necessário treinar o modelo com os fluxos normais e fluxos anormais. Inicialmente, como no exemplo anterior, é realizada a importação das bibliotecas necessárias: Pandas, NumPy e Scikit-Learn, conforme apresentado no Código Fonte 4.7.

A etapa de processamento para realização do treinamento é a mais importante. Um fluxo contém informações que nem sempre são utilizadas, sendo fundamental que haja uma redução da dimensão do conjunto de dados para otimizar os recursos computacionais, principalmente alocação de memória. Evita-se, dessa forma, que informações

Código Fonte 4.6. Configuração do gráfico da série temporal e predição realizada

```
# Plotagem da série original e valores preditos
trainPredictPlot = numpy.empty_like(dataset)
trainPredictPlot[:, :] = numpy.nan
trainPredictPlot[ \
        look_back:len(trainPredict)+look_back, :] = trainPredict
testPredictPlot = numpy.empty_like(dataset)
testPredictPlot[:, :] = numpy.nan
testPredictPlot[len(trainPredict)+(look_back*2)+ \
        1:len(dataset)-1, :] = testPredict
plt.plot(dataframe, label='Valores Originais', color="deepskyblue")
plt.plot(testPredictPlot, label='Valores da Predição', \
        color="darkred", linestyle='--')
plt.ylabel('Componentes não lineares da entropia para protocolo', \
        fontsize=16)
plt.xlabel('Amostras',fontsize=16)
plt.legend(loc="lower left", prop={'size': 13})
plt.yticks(fontsize=16)
plt.xticks(fontsize=16)
plt.show()
```

Código Fonte 4.7. Importação das bibliotecas

```
from __future__ import division
import numpy as np
import pandas as pd
from sklearn import svm
from sklearn import metrics
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
```

que não sejam pertinentes sejam armazenadas em memória. Cabe destacar que a utilização da biblioteca Pandas possui grande eficiência para conjunto de dados com tamanhos até 1 GB. Para conjunto de dados maiores, a biblioteca passa a oferecer baixo desempenho sendo necessário utilizar outras ferramentas como o Apache Spark[12] e Apache Hadoop[13]. No exemplo, é utilizado um conjunto de dados com quarenta e sete características de fluxo e são utilizadas apenas três para identificar anomalias, sendo elas *sflow_fbytes*, *sflow_bbytes* e *class*. Uma função de normalização é aplicada para otimizar o treinamento do modelo. Em seguida, é utilizada a característica *class* para criar dois novos tipos de classes, entre normal e anormal. Nesse caso, como os valores das classes correspondem a valores numéricos, 0 representa fluxos normais e qualquer outro valor representa algum

[12]https://spark.apache.org/
[13]https://hadoop.apache.org/

tipo de anomalia. Então, é feito um mapeamento no qual anomalias são representadas pelo valor -1 e tráfego normal é caracterizado por 1. Por último, o conjunto de dados é dividido entre treinamento e validação com 70% e 30% respectivamente. O Código Fonte 4.8 apresenta essa etapa.

Código Fonte 4.8. Processamento do conjunto de dados.

```
# carregando o Dataset e seleção das características
df = pd.read_csv('__.csv', low_memory=False)
features = ["sflow_fbytes", "sflow_bbytes", "class"]

#redução do dataset apenas com as características necessárias para treinar o modelo
df = df[features]

# normalização dos dados
df["sflow_fbytes"] = np.log((df["sflow_fbytes"]).astype(float))
df["sflow_bbytes"] = np.log((df["sflow_bbytes"]).astype(float))

# este dataset esta rotulado com diversos tipos de ataque. Serão tratados então 0 como trafego "normal" e diferente
# de 0 "anomalia" atribuindo 1 e -1 respectivamente
df.loc[df['class'] == 0, "ataque"] = 1
df.loc[df['class'] != 0, "ataque"] = -1

#
target = df['ataque']

#separação das do dataset em treinamento de teste
trainX, testX, trainY, testY = train_test_split(df, target, train_size = 0.70)
```

A última etapa desse exemplo é apresentada no Código Fonte 4.9, no qual o modelo é treinado e o seu desempenho é avaliado. Na função **svm.OneClassSVM**, são repassados os parâmetros *nu*, *kernel* e *gamma* que representam a precisão da regressão, o mapeamento de observações não lineares em um espaço de dimensão superior e a influência que um único exemplo de treinamento pode alcançar, respectivamente. Além disto, são utilizadas as principais métricas de desempenho de aprendizado de máquinas para avaliar o quão satisfatório é o modelo. Nesse caso, são avaliadas as métricas tanto do treinamento quanto da detecção. Assim, no código, são utilizadas as funções da biblioteca Scikit-Learn: *metrics.accuracy_score* que infere a acurácia; *metrics.precision_score* que avalia a precisão; *metrics.recall_score* diz respeito à revocação; e *metrics.f1_score* utilizada para mensurar medida F1, que é a média harmônica entre precisão e revocação.

Código Fonte 4.9. Validação do modelo.

```
#treinamento do modelo
model = svm.OneClassSVM(nu=0.2, kernel='rbf', gamma='auto')
model.fit(trainX)
values_preds = model.predict(trainX)
values_targs = train_target
print "============TREINAMENTO============"
print "Acurácia: %.2f" % (100 *  metrics.accuracy_score(values_targs,
 ↪  values_preds)), str("%")
print "Precisão: %.2f" % (100  * metrics.precision_score(values_targs,
 ↪  values_preds)), str("%")
print "Recall: %.2f" % (100* metrics.recall_score(values_targs,
 ↪  values_preds)), str("%")
print "F1-Score: %.2f" % (100 * metrics.f1_score(values_targs,
 ↪  values_preds)), str("%")

#Validação do modelo
values_preds_test = model.predict(test_data)
values_targs = test_target
print "============VALIDAÇÃO============"
print "Acurácia: %.2f" % (100* metrics.accuracy_score(values_targs,
 ↪  values_preds_test)), str("%")
print "Precisão: %.2f" % (100 * metrics.precision_score(values_targs,
 ↪  values_preds_test)), str("%")
print "Recall: %.2f" % (100 * metrics.recall_score(values_targs,
 ↪  values_preds_test)), str("%")
print "F1-Score: %.2f" % (100 * metrics.f1_score(values_targs,
 ↪  values_preds_test)), str("%")
print "================================="
```

4.8. Considerações Finais

As redes móveis propiciaram um grande avanço nas comunicações, pois em função de suas características físicas podem estar presentes em qualquer localidade. Além disto, a comunicação de dados através destas redes, influenciou como as relações humanas se desenvolveram ao longo dos últimos anos, pois diversas aplicações foram criadas e outras aprimoradas para possibilitar a interação da informação em tempo real, transmissão de vídeos, entre outras formas de interação humana mediadas pelas redes móveis. Com a chegada da rede 5G, as relações entre dispositivos serão abordadas de maneira acentuada, permitindo que a inteligência artificial seja aplicada para otimização de recursos energéticos, hídricos e até mesmo alimentares. A rede 5G possui avanços significativos com relação as gerações anteriores, tais como Serviços de Missão Crítica (MCS) e Banda larga móvel aprimorada (eMBB). Veículos autônomos e serviços de saúde remotos são exemplos de novos paradigmas para serviços de missão crítica, no qual a baixa latência e as altas taxas de transmissão são fundamentais para serem efetivas, pois podem ocasionar danos irreparáveis. Além disso, as altas taxas de transmissão permitirão uma

maior demanda de vídeos com alta definição, gerando um tráfego acentuado nas redes. Essas premissas podem ser aplicadas para casos em que sejam necessários a utilização de realidade virtual (*Virtual Reality* - VR), realidade aumentada (*Augumented Reality* - AR) ou realizar o monitoramento e rastreamento de doenças infectocontagiosas através de sistemas de vigilância com câmeras infravermelhas.

A predição de tráfego e detecção de anomalias em redes sempre foram atividades desafiadoras em virtude da grande massa de dados. O cenário torna-se ainda mais complexo com na rede 5G, em virtude de a análise em tempo real demandar um processamento de grandes fluxos de dados principalmente na borda da rede. Este capítulo apresentou as principais técnicas de predição de tráfego utilizando modelos estocásticos amplamente estudados, além de ferramentas de aprendizado de máquinas para detecção de anomalias. As **redes definidas por *software*** fazem parte do núcleo das comunicações das tecnologias 5G e 6G, ambas com foco em um maior número de dispositivos conectados nas bordas. A predição de tráfego permite que um controlador SDN realize modificações no roteamento, distribuindo políticas para todos os segmentos de rede e evite congestionamentos de maneira pró-ativa. Além disso, permite a implementação dinâmica de novos serviços como balanceadores de carga, *firewalls*, comutadores virtuais, entre outros, através da **virtualização de funções de rede**. A predição de anomalias por sua vez, permite realizar contramedidas para mitigar os efeitos de ataques na eminência de ocorrer. Com o aumento de dispositivos móveis, técnicas como o aprendizado federado poderão ser úteis para realizar a detecção de ameaças garantido a privacidade dos dados e distribuindo o processamento. Esse paradigma permite ainda, que diferentes entidades possam contribuir no treinamento do modelo sem que os dados possam ser acessados entre as organizações.

O capítulo analisou ainda desafios futuros para as redes de próxima geração, 6G. Os desafios relativos ao aprendizado de máquina são ainda mais críticos para a próxima geração de redes móveis, pois nessas novas redes as aplicações de aprendizado de máquina farão parte do núcleo da rede, em contraposição ao que ocorre nas redes 5G, em que as aplicações de aprendizado de máquina são acessórias ao gerenciamento e controle da rede. O capítulo desenvolveu ainda três exemplos práticos de aplicação de aprendizado de máquina para a gerência da segurança em redes 5G. A predição de tráfego com o modelo auto-regressivo integrado de médias móveis (*Autoregressive Integrated Moving Average* - ARIMA) mostrou a aplicação de uma técnica clássica para a regressão de séries temporais. A predição de tráfego com a rede neural de memória longa de curto prazo (*Long Short-Term Memory* - LSTM) evidenciou o uso de uma aplicação atual, aplicando bibliotecas de código aberto, e com alta acurácia na predição. Por fim, o modelo de máquina de vetor de suporte de uma única classe (*One Class Support Vector Machine* - OCSVM) exemplificou um cenário de identificação de anomalias nos fluxos de rede através do treinamento de um mecanismo de aprendizado de máquina voltado para detecção de amostras destoantes.

Referências

[Ahmad et al., 2020] Ahmad, A., Harjula, E., Ylianttila, M. e Ahmad, I. (2020). Evaluation of machine learning techniques for security in SDN. Em *2020 IEEE Globecom Workshops (GC Wkshps*, p. 1–6.

[Ahmad et al., 2017a] Ahmad, I., Kumar, T., Liyanage, M., Okwuibe, J., Ylianttila, M. e Gurtov, A. (2017a). 5G security: Analysis of threats and solutions. Em *2017 IEEE Conference on Standards for Communications and Networking*, p. 193–199. IEEE.

[Ahmad et al., 2018] Ahmad, I., Kumar, T., Liyanage, M., Okwuibe, J., Ylianttila, M. e Gurtov, A. (2018). Overview of 5G security challenges and solutions. *IEEE Communications Standards Magazine*, 2(1):36–43.

[Ahmad et al., 2019] Ahmad, I., Shahabuddin, S., Kumar, T., Okwuibe, J., Gurtov, A. e Ylianttila, M. (2019). Security for 5G and beyond. *IEEE Communications Surveys & Tutorials*, 21(4):3682–3722.

[Ahmad et al., 2017b] Ahmad, S., Lavin, A., Purdy, S. e Agha, Z. (2017b). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147. Online Real-Time Learning Strategies for Data Streams.

[Alawe et al., 2018] Alawe, I., Ksentini, A., Hadjadj-Aoul, Y. e Bertin, P. (2018). Improving traffic forecasting for 5G core network scalability: A machine learning approach. *IEEE Network*, 32(6):42–49.

[Alwis et al., 2021] Alwis, C. D., Kalla, A., Pham, Q.-V., Kumar, P., Dev, K., Hwang, W.-J. e Liyanage, M. (2021). Survey on 6G frontiers: Trends, applications, requirements, technologies and future research. *IEEE Open Journal of the Communications Society*, 2:836–886.

[Andreoni Lopez et al., 2021] Andreoni Lopez, M., Baddeley, M., Lunardi, W. T., Pandey, A. e Giacalone, J.-P. (2021). Towards secure wireless mesh networks for uav swarm connectivity: Current threats, research, and opportunities. Em *Proceeding of 3rd International Workshop on Wireless Sensors and Drones in Internet of Things (WiDroIT) 2021*, p. 1–6.

[Andreoni Lopez e Mattos, 2021] Andreoni Lopez, M. e Mattos, D. (2021). Resumo de grandes volumes de dados com filtro de bloom: Uma abordagem eficiente para aprendizado profundo com redes neurais convolucionais em fluxos de rede. Em *Anais do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 532–545, Porto Alegre, RS, Brasil. SBC.

[Andreoni Lopez et al., 2019] Andreoni Lopez, M., Mattos, D. M., Duarte, O. C. M. e Pujolle, G. (2019). Toward a monitoring and threat detection system based on stream processing as a virtual network function for big data. *Concurrency and Computation: Practice and Experience*, 31(20):e5344.

[Andreoni Lopez et al., 2016] Andreoni Lopez, M., Mattos, D. M. F. e Duarte, O. C. M. (2016). An elastic intrusion detection system for software networks. *Annals of Telecommunications*, 71(11):595–605.

[Ariyaluran Habeeb et al., 2019] Ariyaluran Habeeb, R. A., Nasaruddin, F., Gani, A., Targio Hashem, I. A., Ahmed, E. e Imran, M. (2019). Real-time big data processing for anomaly detection: A survey. *International Journal of Information Management*, 45:289–307.

[Benslimen et al., 2021] Benslimen, Y., Sedjelmaci, H. e Manenti, A.-C. (2021). Attacks and failures prediction framework for a collaborative 5G mobile network. *Computing*, 103(6):1165–1181.

[Bochie et al., 2020] Bochie, K., Gilbert, M., Gantert, L., Barbosa, M., Medeiros, D. e Campista, M. (2020). Aprendizado profundo em redes desafiadoras: Conceitos e aplicações. Em *Minicursos do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 140–189. SBC.

[Bockelmann et al., 2016] Bockelmann, C., Pratas, N., Nikopour, H., Au, K., Svensson, T., Stefanovic, C., Popovski, P. e Dekorsy, A. (2016). Massive machine-type communications in 5G: Physical and mac-layer solutions. *IEEE Communications Magazine*, 54(9):59–65.

[Boukerche et al., 2020] Boukerche, A., Tao, Y. e Sun, P. (2020). Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems. *Computer Networks*, 182:107484.

[Camilo et al., 2020] Camilo, G. F., Rebello, G. A. F., de Souza, L. A. C. e Duarte, O. C. M. B. (2020). AutAvailChain: Automatic and secure data availability through blockchain. Em *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, p. 1–6.

[Chadza et al., 2020] Chadza, T., Kyriakopoulos, K. G. e Lambotharan, S. (2020). Analysis of hidden markov model learning algorithms for the detection and prediction of multi-stage network attacks. *Future Generation Computer Systems*, 108:636–649.

[Chakraborty et al., 2020] Chakraborty, P., Corici, M. e Magedanz, T. (2020). A comparative study for time series forecasting within software 5G networks. Em *2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS)*, p. 1–7.

[Chowdhury et al., 2020] Chowdhury, M. Z., Shahjalal, M., Ahmed, S. e Jang, Y. M. (2020). 6G wireless communication systems: Applications, requirements, technologies, challenges, and research directions. *IEEE Open Journal of the Communications Society*, 1:957–975.

[Cunha et al., 2019] Cunha, V. A., da Silva, E., de Carvalho, M. B., Corujo, D., Barraca, J. P., Gomes, D., Granville, L. Z. e Aguiar, R. L. (2019). Network slicing security: Challenges and directions. *Internet Technology Letters*, 2(5):e125.

[Cunha Neto et al., 2020] Cunha Neto, H. N., Mattos, D. M. F. e Fernandes, N. C. (2020). Privacidade do usuário em aprendizado colaborativo: Federated learning, da teoria à prática. *Minicursos do Simpósio Brasileiro de Segurança de Informação e de Sistemas Computacionais - SBSeg*, 20:142–195.

[De Assis et al., 2017] De Assis, M. V. O., Hamamoto, A. H., Abrão, T. e Proença, M. L. (2017). A game theoretical based system using holt-winters and genetic algorithm with fuzzy logic for DoS/DDoS mitigation on sdn networks. *IEEE Access*, 5:9485–9496.

[De Ree et al., 2019] De Ree, M., Mantas, G., Radwan, A., Mumtaz, S., Rodriguez, J. e Otung, I. E. (2019). Key management for beyond 5G mobile small cells: A survey. *IEEE Access*, 7:59200–59236.

[Di Bernardino e Brogi, 2019] Di Bernardino, E. e Brogi, G. (2019). Hidden markov models for advanced persistent threats. *International Journal of Security and Networks*, 14:181.

[Dogra et al., 2021] Dogra, A., Jha, R. K. e Jain, S. (2021). A survey on beyond 5G network with the advent of 6G: Architecture and emerging technologies. *IEEE Access*, 9:67512–67547.

[Fernandez Maimo et al., 2018] Fernandez Maimo, L., Perales Gomez, A. L., Garcia Clemente, F. J., Gil Perez, M. e Martinez Perez, G. (2018). A self-adaptive deep learning-based system for anomaly detection in 5G networks. *IEEE Access*, 6:7700–7712.

[Fu et al., 2018] Fu, Y., Wang, S., Wang, C.-X., Hong, X. e McLaughlin, S. (2018). Artificial intelligence to manage network traffic of 5g wireless networks. *IEEE Network*, 32(6):58–64.

[Giordani et al., 2020] Giordani, M., Polese, M., Mezzavilla, M., Rangan, S. e Zorzi, M. (2020). Toward 6G networks: Use cases and technologies. *IEEE Communications Magazine*, 58(3):55–61.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y. e Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

[Hanbanchong e Piromsopa, 2012] Hanbanchong, A. e Piromsopa, K. (2012). SARIMA based network bandwidth anomaly detection. Em *2012 Ninth International Conference on Computer Science and Software Engineering (JCSSE)*, p. 104–108.

[Hasrouny et al., 2017] Hasrouny, H., Samhat, A. E., Bassil, C. e Laouiti, A. (2017). VANet security challenges and solutions: A survey. *Vehicular Communications*, 7:7–20.

[ho Bang et al., 2017] ho Bang, J., Cho, Y.-J. e Kang, K. (2017). Anomaly detection of network-initiated LTE signaling traffic in wireless sensor and actuator networks based on a hidden semi-markov model. *Computers and Security*, 65:108–120.

[Huang et al., 2017] Huang, C.-W., Chiang, C.-T. e Li, Q. (2017). A study of deep learning networks on mobile traffic forecasting. Em *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, p. 1–6.

[Jiang e Schotten, 2019] Jiang, W. e Schotten, H. D. (2019). Neural network-based fading channel prediction: A comprehensive overview. *IEEE Access*, 7:118112–118124.

[Kaelbling et al., 1996] Kaelbling, L. P., Littman, M. L. e Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.

[Kakkar, 2020] Kakkar, A. (2020). A survey on secure communication techniques for 5G wireless heterogeneous networks. *Information Fusion*, 62:89–109.

[Khan et al., 2019] Khan, R., Kumar, P., Jayakody, D. N. K. e Liyanage, M. (2019). A survey on security and privacy of 5G technologies: Potential solutions, recent advancements, and future directions. *IEEE Communications Surveys & Tutorials*, 22(1):196–248.

[Kromkowski et al., 2019] Kromkowski, P., Li, S., Zhao, W., Abraham, B., Osborne, A. e Brown, D. E. (2019). Evaluating statistical models for network traffic anomaly detection. Em *2019 Systems and Information Engineering Design Symposium (SIEDS)*, p. 1–6.

[Kumar Dwivedi et al., 2018] Kumar Dwivedi, R., Pandey, S. e Kumar, R. (2018). A study on machine learning approaches for outlier detection in wireless sensor network. Em *2018 8th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, p. 189–192.

[Lal et al., 2017] Lal, S., Taleb, T. e Dutta, A. (2017). NFV: Security threats and best practices. *IEEE Communications Magazine*, 55(8):211–217.

[Li et al., 2020] Li, J., Li, Z., Tyson, G. e Xie, G. (2020). Your privilege gives your privacy away: An analysis of a home security camera service. Em *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, p. 387–396.

[Li et al., 2011] Li, Y., Kaur, B. e Andersen, B. (2011). Denial of service prevention for 5G. *Wireless Personal Communications*, 57(3):365–376.

[Lobato et al., 2021] Lobato, A. G. P., Andreoni Lopez, M., Cardenas, A. A., Duarte, O. C. M. B. e Pujolle, G. (2021). A fast and accurate threat detection and prevention architecture using stream processing. *Concurrency and Computation: Practice and Experience*, e6561.

[Lopez-Martin et al., 2017] Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A. e Lloret, J. (2017). Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in IoT. *Sensors*, 17(9).

[Lu et al., 2021] Lu, N., Li, D., Shi, W., Vijayakumar, P., Piccialli, F. e Chang, V. (2021). An efficient combined deep neural network based malware detection framework in 5G environment. *Computer Networks*, 189:107932.

[Luo et al., 2020] Luo, C., Ji, J., Wang, Q., Chen, X. e Li, P. (2020). Channel state information prediction for 5G wireless communications: A deep learning approach. *IEEE Transactions on Network Science and Engineering*, 7(1):227–236.

[Medeiros et al., 2019] Medeiros, D., Cunha Neto, H., Andreoni, M., Magalhães, L., Silva, E., Borges, A., Fernandes, N. e Menezes, D. (2019). *Análise de Dados em Redes Sem Fio de Grande Porte: Processamento em Fluxo em Tempo Real, Tendências e Desafios*, p. 142–195. Sociedade Brasileira de Computação.

[Nie et al., 2017] Nie, L., Jiang, D., Yu, S. e Song, H. (2017). Network traffic prediction based on Deep Belief Network in wireless mesh backbone networks. Em *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, p. 1–5.

[Nieto et al., 2019] Nieto, A., Acien, A. e Fernandez, G. (2019). Crowdsourcing analysis in 5G IoT: Cybersecurity threats and mitigation. *Mobile Networks and Applications*, 24(3):881–889.

[Popovski et al., 2018] Popovski, P., Trillingsgaard, K. F., Simeone, O. e Durisi, G. (2018). 5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view. *IEEE Access*, 6:55765–55779.

[Porambage et al., 2021] Porambage, P., Gür, G., Moya Osorio, D. P., Livanage, M. e Ylianttila, M. (2021). 6G security challenges and potential solutions. Em *2021 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*, p. 622–627.

[Ramakrishnan e Soni, 2018] Ramakrishnan, N. e Soni, T. (2018). Network traffic prediction using recurrent neural networks. Em *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, p. 187–193.

[Reddy e Thilagam, 2020] Reddy, K. e Thilagam, P. (2020). Naïve bayes classifier to mitigate the DDoS attacks severity in ad-hoc networks. *International Journal of Communication Networks and Information Security*, 12:221–226.

[Reis et al., 2020] Reis, L. H. A., Magalhães, L. C. S., de Medeiros, D. S. V. e Mattos, D. M. (2020). An unsupervised approach to infer quality of service for large-scale wireless networking. *Journal of Network and Systems Management*, 28(4):1228–1247.

[Scott-Hayward et al., 2015] Scott-Hayward, S., Natarajan, S. e Sezer, S. (2015). A survey of security in software defined networks. *IEEE Communications Surveys & Tutorials*, 18(1):623–654.

[Sedjelmaci, 2021] Sedjelmaci, H. (2021). Cooperative attacks detection based on artificial intelligence system for 5G networks. *Computers and Electrical Engineering*, 91:107045.

[Sivanathan et al., 2019] Sivanathan, A., Gharakheili, H. H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A. e Sivaraman, V. (2019). Classifying iot devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing*, 18(8):1745–1759.

[Sone et al., 2020] Sone, S. P., Lehtomäki, J. J. e Khan, Z. (2020). Wireless traffic usage forecasting using real enterprise network data: Analysis and methods. *IEEE Open Journal of the Communications Society*, 1:777–797.

[Sun et al., 2020] Sun, Y., Liu, J., Wang, J., Cao, Y. e Kato, N. (2020). When machine learning meets privacy in 6G: A survey. *IEEE Communications Surveys Tutorials*, 22(4):2694–2724.

[Swarnkar e Hubballi, 2016] Swarnkar, M. e Hubballi, N. (2016). OCPAD: One class naive bayes classifier for payload based anomaly detection. *Expert Systems with Applications*, 64:330–339.

[Tang et al., 2020] Tang, F., Kawamoto, Y., Kato, N. e Liu, J. (2020). Future intelligent and secure vehicular network toward 6G: Machine-learning approaches. *Proceedings of the IEEE*, 108(2):292–307.

[Tartakovsky et al., 2013] Tartakovsky, A. G., Polunchenko, A. S. e Sokolov, G. (2013). Efficient computer network anomaly detection by changepoint detection methods. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):4–11.

[Trinh et al., 2019] Trinh, H. D., Zeydan, E., Giupponi, L. e Dini, P. (2019). Detecting mobile traffic anomalies through physical control channel fingerprinting: A deep semi-supervised approach. *IEEE Access*, 7:152187–152201.

[Varalakshmi e Judgi, 2017] Varalakshmi, P. e Judgi, T. (2017). Multifaceted trust management framework based on a trust level agreement in a collaborative cloud. *Computers & Electrical Engineering*, 59:110–125.

[Wan et al., 2020] Wan, Y., Xu, K., Xue, G. e Wang, F. (2020). IoTArgos: A multi-layer security monitoring system for internet-of-things in smart homes. Em *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, p. 874–883.

[Wang et al., 2017] Wang, J., Tang, J., Xu, Z., Wang, Y., Xue, G., Zhang, X. e Yang, D. (2017). Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach. Em *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, p. 1–9.

[Wang et al., 2021] Wang, Y., Masoud, N. e Khojandi, A. (2021). Real-time sensor anomaly detection and recovery in connected automated vehicle sensors. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1411–1421.

[Wasicek, 2020] Wasicek, A. (2020). The future of 5G smart home network security is micro-segmentation. *Network & Security*, 2020(11):11–13.

[Wazid et al., 2020] Wazid, M., Das, A. K., Shetty, S., Gope, P. e Rodrigues, J. J. (2020). Security in 5G-enabled internet of things communication: issues, challenges, and future research roadmap. *IEEE Access*, 9:4466–4489.

[Wu et al., 2020] Wu, D., Nekovee, M. e Wang, Y. (2020). Deep learning-based autoencoder for m-user wireless interference channel physical layer design. *IEEE Access*, 8:174679–174691.

[Xu et al., 2015] Xu, Q., Liao, Y., Miskovic, S., Mao, Z. M., Baldi, M., Nucci, A. e Andrews, T. (2015). Automatic generation of mobile app signatures from traffic observations. Em *2015 IEEE Conference on Computer Communications (INFOCOM)*, p. 1481–1489.

[Yang et al., 2021] Yang, H., Li, X., Qiang, W., Zhao, Y., Zhang, W. e Tang, C. (2021). A network traffic forecasting method based on sa optimized arima–bp neural network. *Computer Networks*, 193:108102.

[Yao et al., 2019] Yao, J., Han, Z., Sohail, M. e Wang, L. (2019). A robust security architecture for SDN-based 5G networks. *Future Internet*, 11(4):85.

[Yu, 2010] Yu, S.-Z. (2010). Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243. Special Review Issue.

[Zafeiropoulos et al., 2020] Zafeiropoulos, A., Fotopoulou, E., Peuster, M., Schneider, S., Gouvas, P., Behnke, D., Müller, M., Bök, P.-B., Trakadas, P., Karkazis, P. e Karl, H. (2020). Benchmarking and profiling 5G verticals' applications: An industrial IoT use case. Em *2020 6th IEEE Conference on Network Softwarization (NetSoft)*, p. 310–318.

[Zhan et al., 2020] Zhan, M., Li, Y., Yang, X., Cui, W. e Fan, Y. (2020). NSAPs: A novel scheme for network security state assessment and attack prediction. *Computers and Security*, 99:102031.

[Zhang et al., 2019a] Zhang, C., Patras, P. e Haddadi, H. (2019a). Deep learning in mobile and wireless networking: A survey. *IEEE Communications Surveys Tutorials*, 21(3):2224–2287.

[Zhang et al., 2019b] Zhang, S., Wang, Y. e Zhou, W. (2019b). Towards secure 5G networks: A survey. *Computer Networks*, 162:106871.

[Zhu et al., 2019] Zhu, G., Zan, J., Yang, Y. e Qi, X. (2019). A supervised learning based QoS assurance architecture for 5G networks. *IEEE Access*, 7:43598–43606.

