Mitigação de Ataques DDoS na IoT por meio de Aprendizado de Máquina e Virtualização de Funções de Rede

Guilherme Werneck de Oliveira

Dissertação apresentada ao
Instituto de Matemática e Estatística
da Universidade de São Paulo
para obtenção do título de
Mestre em Ciências

Programa: Ciência da Computação
Orientador: Prof. Dr. Daniel Macêdo Batista
Coorientador: Prof. Dr. Aldri Luiz dos Santos

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro do CNPq
(Processo 130762/2021-0)

São Paulo
24 de Agosto de 2022

Mitigação de Ataques DDoS na IoT por meio de Aprendizado de Máquina e Virtualização de Funções de Rede

Guilherme Werneck de Oliveira

Esta versão da dissertação contém
as correções e alterações sugeridas
pela Comissão Julgadora durante a
defesa da versão original do trabalho,
realizada em 24 de Agosto de 2022.

Uma cópia da versão original está
disponível no Instituto de Matemática e
Estatística da Universidade de São Paulo.

Comissão julgadora:

Prof. Dr. Daniel Macêdo Batista (orientador) – IME-USP
Prof. Dr. Jó Ueyama – ICMC-USP
Dr. Pedro Henrique Gomes da Silva – Ericsson Research

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Agradecimentos

"If I have been able to see further, it was only because I stood on the shoulders of giants."
— Isaac Newton

Agradeço à minha mãe por estar sempre ao meu lado suportando minhas decisões e me apoiando em todos os desafios enfrentados, ao meu orientador Prof. Daniel Macêdo Batista pela oportunidade e pelos direcionamentos na pesquisa, ao meu coorientador Prof. Aldri Luiz dos Santos e à Profª. Michele Nogueira por todo conhecimento compartilhado e lições aprendidas.

Resumo

Guilherme Werneck de Oliveira. **Mitigação de Ataques DDoS na IoT por meio de Aprendizado de Máquina e Virtualização de Funções de Rede**. Dissertação (Mestrado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2022.

A Internet das coisas (IoT – *Internet of Things*) tem passado por uma rápida popularização, alcançando uma grande diversidade de domínios de aplicações. Como consequência, cada vez mais dispositivos IoT com características diversas são implantados em uma variedade de ambientes públicos e privados, tornando-se progressivamente objetos comuns da vida cotidiana. Por outro lado, a infraestrutura física de sistemas heterogêneos é complexa e exige soluções eficientes e dinâmicas para o gerenciamento de desempenho e segurança de redes, em um nível que permita a implantação padronizada e de fácil replicação em indústrias, prédios e cidades inteligentes. Uma abordagem que vem ganhando espaço quando o intuito é responder às ameaças de segurança na IoT consiste no uso de recursos virtualizados de rede por meio da virtualização de funções de rede (NFV – *Network Functions Virtualization*). A literatura apresenta alguns trabalhos que utilizam a NFV para detectar e mitigar ameaças de segurança em redes IoT, mas poucos são os indicadores de desempenho do contexto de rede explorados para a composição de uma solução de maior eficiência. Dessa forma, este trabalho apresenta o desenvolvimento de uma abordagem baseada no desempenho do contexto de rede para mitigar ataques distribuídos de negação de serviço (DDoS – *Distributed Denial of Service*) na IoT, investigando como indicadores de desempenho podem compor modelos de aprendizado de máquina (ML – *Machine Learning*) para o posicionamento de funções virtualizadas de redes (VNF – *Virtual Network Functions*). Modelos de ML foram treinados e testados a partir de indicadores de desempenho do contexto de rede para a realização de dois experimentos. O primeiro relacionado à predição do tempo de resposta de redes de Internet das coisas industriais (IIoT – *Industrial Internet of Things*). O segundo sobre o posicionamento de VNF de segurança, também em redes IIoT. Os respectivos resultados alcançados constituem em predições de tempos de resposta de rede que são no máximo 50 milissegundos maiores do que configurações realizadas por modelos de programação linear e classificações com 99,40% de acurácia para o correto posicionamento de VNF de segurança em redes IIoT sob ataques DDoS.

Palavras-chave: NFV. VNF. IoT. DDoS. Desempenho. Segurança. Aprendizado de Máquina.

Abstract

Guilherme Werneck de Oliveira. **Mitigating DDoS Attacks on IoT through Machine Learning and Network Functions Virtualization**. Thesis (Master's). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2022.

The Internet of Things (IoT) has undergone a rapid popularization, reaching a wide range of application domains. As a consequence, more and more IoT devices with diverse characteristics are deployed in a variety of public and private environments, progressively becoming common objects of everyday life. On the other hand, the physical infrastructure of heterogeneous systems is complex and requires efficient and dynamic solutions for managing network performance and security, at a level that allows standardized deployment and easy replication in smart industries, buildings and cities. An approach that has been gaining ground when the intention is to respond to security threats on IoT is the Network Functions Virtualization (NFV) usage. The literature presents some works that use NFV to detect and mitigate security threats on IoT networks, but few performance indicators of the network context are explored to compose a more efficient solution. Thus, this work presents the development of a network context performance-based approach to mitigate Distributed Denial of Service (DDoS) attacks on IoT, investigating how performance indicators can compose Machine Learning (ML) models for Virtual Network Functions (VNF) placement. ML models were trained and tested using network context performance indicators to carry out two experiments. The first is related to the prediction of the response time of Industrial Internet of Things (IIoT) networks. The second on the security VNF placement, also on IIoT networks. The respective results achieved are predictions of network response times that are at most 50 milliseconds longer than settings realized by linear programming models and classifications with 99.40% of accuracy for the correct placement of security VNF on IIoT networks under DDoS attacks.

Keywords: NFV. VNF. IoT. DDoS. Performance. Security. Machine Learning.

Lista de Figuras

2.1 Uma visão geral da IoT. Adaptado de Farris, Taleb *et al.* (2019). 5
2.2 Visão geral da virtualização de funções de rede na IoT. 6
2.3 Exemplo de um encadeamento de funções em uma rede tradicional. . . . 7
2.4 Exemplo de um encadeamento de funções de rede com NFV. 9
2.5 Arquitetura em camadas da IoT. 10
2.6 Escalando recursos de VNFs de segurança em redes IoT. Adaptado de Farris, Taleb *et al.* (2019). 14
2.7 Migrando recursos de VNFs de segurança em redes IoT. Adaptado de Farris, Taleb *et al.* (2019). 14

3.1 Níveis de processamento e capacidades em diferentes arquiteturas distribuídas. Adaptado de Yousefpour *et al.* (2019) e Lin *et al.* (2017). 22
3.2 Arquitetura vIoTHoneynet. Adaptado de Zarca, Bernabe, Skarmeta *et al.* (2020). 26
3.3 Arquitetura NETRA. Adaptado de Sairam *et al.* (2019). 28
3.4 Métricas utilizadas na avaliação de desempenho. 34
3.5 Tipos de ataques considerados e abordagens apresentadas. 34

4.1 Abordagem baseada em ML. 42
4.2 Estrutura do *framework*. 44
4.3 Impacto dos quatro componentes principais. 46
4.4 Desempenho da Regressão Linear para a previsão do tempo de resposta da rede. 47
4.5 Desempenho do XGBoost para a previsão do tempo de resposta da rede. . 48
4.6 Formas de posicionamento da VNF de segurança em redes IoT sob ataques DDoS. 50
4.7 Desempenho dos modelos de ML de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado. 55

4.8 Desempenho e importância dos indicadores para Árvore de Decisão de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado. 57
4.9 Desempenho e importância dos indicadores para Random Forest de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado. 58
4.10 Desempenho e importância dos indicadores para XGBoost de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado. 59
4.11 Desempenho dos modelos de ML de acordo com o posicionamento da VNF de segurança próximo ao atacante. 60
4.12 Desempenho e importância dos indicadores para Árvore de Decisão de acordo com o posicionamento da VNF de segurança próximo ao atacante. 62
4.13 Desempenho e importância dos indicadores para Random Forest de acordo com o posicionamento da VNF de segurança próximo ao atacante. 63
4.14 Desempenho e importância dos indicadores para XGBoost de acordo com o posicionamento da VNF de segurança próximo ao atacante. 64

A.1 Desempenho da Regressão Logística de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado. 69
A.2 Desempenho da Support Vector Machine de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado. 70
A.3 Desempenho do K Nearest Neighbors de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado. 70
A.4 Desempenho do Naive Bayes de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado. 70
A.5 Desempenho do K-Means de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado. 71
A.6 Desempenho do Hierarchical Cluster de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado. 71
A.7 Desempenho do Gaussian Mixture Model de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado. 71
A.8 Desempenho da Regressão Logística de acordo com o posicionamento da VNF de segurança próximo ao atacante. 72
A.9 Desempenho da Support Vector Machine de acordo com o posicionamento da VNF de segurança próximo ao atacante. 72
A.10 Desempenho do K Nearest Neighbors de acordo com o posicionamento da VNF de segurança próximo ao atacante. 72

A.11 Desempenho do Naive Bayes de acordo com o posicionamento da VNF de segurança próximo ao atacante. 73
A.12 Desempenho do K-Means de acordo com o posicionamento da VNF de segurança próximo ao atacante. 73
A.13 Desempenho do Hierarchical Cluster de acordo com o posicionamento da VNF de segurança próximo ao atacante. 73
A.14 Desempenho do Gaussian Mixture Model de acordo com o posicionamento da VNF de segurança próximo ao atacante. 74

Lista de Tabelas

2.1 Resumo das métricas de qualidade de serviço em NFV. Traduzido de ETSI (2014b). 12

3.1 Protocolos referentes às camadas da IoT. 21
3.2 Tipos de ataques a redes IoT e suas descrições. 23
3.3 Categorias e métricas de desempenho. 33
3.4 Panorama dos indicadores de desempenho para o contexto de segurança. 35
3.5 Panorama dos indicadores de desempenho para o contexto de IoT. 36
3.6 Comparação entre os trabalhos analisados e esta dissertação. 37

4.1 Acurácia dos modelos implementados. 46
4.2 Acurácia dos modelos com dados normalizados. 48
4.3 Racional qualitativo para inclusão de indicadores de desempenho no conjunto de dados. 52
4.4 Distribuição dos servidores em névoa no conjunto de dados de treinamento de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado. 56

Sumário

1 Introdução **1**

2 Conceitos Básicos **5**

2.1 Virtualização de Funções de Rede (NFV) 6

2.2 Redes IoT e Aplicações 9

2.3 Gerenciamento de Redes com NFV 11

2.4 Questões de Desempenho x Segurança com NFV na IoT 11

2.4.1 Separação do *Software* de Segurança do Hardware 15

2.4.2 Escalabilidade Sob Demanda e Tolerância a Falhas para VNF de Segurança 15

2.4.3 Suporte à Mobilidade de VNF de Segurança 16

2.4.4 Encadeamento de Serviços de Segurança de Rede 16

2.5 VNF para Modelos de Aprendizado de Segurança 17

2.6 Indicadores e Gerenciamento de Desempenho 18

3 Revisão da Literatura **19**

3.1 Análise dos Trabalhos Relacionados 24

3.2 Discussão Entre a Literatura e as Questões de Pesquisa 33

4 Abordagem Baseada no Desempenho do Contexto de Rede para Mitigar Ataques DDoS na IoT **39**

4.1 Predição do Tempo de Resposta em Ambientes SDN-Fog para Aplicações IIoT 40

4.1.1 IIoT em Névoa 41

4.1.2 O Conjunto de Dados 42

4.1.3 O *Framework* 43

4.1.4 Experimentos, Resultados e Discussões 45

4.2 Posicionamento Inteligente de VNF para Mitigar Ataques DDoS na IIoT . 49

4.2.1 O Conjunto de Dados 51

4.2.2 Modelos de ML para Posicionamento de VNF de Segurança . . . 52
4.2.3 Experimentos, Resultados e Discussões 54

5 Conclusão **65**

Apêndices

A Modelos de Aprendizado de Máquina com Baixo Desempenho **69**
A.1 Posicionamento da VNF de Segurança Próximo ao Dispositivo IoT Atacado 69
A.1.1 Modelos de Aprendizado Supervisionado 69
A.1.2 Modelos de Aprendizado Não Supervisionado 71
A.2 Posicionamento da VNF de Segurança Próximo ao Dispositivo Atacante . 72
A.2.1 Modelos de Aprendizado Supervisionado 72
A.2.2 Modelos de Aprendizado Não Supervisionado 73

Referências **75**

Capítulo 1

Introdução

A Internet das coisas (IoT – *Internet of Things*) tem passado por uma rápida popularização, alcançando uma grande diversidade de domínios de aplicações, como por exemplo cuidados com a saúde, monitoramento ambiental, automação residencial, mobilidade inteligente e Indústria 4.0 (V. D. da Silva *et al.*, 2021; Batista *et al.*, 2016; Rosário *et al.*, 2014). Como consequência, cada vez mais dispositivos IoT[1] com características diversas são implantados em uma variedade de ambientes públicos e privados, tornando-se progressivamente objetos comuns da vida cotidiana.

A IoT tem sido vista por muitos como o próximo estágio da Internet. Sua implementação possibilita a evolução na sociedade e indústria, como no caso das cidades inteligentes, pois com a IoT os sistemas de transporte, de controle de resíduos, de energia, entre outros, tornam-se mais eficientes e melhoram a qualidade de vida dos cidadãos.

Um fator que reforça ainda mais a presença da IoT, é a evolução dos meios de comunicação. Segundo dados mais recentes (5G-Americas, 2022; Omdia, 2021), a conexão global em 5G alcançou 701 milhões de conexões até o final do primeiro trimestre de 2022. Até 2026, é esperado que esse número chegue a 4,8 bilhões de conexões. Mas essa evolução não se restringe ao 5G. A comunicação em 6G vem para suportar requisitos ainda mais críticos em sistemas densos e heterogêneos, principalmente aqueles relacionados a altas taxas de transferência de dados, à baixa latência, ao uso eficiente de energia e ao gerenciamento de recursos da própria rede (Ericsson, 2022; Akyildiz *et al.*, 2020).

Por outro lado, a infraestrutura física de sistemas densos e heterogêneos é complexa e exige soluções eficientes e dinâmicas para o gerenciamento e a configuração de redes em um nível que permita a implantação padronizada e de fácil replicação em indústrias, prédios e cidades inteligentes (Alam *et al.*, 2020; Chi *et al.*, 2019).

Como ocorre em redes de telecomunicações de um modo geral, dois dos requisitos que se destacam em IoT, além da baixa disponibilidade de energia, são desempenho e segurança. No entanto, muitos dispositivos na IoT não oferecem suporte a mecanismos

[1] No decorrer deste trabalho, o termo "dispositivos IoT" será usado para descrever dispositivos de hardware voltados para aplicações de Internet das Coisas, como assistentes pessoais, relógios inteligentes, sensores e atuadores de um modo geral.

de segurança fortes e, portanto, podem ser alvos e até mesmo meios para uma série de ataques (Meneghello *et al.*, 2019). Dentre os vários tipos de ataques, vale ressaltar a gravidade dos ataques distribuídos de negação de serviço (DDoS – *Distributed Denial of Service*) que podem afetar tanto redes formadas por dispositivos IoT quanto a Internet de um modo geral (Cloudflare, 2021; Griffioen e Doerr, 2020).

De acordo com a empresa Kaspersky (Kaspersky, 2020; Kaspersky, 2019), os ataques a dispositivos IoT têm se tornado frequentes. Nos primeiros seis meses de 2019, foram contabilizados 105 milhões de ataques a dispositivos IoT oriundos de 276 mil endereços IP exclusivos. Esses ataques são dos mais variados tipos e se aproveitam de diversas vulnerabilidades, como aquela descoberta em agosto de 2020 pela empresa Check Point (CheckPoint, 2020) contra a assistente pessoal Alexa, que permitia a manipulação de seus *tokens* e a realização de ações em nome da vítima. Outro dado importante é que 28% das empresas que utilizam dispositivos IoT reportam ter encontrado incidentes envolvendo dispositivos conectados. Com o grande volume de dados gerados por sensores e dispositivos inteligentes, inclusive dados sigilosos, as consequências de tais incidentes podem ser graves, tanto do ponto de vista financeiro quanto do legal, como prevê a Lei Geral de Proteção de Dados Pessoais (LGPD)[2].

Uma abordagem que vem ganhando espaço quando o intuito é responder às ameaças na IoT consiste no uso de recursos virtualizados de rede por meio da virtualização de funções de rede (NFV – *Network Functions Virtualization*) (Alam *et al.*, 2020; Farris, Taleb *et al.*, 2019; Zarca, Bernabe, Farris *et al.*, 2018). A NFV apresenta um novo grau de flexibilidade e escalabilidade criando recursos de rede virtuais sob demanda, como *firewalls*, sistemas de detecção de intrusão (IDS – *Intrusion Detection Systems*) e sistemas de inspeção profunda de pacotes (DPI – *Deep Packet Inspection*). A abordagem de virtualização permite que várias instâncias de um mecanismo de detecção ou mitigação específico possam ser implementadas em diferentes locais na rede, considerando as restrições impostas pela ocorrência de eventos maliciosos (Farris, Taleb *et al.*, 2019) e pelo poder limitado de processamento e armazenamento de muitos dispositivos IoT.

Além da localização das funções virtualizadas, limitada pela capacidade dos dispositivos IoT e pelas particularidades dos ataques, há outras questões relacionadas ao uso de NFV. Uma delas diz respeito ao gerenciamento do desempenho dos recursos virtualizados. Para Laghrissi e Taleb (2019), nesse contexto, o verdadeiro desafio é produzir *software* eficiente e escalável para orquestrar as redes virtuais, a fim de permitir que estas redes sejam facilmente configuradas e tenham seus ciclos de vida gerenciados. Em Mijumbi *et al.* (2016), além da gestão e da orquestração de recursos, os autores também consideram a eficiência energética, a própria eficiência do recurso provisionado, a modelagem e a alocação de recursos como desafios em aberto.

Tudo isso se relaciona ao gerenciamento de desempenho em ambientes virtualizados, tópico que tem levantado diversos desafios de pesquisa, como orquestração de funções de rede virtualizadas (VNF – *Virtual Network Function*), otimização da cadeia de funções (SFC – *Service Function Chaining*), posicionamento de VNFs em localidades distintas, entre outros (Zarca, Bagaa *et al.*, 2020; Gupta *et al.*, 2019; Zhang *et al.*, 2016). Esses desafios de

[2] Lei Nº 13.709, de 14 de Agosto de 2018. Fonte: http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm. Acessado em 03 de Setembro de 2022.

pesquisa podem ser enfrentados por meio da aplicação de outros campos de estudos, como o aprendizado de máquina (ML – *Machine Learning*), por exemplo. ML pode ser aplicado em vários problemas distintos em redes, desde a camada física até a camada de aplicação (Akyildiz *et al.*, 2020; Boutaba *et al.*, 2018). O mais importante para este trabalho é o gerenciamento e a orquestração de recursos da rede de acordo com sua demanda.

O objetivo geral desta dissertação é apresentar uma abordagem baseada no desempenho do contexto de rede para mitigar ataques DDoS na IoT, investigando como as métricas - ou indicadores - de desempenho podem compor, como variáveis de decisão, modelos de ML para o posicionamento adequado de VNF. Com isso, são alcançadas as seguintes contribuições:

- Identificação das abordagens da literatura em que a NFV tem sido utilizada para detectar e mitigar ataques na IoT;
- Análise de como as métricas - ou indicadores - de desempenho são aferidos nas abordagens encontradas;
- Avaliação de diferentes composições de métricas - ou indicadores - de desempenho como variáveis de decisão em diferentes modelos de ML para mitigar ataques DDoS na IoT, de acordo com o posicionando de VNF;
- Disponibilização de um algoritmo de posicionamento de VNF de segurança baseado em ML;
- Publicização da implementação em Python de modelos de ML juntamente com o conjunto de dados correspondente a redes IIoT sob ataques DDoS.

O foco na utilização da tecnologia NFV visa não só considerar aspectos ligados à redução de custos, devido à sua utilização sob demanda, mas também à redução de forma eficaz dos riscos e dos prejuízos causados por ataques na IoT. Assim, este trabalho responde as seguintes questões de pesquisa (QP) com o intuito de validar a hipótese de que **existem vantagens em implementar estratégias baseadas em ML para mitigar ataques DDoS em redes IoT[3], utilizando como variáveis de decisão as métricas - ou indicadores - de desempenho do próprio recurso NFV e de recursos da rede**:

1. QP1: Como os recursos NFV estão sendo utilizados para detectar e mitigar ataques na IoT?
2. QP2: Quais são as métricas - ou indicadores - utilizados para aferir o desempenho do uso de NFV na detecção e mitigação de ataques em redes IoT e como eles impactam o gerenciamento de recursos virtualizados?

Modelos de ML foram treinados e testados a partir de indicadores de desempenho do contexto de rede para a realização de dois experimentos. O primeiro relacionado à predição do tempo de resposta de redes de Internet das coisas industriais (IIoT – *Industrial Internet of Things*). O segundo sobre o posicionamento de VNF de segurança, também em redes IIoT. Os respectivos resultados alcançados constituem em predições de tempos de resposta

[3] No decorrer deste trabalho, os termos "redes IoT" e "redes IIoT" serão usados para descrever redes que visam suportar requisitos específicos de aplicações de Internet das Coisas e Internet das Coisas Industriais, como baixa latência, redes sem fio, multi-protocolos etc.

de rede que são no máximo 50 milissegundos maiores do que configurações realizadas por modelos de programação linear e classificações com 99,40% de acurácia para o correto posicionamento de VNF de segurança em redes IIoT sob ataques DDoS.

No momento da escrita desta dissertação, os resultados alcançados durante o mestrado foram disseminados em duas publicações e em uma submissão:

- Guilherme Werneck de Oliveira, Jonathan Rangel Porto *et al.* "Virtualização de Funções de Rede na IoT: Um Panorama do Gerenciamento de Desempenho x Segurança". Publicado em: *Livro de Minicursos do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC).* 2021. Cap. 3 (Oliveira, Porto *et al.*, 2021): Neste capítulo de livro são apresentadas questões e o estado da arte relacionados ao desempenho do uso de NFV na detecção e mitigação de ameaças em redes específicas de IoT. Também é apresentado um estudo de caso sobre o MENTORED Testbed, uma ferramenta aberta à comunidade que disponibiliza cenários realísticos para simulação de vulnerabilidades específicas e para avaliação de mecanismos de segurança de redes de forma controlada;

- Guilherme Werneck de Oliveira, Rodrigo Ney *et al.* "Predicting Response Time in SDN-Fog Environments for IIoT Applications". Publicado em: *Proceedings of the 2021 IEEE Latin-American Conference on Communications (LATINCOM).* 2021 (Oliveira, Ney *et al.*, 2021): Neste artigo é apresentada uma solução que utiliza as próprias variáveis do contexto de rede para predição do tempo de resposta de aplicações de IIoT, utilizando modelos de ML supervisionados e não supervisionados em um ambiente com Redes Definidas por Software (SDN – *Software Defined Networks*);

- Guilherme Werneck de Oliveira, Michele Nogueira *et al.* "Intelligent VNF Placement to Mitigate DDoS Attacks on Industrial IoT". Submetido para: *IEEE Transactions on Network and Service Management (TNSM).* 2022: Este artigo apresenta uma evolução dos experimentos iniciais realizados em (Oliveira, Ney *et al.*, 2021). Nele é apresentado um estudo de modelos de ML, supervisionados e não supervisionados, treinados e testados com indicadores de desempenho da rede para diferentes estratégias de posicionamento de uma VNF de segurança na mesma rede IIoT, agora sob ataques DDoS.

Esta dissertação está organizada da seguinte forma: o Capítulo 2 apresenta as características relacionadas à NFV, à construção, arquiteturas e protocolos de redes que suportam dispositivos IoT, bem como os aspectos de desempenho de NFV voltados à segurança em ambientes IoT. Em seguida, o Capítulo 3 apresenta a revisão da literatura sobre o desempenho da NFV quando empregada para a detecção e mitigação de ameaças na IoT, capítulo este que responde às questões QP1 e QP2. Na sequência, no Capítulo 4, é apresentado o objetivo geral da pesquisa além de resultados de acordo com dois experimentos baseados em ML e indicadores de desempenho de redes IIoT. O primeiro relacionado à predição do tempo de resposta de redes e o segundo sobre o posicionamento de VNF de segurança. Dessa forma, este capítulo valida a hipótese. Por fim, o Capítulo 5 apresenta a conclusão do trabalho realizado.

Capítulo 2

Conceitos Básicos

A Figura 2.1 ilustra uma visão geral da IoT. Os dispositivos IoT, na camada mais baixa da figura, interagem com o ambiente físico nos papéis de sensores e atuadores. Estes dispositivos são necessários para o correto funcionamento das aplicações IoT, representadas na camada mais alta da figura. As comunicações entre os vários componentes ilustrados na figura acontecem graças à infraestrutura de tecnologia da informação, disponível na Internet por meio de recursos na borda e no núcleo da rede.

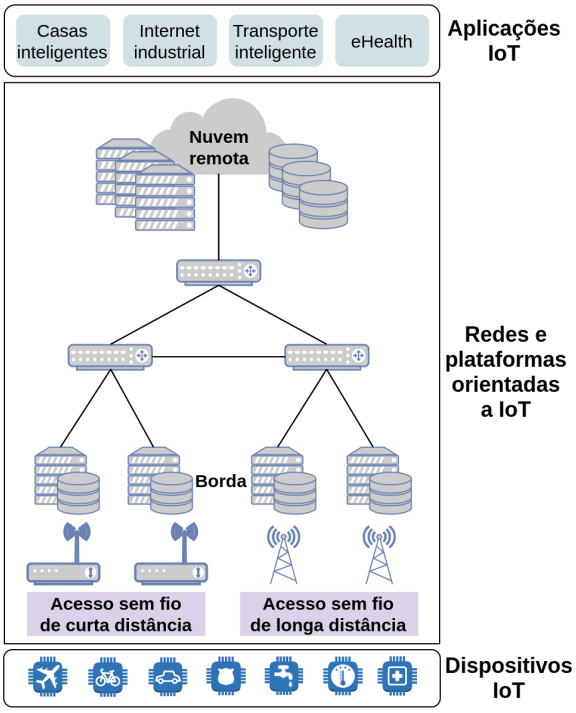


Figura 2.1: *Uma visão geral da IoT. Adaptado de* Farris, Taleb *et al.* (2019).

Diversos componentes de rede podem fazer parte do cenário ilustrado na Figura 2.1 para garantir a comunicação entre os dispositivos IoT e suas aplicações no mundo real. Tanto componentes físicos, como pontos de acesso, *switches*, roteadores, quanto funções de rede, como sistema de nomes de domínio (DNS – *Domain Name System*), protocolo

de configuração dinâmica de endereços (DHCP – *Dynamic Host Configuration Protocol*), tradução de endereço de rede (NAT – *Network Address Translation*), entre outras.

Essas funções de rede podem ser implementadas e executadas sobre diferentes tipos de plataformas. Uma delas é a virtualizada, como ilustra a Figura 2.2 baseada na padronização de arquiteturas NFV (MANO – *Management and Orchestration Architectural Framework*) (ETSI, 2022) especificada pelo Instituto Europeu de Normas de Telecomunicações (ETSI – *European Telecommunications Standards Institute*). Os dispositivos reais, na camada física, utilizam os serviços das VNFs disponíveis na camada de serviços. As VNFs são instanciadas graças à infraestrutura NFV, na camada de virtualização, e são gerenciadas pelo gerenciador de VNF e pelo gerenciador de infraestrutura virtualizada na camada de orquestração. Em um cenário ideal, algumas das VNFs fornecem serviços de segurança e nenhuma das VNFs possui vulnerabilidades. O foco desta dissertação é na utilização de VNFs que implementem mecanismos de segurança da rede, como *firewall*, IDS e DPI.

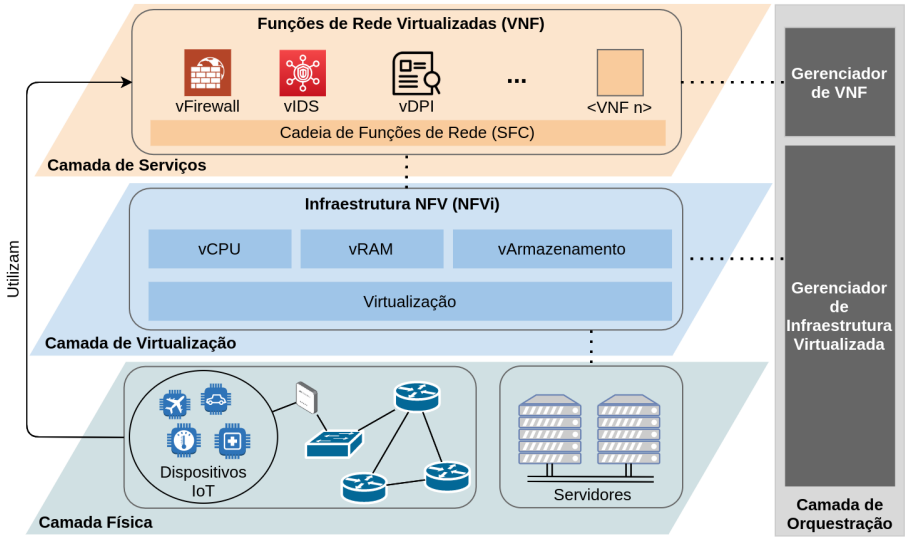


Figura 2.2: *Visão geral da virtualização de funções de rede na IoT.*

As seções seguintes apresentam os conceitos básicos para que os componentes apresentados na Figura 2.2 sejam compreendidos: a Seção 2.1 apresenta a definição e as características de NFVs, relacionando-as com as camadas de virtualização, serviços e orquestração. A Seção 2.2 define a IoT e apresenta aplicações nesse contexto, relacionando-as à camada física. A Seção 2.3 descreve conceitos presentes no gerenciamento de redes que utilizam NFV, relacionando-os com a camada de orquestração; e a Seção 2.4 apresenta conceitos que ajudam a compreender o *tradeoff* entre desempenho e segurança quando utiliza-se NFV para detecção e mitigação de ameaças de segurança na IoT, relacionando as camadas física, de virtualização, de serviços e de orquestração.

2.1 Virtualização de Funções de Rede (NFV)

As funções de rede como filtragem de pacotes, tradução de endereços e balanceamento de carga são essenciais em muitas redes de computadores. Mesmo em uma rede

residencial de pequeno porte, em que os usuários locais agem como clientes, acessando servidores externos localizados na Internet, os roteadores fornecidos pelo provedor de Internet costumam já vir com algumas destas funções embutidas de fábrica. Em uma rede de grande porte, como as existentes em *campi* universitários, as funções de rede costumam ser implementadas em hardware dedicado e são configuradas de modo a formar um encadeamento de funções de rede ou cadeias de funções (SFCs). Um exemplo de SFC aplicada a um fluxo de pacotes entrando em uma rede para acesso a um serviço web está ilustrado pelos dispositivos interconectados da Figura 2.3. As funções de rede realizadas pelos dispositivos na figura e os seus objetivos são:

1. **Filtragem de pacotes:** para verificar se as características do fluxo (endereço IP de origem, endereço IP de destino, porta origem, porta destino, protocolos etc.) coincidem com alguma regra que represente um potencial ataque. Caso represente, os pacotes são descartados. Caso não represente, eles seguem para a próxima função de rede;

2. **Tradução de endereços:** para repassar os pacotes da requisição para uma rede interna com endereços IP não roteáveis na Internet, onde o serviço *web* será fornecido por um conjunto de servidores;

3. **Balanceamento de carga:** para distribuir as requisições entre o conjunto de servidores *web*, visando a manutenção do tempo de resposta ideal e reduzindo a chance de sucesso de um ataque de negação de serviço.

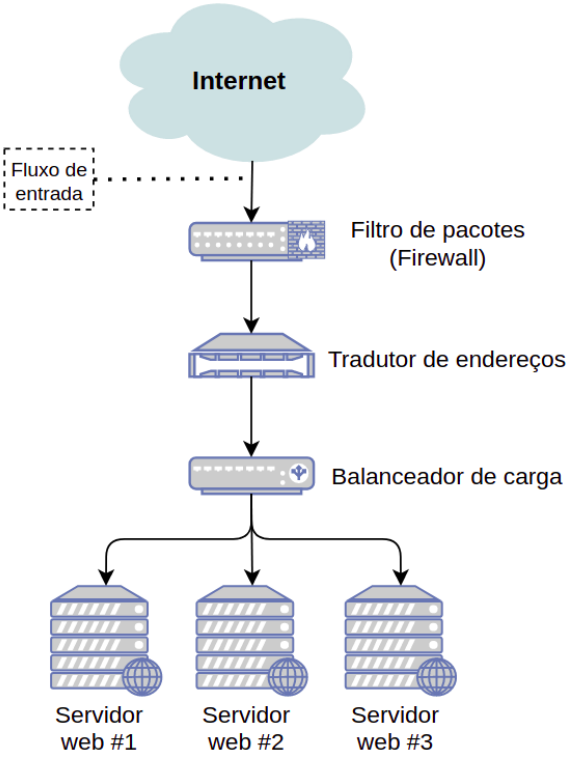


Figura 2.3: *Exemplo de um encadeamento de funções em uma rede tradicional.*

Uma rede como a ilustrada na Figura 2.3, em que as funções de rede são fornecidas por equipamentos de hardware dedicados conhecidos como *appliances* ou *middleboxes*, será

chamada no decorrer desta dissertação de uma rede tradicional. Nessa rede, se o operador tiver que adicionar uma nova função, como um sistema de detecção de intrusão (IDS), por exemplo, depois do *firewall* (filtro de pacotes), ele precisará adquirir o *appliance* específico e fazer sua instalação física e configuração na rede local, considerando possíveis mudanças em rotas nos *appliances* existentes.

A NFV permite a criação de uma arquitetura de redes que flexibiliza o processo de aquisição e reconfiguração de funções de rede por meio da virtualização destas funções. Diferentemente de uma rede tradicional, em uma rede baseada em NFV, as VNFs são fornecidas como *software* e instanciadas em hardware de propósito geral. Neste caso, o operador poderia contratar um serviço de IDS, que estaria implementado completamente em *software*, e instanciar o mesmo em um servidor localizado no *data center* de um provedor de nuvem ou em computadores na névoa ou na borda. Possíveis mudanças nas configurações de *appliances* existentes poderiam ser necessárias, assim como no caso da rede tradicional, mas todas as ações relacionadas à aquisição do equipamento específico e sua instalação física na rede local passam a ser desnecessárias. Neste caso da rede baseada em NFV, caso o operador resolva passar a utilizar outra implementação de IDS, bastaria alterar o *software* instanciado na nuvem/névoa/borda, sem necessidade de aquisição de um novo hardware. Além disso, em momentos de pico, em que o IDS precisasse realizar tarefas intensivas em memória e CPU, bastaria solicitar ao provedor que ampliasse esses recursos ou migrasse a VNF para um recurso com maior capacidade. Essas facilidades nas mudanças da implementação das VNFs e da capacidade das mesmas tornam a NFV ideal para ambientes heterogêneos, com constantes mudanças e com recursos limitados, como é o caso da IoT (Alam *et al.*, 2020).

A Figura 2.4 ilustra uma possível implementação da SFC da Figura 2.3 em uma arquitetura de NFV. Neste caso, o fluxo de pacotes alcança a primeira função de rede, o *firewall*, que está instanciado na nuvem, na névoa ou na borda. O fluxo percorre as demais funções, também instanciadas na nuvem/névoa/borda, até alcançar os servidores web. Vale notar que neste exemplo, o filtro de pacotes e o tradutor de endereços estão compartilhando uma mesma máquina física, mas essa configuração não precisa ser estática. Em um momento em que o tráfego na rede esteja relativamente alto, as instanciações podem ser feitas em máquinas separadas para que cada uma possa tirar proveito de todos os recursos físicos de cada um dos servidores.

O compartilhamento de um mesmo servidor físico seria interessante em momentos de tráfego baixo, permitindo que menos servidores permaneçam ligados, levando a uma economia em termos de energia elétrica. Além disso, também não é necessário que todas as funções estejam virtualizadas. Seria possível, por exemplo, instanciar apenas o *firewall*.

Para que a NFV funcione corretamente é necessário que haja a interação de diversos componentes, que podem ser agrupados em camadas, como apresentado na Figura 2.2. As VNFs precisam ser implementadas como *software* (contêineres ou máquinas virtuais, por exemplo) e mantidas em algum repositório, compondo a camada de serviços. A camada de serviços também pode fornecer SFCs inteiras.

Sempre que uma VNF é requisitada, ela deve ser instanciada em alguma máquina física. A infraestrutura da NFV (NFVi – *Network Functions Virtualization infrastructure*),

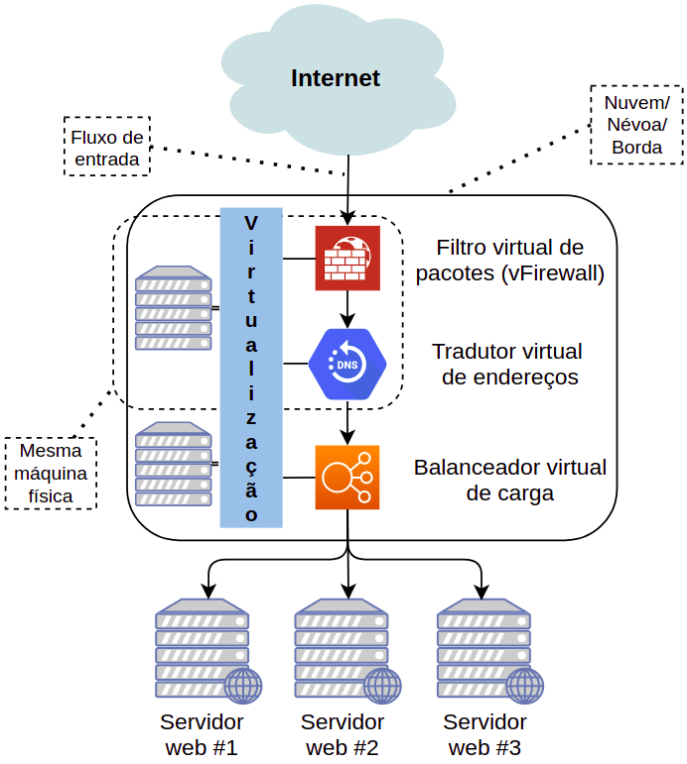


Figura 2.4: *Exemplo de um encadeamento de funções de rede com NFV.*

na camada de virtualização, é responsável por abstrair os recursos da infraestrutura física, que possivelmente estarão em um ou mais provedores, permitindo que equipamentos de diferentes fabricantes possam oferecer os recursos físicos necessários para a instanciação das VNFs. A NFVi também cuida da interconexão das VNFs entre si e com o mundo exterior, permitindo a interação com a Internet. Para garantir que os recursos da infraestrutura sejam alocados da melhor forma possível, é necessário gerenciar o ciclo de vida de cada uma das VNFs.

O gerenciador de VNF na camada de orquestração tem a responsabilidade de cuidar desse ciclo de vida, instanciando, atualizando e encerrando as VNFs. Ainda na camada de orquestração, o gerenciador de infraestrutura virtualizada tem o papel de manter um controle do mapeamento dos recursos físicos para os recursos virtuais.

2.2 Redes IoT e Aplicações

A IoT é um paradigma de comunicação com o objetivo de conectar diversos tipos de objetos à Internet (Meneghello *et al.*, 2019). Dentre esses objetos há, por exemplo, relógios, veículos, prédios, sensores e atuadores de um modo geral. Adicionar suporte à IoT em um objeto consiste na inclusão de recursos como bateria, memória, processador e interface de rede sem fio. Estes recursos são utilizados para execução de protocolos e aplicações da IoT. Com esse suporte, é comum que os objetos passem a ser chamados de "inteligentes". Este adjetivo também é usado para descrever aplicações executadas nos objetos. No cenário da Figura 2.2, os objetos inteligentes estão localizados na camada

física.

A IoT não é uma rede à parte da Internet, mas sim uma extensão da mesma que permite a conexão de diversos tipos de objetos do cotidiano. Diferente de um computador convencional, seja ele um computador pessoal ou um servidor dedicado, é comum que um dispositivo IoT seja exposto a ambientes críticos para equipamentos eletroeletrônicos, com altas temperaturas e umidade, por exemplo. A depender dos ambientes e das aplicações que serão executadas nos objetos, eles têm restrições nas dimensões físicas. As limitações impostas pelo ambiente e pela dimensão física impedem que dispositivos IoT tenham um alto poder computacional, exigindo que seja dada preferência pela execução de protocolos e aplicações leves, que garantam a qualidade de experiência dos usuários sem que haja um consumo elevado dos recursos do dispositivo, principalmente da bateria.

A Figura 2.5 ilustra a típica arquitetura em três camadas da IoT (Lin *et al.*, 2017). A camada de percepção está na parte inferior da arquitetura. Ela é responsável por interagir com os dispositivos e componentes físicos dos objetos inteligentes, obtendo dados e enviando comandos. A camada de rede integra diversos dispositivos de interconexão, como *switches* e pontos de acesso, a várias tecnologias de comunicação, como Bluetooth e Wi-Fi. Nessa camada são determinadas as rotas das comunicações. A camada de aplicação é a camada no topo da arquitetura e é onde as aplicações são propriamente implementadas. Aplicativos de cidades inteligentes, indústrias inteligentes e casas inteligentes estão nesta camada. O ideal é que esses aplicativos utilizem protocolos leves da camada de aplicação da Arquitetura Internet, como o MQTT, o CoAP e o AMQP (Naik, 2017).

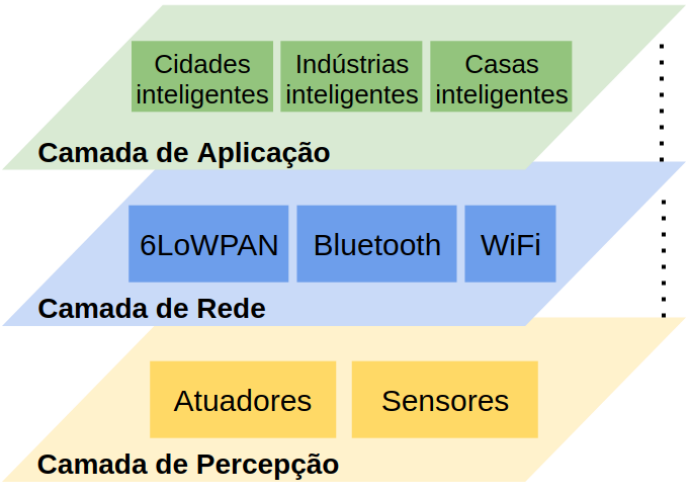


Figura 2.5: *Arquitetura em camadas da IoT.*

A limitação de recursos dos dispositivos conectados à IoT, unida às configurações padrão dos dispositivos, que não levam em conta alguns aspectos básicos de segurança como criptografia e a obrigatoriedade de senhas fortes (Giaretta *et al.*, 2019), têm tornado a IoT alvo de ataques de segurança. Parte desses ataques é realizada para obter o controle remoto dos dispositivos, que passam a fazer parte das chamadas *botnets*, redes criadas para disparar ataques de larga escala contra a Internet (Schwengber *et al.*, 2020).

2.3 Gerenciamento de Redes com NFV

A ausência de dispositivos com recursos abundantes para a execução de funções de rede relacionadas à segurança tornam a arquitetura de NFV interessante para prover segurança na IoT. Nesse caso, as VNFs de segurança podem ser instanciadas em dispositivos externos, fornecidos por provedores de nuvem, de névoa ou na borda. Para que isso seja possível, os componentes da camada de orquestração (Figura 2.2) têm papel fundamental (Medhat *et al.*, 2017).

O gerenciador de infraestrutura virtualizada está presente em cada um dos domínios de infraestrutura virtual. No caso de uma infraestrutura mantida pela plataforma OpenStack[1], por exemplo, o gerenciador utilizado é o `neutron`. Um gerenciador de infraestrutura virtualizada como o `neutron` é responsável por cuidar do endereçamento entre os dispositivos virtualizados, da topologia virtual, e da potencial conexão entre diversas infraestruturas por meio de redes privadas virtuais (VPNs - *Virtual Private Networks*), entregando "rede como um serviço".

O gerenciador de VNF é responsável por gerenciar o ciclo de vida das VNFs, como instanciação, atualização, mudança de configuração, escala vertical (mudança na quantidade de recursos físicos disponibilizados por um computador hospedeiro para uma função de rede), escala horizontal (mudança na quantidade de computadores hospedeiros usados para instanciar uma função de rede) e finalização.

2.4 Questões de Desempenho x Segurança com NFV na IoT

Um fator a ser considerado no uso de qualquer recurso computacional é o seu desempenho. Esse caracteriza-se por dois fatores: *métricas* e *medições*. As métricas são definições padrões de quantidades produzidas em um experimento, com uma utilidade pretendida. Elas devem ser cuidadosamente especificadas para transmitir o significado exato de um valor medido. A medição remete a um conjunto de operações com o objetivo de determinar o valor de uma métrica (IETF, 2011; IETF, 1998).

Dessa forma, os indicadores de desempenho computacional estão diretamente relacionados às métricas dos recursos que demonstram o grau de funcionalidade de um sistema computacional e, com isso, representam de forma abstrata o estado desse sistema. Por exemplo, o tempo necessário para o carregamento de um aplicativo baseado na *web* é o indicador mais perceptível de que o sistema está funcionando no nível esperado. Assim, tempos de carregamento mais longos do que o normal indicam que o sistema pode estar com problemas, exigindo atuações de seus administradores. Esses indicadores mudam ao longo do tempo e são fontes iniciais de informação sobre a "saúde" do sistema (Moghaddam *et al.*, 2019; Jain, 1991).

Do ponto de vista de NFV, a Tabela 2.1 apresenta métricas de desempenho relacionadas à

[1] OpenStack é uma plataforma de código aberto desenvolvida para gerenciar múltiplos recursos virtualizados. Fonte: https://www.openstack.org. Acessado em 15 de Julho de 2022.

velocidade, acurácia e confiabilidade de acordo com as categorias referentes à orquestração, operação de máquina virtual, estabelecimento e operação de redes e componentes de tecnologia como serviço. Essas métricas foram definidas pelo ETSI (ETSI, 2014a; ETSI, 2014b) para entidades que fornecem VNFs e que gerenciam infraestruturas virtualizadas, no intuito de garantir que os recursos oferecidos tenham o desempenho necessário de acordo com os requisitos de qualidade previstos.

Por exemplo, uma rede que está sendo altamente demandada por requisições benignas, ou seja, que não são ataques, e possui a latência de provisionamento de máquina virtual (VM – *Virtual Machine*) alta para o serviço virtualizado de *firewall*, poderá gerar grandes atrasos nas respostas às requisições, o que afeta diretamente a qualidade de experiência do usuário final do serviço.

Categoria da métrica	**Velocidade**	**Acurácia**	**Confiabilidade**
Primeira etapa da orquestração (por exemplo, alocação de recursos, configuração e instalação)	Latência de provisionamento de VM	Política de posicionamento de VM e conformidade	Confiabilidade de provisionamento de VM
Operação de VM	Interrupção da VM (duração e frequência do evento) e Latência de configuração da VM	Erro de *clock* da VM	Taxa de liberação prematura de VM
Estabelecimento de rede virtual (VN – *Virtual Network*)	Latência de provisionamento da VN	Conformidade de diversidade de VN	Confiabilidade de provisionamento de VN
Operação de rede virtual	Atraso de pacote, *jitter* (variação no atraso) e taxa de transferência de pacotes entregues	Taxa de perda de pacotes	Conexão interrompida
Segunda etapa da orquestração (por exemplo, liberação de recursos)	–	–	Taxa de liberação de VM com falha
Componente de tecnologia como serviço (TcaaS)	Latência do serviço TcaaS	–	Confiabilidade TcaaS (por exemplo, relação de transação defeituosa) e interrupção do TcaaS

Tabela 2.1: *Resumo das métricas de qualidade de serviço em NFV. Traduzido de ETSI (2014b).*

Na mesma linha do ETSI, a Força-Tarefa de Engenharia da Internet (IETF –*Internet Engineering Task Force*) (IETF, 2017a) designou um grupo específico, chamado Grupo de

Trabalho de Metodologia de *Benchmarking* (BMWG – *Benchmarking Methodology Working Group*), para produzir uma série de recomendações sobre as principais características e análise de desempenho de dispositivos, sistemas e serviços de rede. Assim, a Metodologia de *Benchmarking* para Desempenho de Virtualização de Redes foi desenvolvida, considerando como métricas de desempenho relacionadas ao recurso virtualizado, tais como o consumo de CPU e RAM, e relacionadas à rede, como latência, taxa de transferência e taxa de perda de quadros.

É possível encontrar, também, métricas relacionadas à NFV em outros trabalhos. Em Gupta *et al.* (2019), por exemplo, os autores abordam questões relacionadas à tolerância a falhas de NFV e apresentam métricas específicas para o problema, tais como: perda de pacotes, taxa de transferência média de pacotes, congestionamento de pontos de interconexão, entre outras. Em Kim *et al.* (2015), os autores apresentam métricas de comparação de desempenho para controladores SDN e NFV. As métricas apresentadas sobre NFV são concentradas em vCPU e vMemória (associadas à computação), latência e taxa de transferência (associadas à comunicação), taxa de E/S e tempo de recuperação de VNFs (associadas ao armazenamento). Em Zhang *et al.* (2016), os autores utilizam métricas como uso de CPU, taxa de transferência de pacotes e latência para apresentar como o efeito do acesso não uniforme a memória (NUMA – *Non-Uniform Memory Access*) e o posicionamento da cadeia de serviços impactam no desempenho de NFV.

Quando as características de dispositivos IoT são consideradas na implementação de uma rede, como baixo consumo de energia e recursos computacionais limitados, deve-se observar principalmente o comportamento da comunicação entre os dispositivos e os serviços disponibilizados a eles. Por exemplo, em um cenário de cidades inteligentes, um determinado dispositivo IoT acoplado a um semáforo deve solicitar a um servidor remoto a resposta de um serviço de cálculo, baseado nas informações de volume de automóveis coletadas, para que o semáforo tenha seus tempos de abertura e fechamento alterados de acordo com o tráfego de veículos. Se essa comunicação não levar em consideração métricas como latência entre o dispositivo IoT e o servidor, o tempo de resposta para a requisição fará com que o serviço prestado não tenha a qualidade pretendida em relação à otimização da gestão do tráfego de veículos.

Além disso, é necessário observar o comportamento dos serviços disponibilizados na rede IoT. Nesse contexto, a Figura 2.6 apresenta um cenário onde o serviço de IDS, provido por uma VNF, é consumido sob demanda de acordo com o volume de solicitações das câmeras, independentemente do *hardware* que o sustenta. Nesta figura, os valores $t_i (i \in \mathbb{N})$ representam diferentes instantes de tempo ordenados de forma crescente para mostrar que a demanda pelo serviço está aumentando. A Figura 2.7 ilustra não só o consumo do serviço virtualizado de *firewall* feito pelo dispositivo IoT A em movimento, mas também a migração do estado em que ele se encontra entre os diferentes provedores de VNF, nesse caso, Nuvem #1 e Nuvem #2.

Um exemplo prático desse cenário seria um sistema de transporte urbano de uma cidade inteligente, onde os ônibus poderiam ser representados pelos dispositivos IoT, e as nuvens pelos diferentes provedores de computação disponíveis em localidades geográficas distintas. Para os dois cenários ilustrados, a IETF (2017b) adiciona outras métricas ao desempenho de NFV, como tempo para implantar e migrar VNFs, que devem ser levadas em consideração

para que o tempo de resposta em uma rede IoT não seja afetado negativamente.

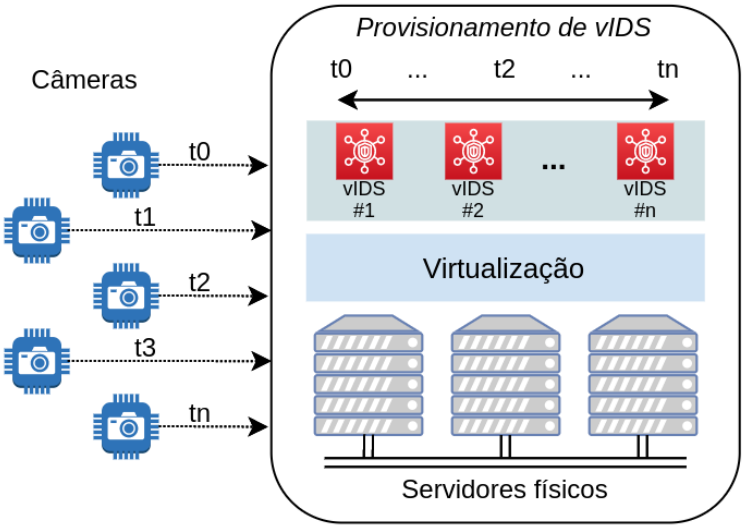


Figura 2.6: *Escalando recursos de VNFs de segurança em redes IoT. Adaptado de* Farris, Taleb *et al. (2019).*

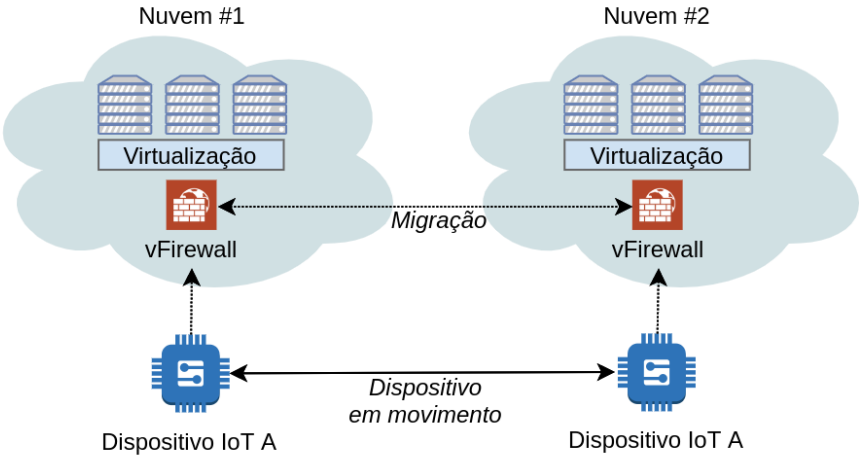


Figura 2.7: *Migrando recursos de VNFs de segurança em redes IoT. Adaptado de* Farris, Taleb *et al. (2019).*

A aplicação de NFV especificamente para segurança em redes IoT é uma abordagem que vem ganhando espaço atualmente (Alam *et al.*, 2020; Farris, Taleb *et al.*, 2019; Zarca, Bernabe, Farris *et al.*, 2018). Nesse caso, além das justificativas para as métricas apresentadas na Tabela 2.1, a própria funcionalidade de segurança fornecida pela VNF pode ter requisitos mais rígidos, principalmente em termos de tempo de resposta, para garantir que a detecção e a mitigação de um ataque sejam realizadas antes que seja tarde demais. Por exemplo, um IDS instalado na rede local onde os dispositivos de IoT estão conectados pode agir dentro do tempo esperado pelo fato de ele estar localizado no exato local onde os fluxos de interesse estão trafegando. Repassar esse tráfego para um ambiente computacional

remoto, mesmo que esse ambiente tenha alta capacidade de processamento, pode tornar o IDS inútil caso a latência entre a rede local e o ambiente remoto seja muito alta.

Esse é um caso em que não há diferentes gradientes para a qualidade do serviço, como no caso de um serviço que, mesmo se estiver "lento", ainda é útil. O não atendimento de um tempo mínimo, permitindo que o ataque tenha sucesso, significa que o serviço requisitado não foi entregue.

De um ponto de vista da camada de serviços, ilustrada na Figura 2.2, o trabalho de Farris, Taleb *et al.* (2019) apresenta abordagens para a aplicação de NFV voltadas à segurança de redes IoT. Isso possibilita a abrangência e a customização de serviços de proteção à rede oferecidos, como por exemplo, segurança como serviço (SECaaS – *Security as a Service*). Os autores classificam os trabalhos da literatura de acordo com quatro aspectos: separação do *software* de segurança do hardware; escalabilidade sob demanda e tolerância a falhas para VNF de segurança; suporte à mobilidade de VNF de segurança; e encadeamento de serviços de segurança de rede. Estendendo essa classificação proposta pelos autores, é possível considerar também o uso de VNF para implementação de modelos de ML no intuito de detectar e mitigar ataques na rede, denominando a proposta de VNF para modelos de aprendizado de segurança, Seção 2.5.

2.4.1 Separação do *Software* de Segurança do Hardware

Em Bremler-Barr *et al.* (2014), os autores apresentam uma abordagem de virtualização de serviços de DPI cujo desempenho, medido em taxa de transferência de pacotes, foi melhorado quando comparado a soluções implementadas em *middleboxes*. Em Montero, Yannuzzi *et al.* (2015), os autores propõem virtualizar as aplicações de segurança na borda da rede com base em domínio virtual de confiança (TVD – *Trusted Virtual Domain*), um contêiner lógico instanciado na rede composto por aplicativos de segurança do usuário e dados de controle de acesso a outras TVDs.

Em Yu *et al.* (2015), os autores propõem uma arquitetura de segurança chamada IoTSec, a qual contempla *micro-middleboxes* customizados (*µboxes*) que atuam como *gateways* de segurança para cada dispositivo IoT e cuja implementação pode se dar por meio de VNF. A IoTSec é composta por um controlador centralizado que monitora os contextos dos dispositivos e o ambiente operacional para gerar uma visão global e aplicar políticas entre os dispositivos. Com base nessa visualização, ele instancia e configura *µboxes* individuais e os mecanismos de encaminhamento necessários para rotear pacotes para elas.

2.4.2 Escalabilidade Sob Demanda e Tolerância a Falhas para VNF de Segurança

Em Cao *et al.* (2015), os autores apresentam uma solução chamada de NFV-VITAL. Seu objetivo é determinar a configuração que produz o maior desempenho de uma VNF, ou seja, ela computa a carga de trabalho máxima que uma VNF suporta antes que a qualidade do serviço se reduza, usando diferentes tamanhos de implantação e opções

de virtualização. A análise de soluções de IDS virtualizadas, como Snort[2] e Suricata[3], demonstrou os benefícios de selecionar o dimensionamento e as configurações ideais para os mecanismos de segurança. Em Hafeez *et al.* (2016), os autores abordam a manutenção de réplicas com reconhecimento de estado de *middleboxes* virtuais com o objetivo de, em caso de falhas, instanciar novos serviços.

2.4.3 Suporte à Mobilidade de VNF de Segurança

É possível citar uma estrutura de suporte à migração de instâncias virtuais de segurança próximas aos dispositivos do usuário final, como proposto por Montero e Serral-Gracià (2016). A abordagem aproveita o uso de tecnologias de virtualização e de SDN para gerenciar a migração de aplicativos de segurança na extremidade da rede, enquanto minimiza a interrupção das conexões em andamento. A solução é composta por quatro componentes: um contêiner virtual de segurança, cuja função é fornecer aplicativos como *firewall* ou alguma composição entre eles; um controlador de rede, que é responsável por configurar o direcionamento do tráfego de rede para os aplicativos de segurança; um migrador de recursos, que realiza a função de mover o estado de um aplicativo de segurança específico para o novo local do usuário; e um orquestrador que coordena a alocação dos recursos, a configuração de rede e a migração dos contêineres virtuais de segurança.

2.4.4 Encadeamento de Serviços de Segurança de Rede

Em Gember-Jacobson *et al.* (2014), os autores apresentam o OpenNF, uma solução que implementa um plano de controle dedicado para garantir o controle coordenado dos estados da VNF e dos estados de encaminhamento da rede. Em Qazi *et al.* (2013), os autores desenvolveram uma abordagem baseada na aplicação de políticas de SDN para simplificar o encaminhamento de tráfego entre *middleboxes*. A solução, chamada de SIMPLE (*Software-defIned Middlebox PoLicy Enforcement*), é composta pelo processamento de políticas nas quais os administradores de rede configuram suas lógicas de processamento, abstraindo informações sobre onde esse processamento ocorre ou como o tráfego precisa ser roteado. Assim, o SIMPLE traduz as políticas configuradas para a infraestrutura física considerando os dados da topologia e do tráfego da rede. Por fim, a solução também considera restrições de dispositivos como CPU, memória, aceleradores para diferentes *middleboxes* e quantidade de TCAM (*Ternary Content Addressable Memory*) disponível para instalar regras de encaminhamento nos *switches* SDN.

O trabalho de T. Wang *et al.* (2021) não se restringe a serviços de segurança. Nele os autores propõem uma abordagem baseada em aprendizado por reforço profundo (do inglês, *deep reinforcement learning*) para posicionar as cadeias de serviços de rede de forma adaptativa. O algoritmo extrai características da rede física em tempo de execução por meio de um grafo de rede convolucional (GCN – *Graph Convolutional Network*) e gera estratégias de posicionamento de serviços através de um modelo de sequência a sequência (Seq2Seq

[2] Snort é um sistema de prevenção de intrusão (IPS – *Intrusion Prevention System*) baseado em código aberto. Fonte: https://www.snort.org. Acessado em 15 de Julho de 2022.

[3] Suricata é um IDS e um IPS também baseado em código aberto. Fonte: https://suricata.io. Acessado em 15 de Julho de 2022.

– *Sequence-to-Sequence*), que aprende a tomar decisões de posicionamento de cadeias de serviços observando o desempenho correspondente de decisões anteriores.

2.5 VNF para Modelos de Aprendizado de Segurança

ML pode ser definido como o campo de estudo que possibilita os computadores aprenderem sem necessariamente serem programados para isso. Dentre os principais paradigmas de aprendizado de máquina é possível citar o supervisionado, o não supervisionado, o semi-supervisionado e o aprendizado por reforço (Boutaba *et al.*, 2018; Skiena, 2017). O aprendizado supervisionado necessita que os dados estejam rotulados, ou identificados, para que seus ajustes internos sejam realizados. Por exemplo, para que um computador classifique uma determinada imagem como um certo animal, ele precisa ser treinado com diversas imagens de animais já rotuladas, diferentemente do aprendizado não supervisionado. Este realiza os ajustes em seus parâmetros internos buscando certos padrões nos dados, aglomerando-os de acordo com características em comum. O aprendizado semi-supervisionado se baseia no supervisionado. No entanto, o treinamento é caracterizado por poucos exemplos de dados rotulados e um grande número de exemplos não rotulados. Já o aprendizado por reforço inclui um sistema de recompensa e punição para determinar se as ações tomadas pelos agentes são válidas ou não. Ou seja, os agentes, que são responsáveis por executar uma determinada tarefa, aprendem tanto com os erros quanto com os acertos.

Técnicas de ML têm sido aplicadas em vários domínios de problemas distintos. Redes de computadores é um deles. Predição e classificação de tráfego, roteamento, controle de congestionamento, alocação de recursos, tolerância a falhas, gestão de QoS e segurança são alguns dos campos em que ML pode ser aplicado com o intuito de identificar e explorar padrões ocultos em dados que descrevem agrupamentos, prevem resultados de eventos futuros para problemas de classificação e regressão, além de possibilitar a extração de regras e inferências (Akyildiz *et al.*, 2020; Boutaba *et al.*, 2018).

Dentre os trabalhos da literatura relacionados ao uso de VNF para modelos de aprendizado de segurança, é possível citar Bülbül e Fischer (2020), onde é proposta uma abordagem de mitigação de ataques DDoS. A abordagem utiliza SDN e NFV por meio de um algoritmo de ML baseado em generalização, sumarização e LMP (do inglês, *Longest Matching Prefix*), que derivam padrões para descrever ataques na rede. Após a identificação do ataque, o módulo de geração de padrões deriva uma regra OpenFlow[4] para os demais *switches* da rede, solicitando a filtragem do tráfego. Caso o ataque extrapole a rede local, as regras de filtros de tráfego também podem ser propagadas para roteadores externos à rede local.

Em Y. Jia *et al.* (2020), os autores têm como objetivo propor técnicas de defesa contra ataques DDoS de IoT e apresentam um esquema de defesa centrado na borda da rede,

[4] OpenFlow é um protocolo de comunicação que permite o acesso ao plano de encaminhamento de um *switch* ou roteador, possibilitando o direcionamento e até mesmo o descarte de pacotes da rede. Fonte: https://opennetworking.org. Acessado em 15 de Julho de 2022.

denominado FlowGuard. Esse esquema consiste na composição de dois fluxos, o de filtro e o de manipulação. O fluxo de filtro utiliza uma tabela que define as regras de filtragem de fluxo para diferentes tipos de ataque DDoS. Também é composto por um algoritmo que detecta anomalias no tráfego de dispositivos IoT que passa por servidores de borda. Uma vez identificada a anomalia na rede, o fluxo de manipulação é ativado e a classificação do ataque DDoS ocorre utilizando técnicas de ML, como memória de curto prazo longo (LSTM – *Long Short-Term Memory*) e rede neural convolucional (CNN – *Convolutional Neural Network*), que podem ser implementadas por meio de VNFs.

2.6 Indicadores e Gerenciamento de Desempenho

Pode-se dizer que métricas e medições são as bases para o processo de monitoramento de qualquer tipo de rede de computadores, sendo utilizadas para determinar o estado dos componentes e, também, para servir de insumos para o gerenciamento de desempenho. Em redes tradicionais, o desempenho da rede leva em conta questões como transferência de dados, o tempo de resposta da rede, a perda de pacotes, a porcentagem do uso de recursos, a utilização da conexão de dados, entre outras.

Contudo, quando um contexto de rede envolve dispositivos IoT juntamente com abordagens de segurança baseadas em NFV, outras questões devem ser incluídas para nortear a análise de seu comportamento e o impacto que elas trazem para o seu desempenho. Alguns exemplos são: no domínio de IoT, tipos de dispositivos e protocolos de comunicação utilizados por eles; em NFV, provisionamento, escalabilidade, conexões de rede e orquestração de VNFs; e no contexto de segurança, ferramentas, algoritmos, tempo de detecção e mitigação de ameaças.

Capítulo 3

Revisão da Literatura

O gerenciamento de desempenho parte do princípio de como utilizar os indicadores ou métricas coletadas da rede para provisionar de forma eficiente recursos necessários para a manutenção da qualidade dos serviços entregues aos seus usuários (Moghaddam *et al.*, 2019; Jain, 1991). Por isso, para que a operação de um recurso de NFV seja realizada de forma eficiente, os indicadores de desempenho devem ser bem definidos e aferidos a fim de demonstrarem o comportamento e a "saúde" do sistema.

Por serem recursos virtualizados, um fator a ser ponderado por meio da medição dos indicadores é a capacidade máxima da infraestrutura física em que estão hospedados, ou seja, o orquestrador de VNFs não deve permitir a hiper alocação de recursos em uma infraestrutura compartilhada, fazendo com que outros recursos disponibilizados virtualmente tenham seu desempenho afetado. Nessa mesma linha, o orquestrador também não deve manter alocados recursos que não têm utilidade em um determinado contexto que a rede contempla.

Em outras palavras, os indicadores de desempenho contribuem para a tomada de decisão no uso ou não de um determinado recurso para um propósito identificado. Por exemplo, um recurso virtualizado de DPI pode ser alocado a partir do momento em que a rede começa a receber requisições de origens que ela geralmente não recebe, fazendo com que os pacotes sejam inspecionados de forma profunda, com o objetivo de detectar códigos maliciosos. Se não for identificado nenhum tipo de ameaça, se o endereço que originou a análise encerrar sua comunicação ou, se em caso de ataque, o mesmo for mitigado, o recurso virtualizado de DPI deve ser desalocado, liberando recursos computacionais às demais necessidades da rede.

Assim, a revisão da literatura neste Capítulo analisa como as métricas e medições de desempenho, apresentadas na Seção 2.4, foram abordadas por diversos autores em soluções baseadas em NFV voltadas para detecção e mitigação de ameaças na IoT. Além disso, ao se tratar de dispositivos IoT, é preciso investigar a fundo suas características antes mesmo de propor qualquer tipo de infraestrutura ou mecanismos de segurança dedicados a eles (Lin *et al.*, 2017). Portanto, as seguintes características são levadas em consideração para a análise dos trabalhos da literatura:

- **Dispositivos** utilizados no experimento: a variedade de dispositivos contidos no

guarda-chuva de IoT é grande e tende a aumentar. Segundo a Corporação Internacional de Dados (IDC – *International Data Corporation*) (IDC, 2020), em 2025, 75% dos mais de 55 bilhões de dispositivos previstos em todo o mundo estarão conectados a uma plataforma IoT. Além disso, a corporação estima que os dados gerados a partir de dispositivos IoT conectados estejam na casa dos 73 ZB (73×10^{21} Bytes) até 2025, sendo a maioria proveniente de segurança e vigilância por vídeo, além de aplicações industriais. É possível dizer que grande parte dos dados gerados e transitados nessas redes envolve informações privativas de usuários, tais como imagens, documentos, localizações, preferências, entre outros. Por isso, identificar como esses dispositivos estão sendo utilizados auxilia no reconhecimento de possíveis limitações de segurança que podem ser supridas por recursos de NFV;

- **Protocolo** abordado: a aplicação de padrões da Internet já consolidados a dispositivos inteligentes - como o IP - pode simplificar a integração dos cenários previstos em IoT. No entanto, os protocolos de comunicação convencionais da Internet precisam ser modificados ou estendidos para oferecer suporte a necessidades específicas de aplicações IoT. Nessa linha, Sobin (2020) contempla em seu trabalho um estudo dos protocolos e tecnologias de comunicação dedicadas às camadas: física, contida pelo protocolo IEEE 802.12.4; camada MAC, por IEEE802.15.4E, ZigBee e TSMP (do inglês, *Time Synchronized Mesh Protocol*); camada de convergência, contida por 6LoWPAN; camada de transporte, por UDP (do inglês, *User Datagram Protocol*), IPv6, uIP (*micro* IP), ROLL (do inglês, *Routing Over Low power and Lossy networks*), RPL (do inglês, *IPv6 Routing Protocol for Low Power and Lossy Networks*) e DTLS (do inglês, *Datagram Transport Layer Security*); e camada de aplicação, por CoAP (do inglês, *Constrained Application Protocol*), MQTT (do inglês, *Message Queuing Telemetry Transport*), DDS (do inglês, *Data-Distribution Service*), AMQP (do inglês, *Advanced Message Queuing Protocol*), SMCP (do inglês, *Secure Mobile Crowdsensing Protocol*) e LLAP (do inglês, *Lightweight Local Automation Protocol*). Diferentemente de Sobin (2020), Lin *et al.* (2017) consideram que a camada de percepção é dedicada a tecnologias como RFID (do inglês, *Radio Frequency Identification*) e demais sensores de redes IoT. Os protocolos IEEE 802.15.4, 6LoWPAN, ZigBee, Z-Wave, MQTT, CoAP, DDS, AMQP e XMPP (do inglês, *Extensible Messaging and Presence Protocol*) são categorizados na camada de rede, pois a camada de aplicação refere-se ao contexto de negócio do uso de IoT, como cidades e transportes inteligentes. Outros trabalhos (Meneghello *et al.*, 2019; Neshenko *et al.*, 2019; Deogirikar e Vidhate, 2017; Y. Yang *et al.*, 2017; Y. Gao *et al.*, 2013) também fazem estudos de camadas e protocolos de comunicação a fim de direcionar soluções, desafios e limitações no contexto de IoT. Portanto, identificar e relacionar os protocolos de comunicação ajuda a compreender de que forma a NFV tem contribuído com esses desafios. Baseado nisso, a Tabela 3.1 traz um consenso entre as camadas e os protocolos de comunicação IoT abordados nos trabalhos analisados;

- **Arquitetura** apresentada: existem diversas vantagens em considerar a utilização da computação em nuvem para aplicações tradicionais, como a virtualização de recursos, escalabilidade, redundância, pagamento de acordo com o uso (do inglês, *pay-as-you-use*), entre outros. No entanto, para aplicações IoT nas quais os dispositivos possuem capacidades limitadas, o processamento de determinadas tarefas deve ser dividido

Camada	Protocolo
Aplicação	HTTP(S), MQTT, CoAP, AMQP, LLAP, SMCP, DDS, XMPP
Rede	TCP, UDP, UIP, ROLL, RPL, DTLS, Zigbee, Z-Wave, IPV6, 6LoWPAN, LoRaWAN, BLE
Percepção	IEEE 802.15.4, IEEE 802.15.4E, TSMP

Tabela 3.1: *Protocolos referentes às camadas da IoT.*

e realizado entre os demais nós da rede em diferentes localidades, e quando este cenário abrange milhares de dispositivos interconectados, é preciso pensar em outras estratégias de infraestrutura para que o desempenho das aplicações não seja afetado negativamente. Por essa razão, outros paradigmas, como a computação em névoa ou *fog computing*, surgiram para suportar tal realização (Yousefpour *et al.*, 2019; Lin *et al.*, 2017; Singh *et al.*, 2016). Diferentes abordagens podem ser encontradas no intuito de aproximar o processamento computacional dos dispositivos IoT e reduzir a latência entre o provisionamento de serviços. Por exemplo, a Figura 3.1 ilustra a quantificação dos atributos quando próximos ou distantes da nuvem, além de diferentes níveis de processamento, partindo do nível mais capacitado (nuvem) para o menos capacitado (borda), contemplando um nível intermediário (névoa). Dessa forma, entender as características de infraestrutura consideradas pelos autores dos trabalhos analisados ajuda a entender como elas afetam diretamente o desempenho do uso de recursos NFV.

É fato que dispositivos IoT possuem características específicas e são desenvolvidos para atuar em cenários bem definidos, ou seja, não possuem um propósito geral de atuação, como computadores pessoais. Por exemplo, equipamentos hospitalares são desenvolvidos visando alcançar a máxima redução de tempo entre a coleta dos sinais vitais de uma pessoa e sua exibição em um sistema de monitoramento. Em um contexto industrial, sensores são utilizados para determinar a abertura ou fechamento de comportas de acordo com o volume de determinado líquido. Na gestão de tráfego urbano, dispositivos inteligentes podem coletar informações referentes ao número de carros nas vias e determinar o tempo de fechamento de um ou mais semáforos. Os exemplos são extensos.

No entanto, baixo consumo de energia, protocolos de comunicação otimizados e suporte à conexão entre dispositivos heterogêneos são alguns dos requisitos em comum que devem ser observados durante o desenvolvimento de mecanismos de segurança. Assim, do ponto de vista de segurança, a análise dos trabalhos da literatura apresentada neste Capítulo foi baseada nos seguintes itens:

- **Abordagem** da solução proposta: o propósito de um determinado mecanismo de segurança deve ser definido logo na etapa de planejamento de um projeto, sempre levando em conta os fundamentos de confidencialidade, integridade e disponibilidade. Quando em pleno funcionamento, o mecanismo deve objetivar duas principais ações: a detecção e a mitigação de ameaças à rede. A ação de detectar ameaças pode ser considerada uma etapa preliminar do processo de resposta a incidentes de segurança. Do mesmo modo, a mitigação pode ser vista como uma etapa subsequente, cujo

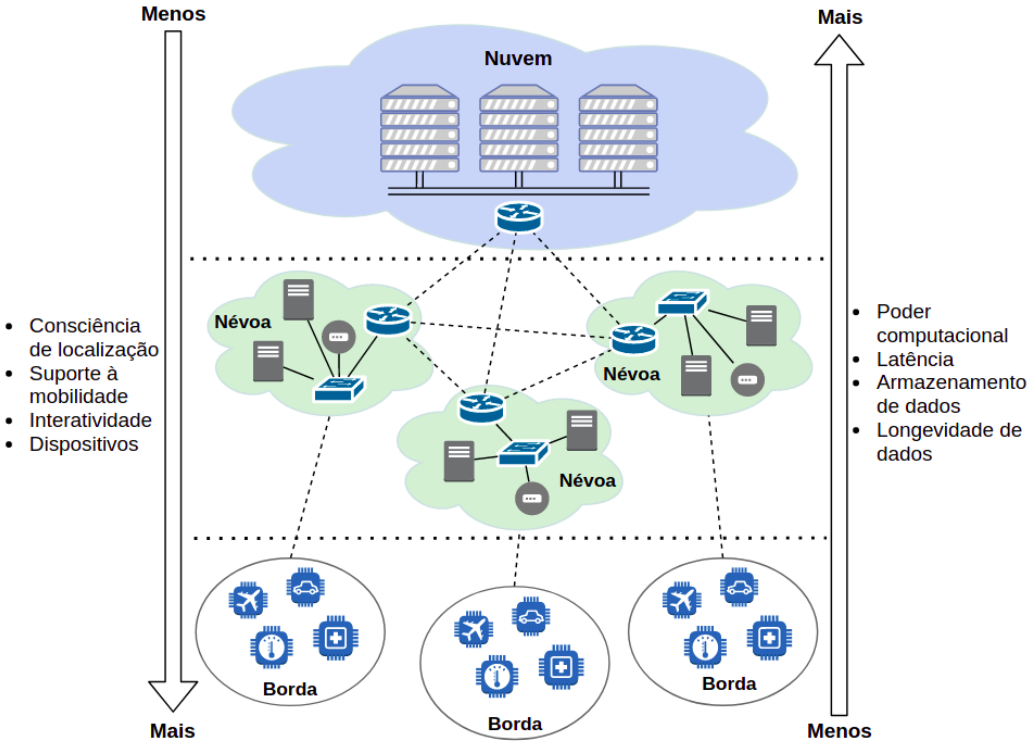


Figura 3.1: *Níveis de processamento e capacidades em diferentes arquiteturas distribuídas. Adaptado de* Yousefpour *et al.* (2019) *e* Lin *et al.* (2017).

objetivo é desabitar ou neutralizar a ameaça presente na rede. Por exemplo, uma vez identificado um ataque DDoS pelo mecanismo de detecção, o sistema de mitigação pode entrar em ação no intuito de negar ou redirecionar os pacotes maliciosos para o descarte. Entender e avaliar esses mecanismos ajuda a compreender em que nível de completude e maturidade do uso de NFV a solução proposta está;

- **Ataque** considerado: qualquer computador conectado via rede pode ser alvo de um ataque, da mesma forma que pode participar de um ataque, e isso não é diferente em redes IoT. Neste contexto, o trabalho realizado por Meneghello *et al.* (2019) apresenta vulnerabilidades em aberto de tecnologias como Zigbee, BLE (do inglês, *Bluetooth Low Energy*), 6LoWPAN e LoRaWAN da camada de rede. Y. Yang *et al.* (2017) pontua as principais limitações de dispositivos IoT, além de apresentar e analisar questões de segurança classificadas pelas camadas de percepção, rede, transporte e aplicação. No trabalho de Deogirikar e Vidhate (2017), além de serem considerados os ataques a redes IoT em camadas física, de rede e de *software*, são pontuados ataques específicos de encriptação, tais como: *side-channel*, *cryptanalysis of a simple modular exponentiation*, *ciphertext only*, *known plaintext*, *chosen plaintext*, *chosen ciphertext* e homem no meio (MITM – *Man-in-the-Middle*). Neshenko *et al.* (2019) vai além da classificação feita pelas camadas baseadas em dispositivo, rede e *software*, considerando o impacto que os ataques têm em relação à confidencialidade, integridade, disponibilidade e responsabilização. Para isso, os autores consideram que para a camada baseada em dispositivo, as vulnerabilidades identificadas são referentes à deficiência física ou captação de energia insuficiente. As vulnerabilidades

de autenticação inadequada, encriptação inapropriada e portas desnecessariamente abertas são citadas para a camada baseada em rede e, para a camada de *software*, são considerados itens como controle de acesso insuficiente, gerenciamento impróprio de atualizações de correção, práticas inseguras de desenvolvimento de código e mecanismos insuficientes de autenticação. A Tabela 3.2 apresenta uma descrição dos diferentes tipos de ataques em redes IoT e auxilia na compreensão das ameaças que têm sido detectadas e mitigadas pelo uso de NFV.

Ataque	**Descrição**
Vírus, *worm*, cavalo de tróia, *spyware* e *adware*	O invasor pode alterar o comportamento do sistema, roubar dados ou invalidar acessos usando esses códigos maliciosos.
Script malicioso	Ao injetar um *script* malicioso, o invasor obtém acessos privilegiados ao sistema.
Phishing	O invasor obtém as informações confidenciais como nome de usuário e senhas através da divulgação de *links* via e-mail, redirecionando a vítima para páginas falsas.
Negação de serviço (DoS) ou negação de serviço distribuída (DDoS)	O invasor faz com que o sistema alvo pare de fornecer seus serviços, ou seja, negando seu acesso aos demais usuários.
Sinkhole	O invasor compromete um nó dentro da rede e executa o ataque a partir desse nó. O nó comprometido envia informações de roteamento falsas para os nós vizinhos de que tem o caminho mínimo e, em seguida, atrai o tráfego para ele.
Roteamento	O invasor modifica e envia informações de roteamento anormais. Isso pode resultar no descarte de pacotes, encaminhamento de dados errados ou particionamento da rede.
Homem no meio (MITM)	O invasor intercepta a comunicação entre dois nós da rede e obtém acesso aos pacotes trocados por eles.
Sybil	O nó malicioso assume ilegalmente várias identidades e age de acordo com elas.
Captura de tráfego	O invasor intercepta e examina as mensagens para obter informações da rede.
Esgotamento de bateria	O invasor consome mais energia do que o previsto e, consequentemente, desativa o dispositivo.
Interferência de rádio frequência (RF) em RFIDs	O invasor envia sinais com ruídos via rádio frequência a fim de clonar a etiqueta do dispositivo alvo para obter acesso não autorizado ou desabilitar a comunicação.

Tabela 3.2: *Tipos de ataques a redes IoT e suas descrições.*

A seguir, uma análise dos trabalhos presentes na literatura, que fazem parte do contexto descrito anteriormente, é apresentada. Além disso, é realizada uma discussão entre os trabalhos relacionados e as questões de pesquisa com o intuito de suportar a validação da hipótese trabalhada nesta dissertação.

3.1 Análise dos Trabalhos Relacionados

Farris, Bernabe *et al.* (2017) apresentam uma proposta de um *framework* cujo objetivo é gerenciar e orquestrar políticas de segurança em redes IoT, na qual a heterogeneidade de dispositivos físicos e virtuais prevalece. A arquitetura do *framework* é composta pelos planos de usuário, de orquestração, de aplicação de segurança e de gerenciamento.

O plano de usuário implementa e oferece uma interface para a definição de políticas de segurança, tais como autenticação e autorização, filtragem e encaminhamento e proteção de canal. Após as políticas de segurança serem definidas pelo usuário, o plano de orquestração as interpreta e as incorpora no processo de execução. Além disso, o plano de orquestração também inclui os módulos de monitoramento, reação, orquestração de segurança e ativadores, que são responsáveis, respectivamente, por coletar as informações em tempo real dos componentes da rede; interpretar as informações recebidas do módulo de monitoramento e selecionar as contramedidas de acordo com as políticas de segurança definidas; selecionar os ativadores de acordo com a contramedida; e aplicar as políticas de segurança. Os ativadores, neste caso, são os recursos provisionados via VNFs.

O plano de aplicação de segurança é contido pelos domínios de gerenciamento e controle, infraestrutura e virtualização, VNF e IoT. O domínio de gerenciamento e controle supervisiona o uso dos recursos em tempo de execução, além de orquestrar as VNFs sobre a infraestrutura virtualizada seguindo as recomendações de gerenciamento e orquestração da ETSI. O domínio de infraestrutura e virtualização é responsável por disponibilizar, de forma virtualizada, os recursos de CPU, memória e elementos de redes. O domínio de VNF provê os recursos virtualizados e utilizados sob demanda para a rede, tais como *firewall*, IDS, DPI, entre outros. Por fim, o domínio de IoT inclui os dispositivos que fazem parte da rede a ser gerenciada.

Para demonstrar a aplicabilidade do *framework*, os autores descrevem dois estudos de caso e avaliam seu desempenho em uma prova de conceito em Zarca, Bernabe, Farris *et al.* (2018). O primeiro estudo de caso considera um cenário MEC (do inglês, *Multi-access Edge Computing*), no qual o principal objetivo do *framework* é suportar a capacidade de prover VNFs mais próximas dos dispositivos IoT, fazendo com que os requisitos de qualidade da rede sejam atendidos. O segundo estudo de caso apresenta um cenário relacionado a prédios inteligentes, BMS (do inglês, *Building Management System*), onde a proteção de dispositivos distintos, como computadores e aparelhos de ar condicionado, pode se dar por políticas de segurança que diferem os canais de comunicação.

A prova de conceito foi focada na aplicação de políticas para provisionar mecanismos de segurança em SDN. Os experimentos consideraram a comparação entre três abordagens. A primeira e a segunda foram implementadas pelos controladores SDN *OpenDaylight* (ODL) e *Open Network Operating System* (ONOS), respectivamente, e a terceira abordagem por um componente de VNF de *firewall* baseado em NETCONF (do inglês, *Network Configuration Protocol*) e iptables. Também consideraram um ambiente sob ataque do tipo MITM já detectado, utilizando como estratégia de mitigação o isolamento dos sensores comprometidos. Sobre os dispositivos IoT, foi utilizada uma máquina virtual executando um emulador Cooja Contiki, contido por sensores IoT conectados por meio do protocolo 6LoWPAN.

Os resultados foram aferidos de acordo com o tempo de reação necessário para cada abordagem implementada em um intervalo de 1 a 1000 aplicações de políticas de segurança para a mitigação do ataque. Em comparação com os controladores SDN ODL e ONOS, a implementação do *firewall* em VNF teve um desempenho inferior, principalmente na etapa de aplicação das políticas. Segundo os autores, o tipo de comunicação foi o principal motivo para essa perda de desempenho. Enquanto o mecanismo SDN estabelece inicialmente uma conexão TLS (do inglês, *Transport Layer Security*) e a mantém aberta durante a seção em que está ativo, o mecanismo NETCONF realiza a abertura de um canal SSH para cada solicitação.

A evolução do *framework* proposto inicialmente por Farris, Bernabe *et al.*, 2017 e avaliado por Zarca, Bernabe, Farris *et al.*, 2018 teve sua continuação em outros trabalhos. Por exemplo, em Zarca, Garcia-Carrillo *et al.*, 2019, os autores apresentam um abordagem para gerenciar políticas de autenticação, autorização e responsabilização (AAA – *Authentication, Authorization e Accounting*) na IoT através de uma VNF chamada de vAAA. Ela se estabelece em nível de névoa juntamente com políticas de segurança, que são executadas em um controlador SDN em nível de nuvem. Os autores também propõem realizar a comunicação entre dispositivos IoT e serviços da rede por meio de canais virtuais protegidos (vChannel-Protection), em específico, utilizando o DTLS. Para a experimentação, foram utilizados dispositivos IoT em uma versão personalizada do Contiki OS 2.7 e do servidor erbium CoAP, juntamente com o protocolo 6LoWPAN.

O desempenho dos componentes propostos foi avaliado de acordo com o processo de comunicação segura entre os dispositivos IoT e o *broker*[1], que consiste nas etapas de autenticação, autorização e canal protegido a partir das ações de refinamento, tradução e aplicação da política de segurança. Sob essa dimensão, os valores apresentados para as etapas de autenticação, autorização e canal protegido se mantiveram constantes entre 0 e 0,1 segundos para todas as ações, exceto para a de aplicação da política na etapa de canal protegido, que consumiu 0,35 segundos. Os autores também avaliaram o consumo de tempo para a aplicação de políticas, CPU e memória RAM de acordo com a escalabilidade de dispositivos IoT em um intervalo de 1 a 500. Os resultados mostraram que o tempo médio para as ações de refinamento, tradução e aplicação das políticas foi em média de 30 segundos para 500 dispositivos IoT na rede. O consumo de CPU tornou-se constante, próximo dos 100%, a partir de 100 dispositivos IoT; e o consumo de memória RAM se manteve estabilizado para as ações de refinamento e tradução de políticas, em torno de 40 MB; enquanto que para a ação de aplicação da política, o consumo chegou aos 120 MB.

Em Zarca, Bernabe, Trapero *et al.* (2019), os autores apresentam uma arquitetura referente ao gerenciamento de segurança para o *framework*. Nessa etapa, os autores desenvolveram um módulo de gerenciamento composto por um componente chamado de filtragem e pré-processamento de dados, que realiza a captura de informações da rede e as encaminha ao detector de incidentes. Este último executa a análise de segurança, procurando possíveis ataques na rede. O módulo também contém um banco de dados de assinaturas de ataques e um outro componente baseado em inteligência artificial, que aplica

[1] Um *broker* é um elemento de software usado em protocolos do tipo *publish-subscribe* que recebe as mensagens dos publicadores e as envia para os assinantes.

técnicas de ML para detectar anomalias de comportamento, componente esse desenvolvido por Bagaa *et al.* (2020).

Os experimentos realizados consideraram os dois cenários citados anteriormente, MEC e BMS, e os mesmos dispositivos de Zarca, Garcia-Carrillo *et al.* (2019). Além disso, foram realizados ataques DDoS no cenário MEC e a simulação de um dispositivo IoT infectado por um *malware* a fim de demonstrar a capacidade do módulo em detectar e mitigar a ameaça na rede. Os resultados apresentados na análise do desempenho foram relacionados à detecção e mitigação dos ataques. Para a detecção, a solução mostrou um tempo médio de 460 milissegundos em 100 testes realizados. Para a mitigação, ou seja, a aplicação de políticas de segurança na rede, o tempo médio entre tradução e aplicação foi de 406 milissegundos também para as 100 iterações. Portanto, a execução do ciclo completo de resposta a ataques de DDoS levou menos de um segundo.

Seguindo a mesma linha de evolução do *framework*, o trabalho de Zarca, Bernabe, Skarmeta *et al.* (2020) traz uma nova abordagem de mitigação de ataques à rede IoT por meio da implementação automatizada de *honeynets* baseadas em VNFs. A Figura 3.2 ilustra a criação de uma rede de dispositivos IoT virtualizados (vIoTHoneynet) sem nenhuma função para que, após a detecção de um ataque, o redirecionamento do tráfego seja feito para a vIoTHoneynet através de regras atualizadas no controlador SDN. Os experimentos foram baseados em dois cenários BMS e os experimentos foram executados 100 vezes para o mesmo caso de teste.

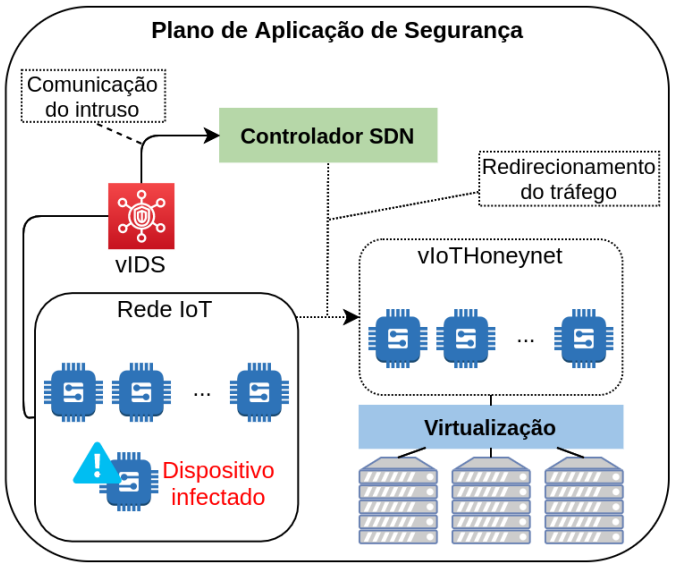


Figura 3.2: *Arquitetura vIoTHoneynet. Adaptado de Zarca, Bernabe, Skarmeta et al. (2020).*

Considerando que a vIoTHoneynet é instanciada sob demanda, ou seja, apenas quando um ataque é detectado na rede IoT, os experimentos consideraram dois tipos de tempos para compor o tempo total de mitigação. O primeiro tempo foi referente à configuração da vIoTHoneynet, composta pelos tempos de compilação e carregamento dos códigos dos novos dispositivos IoT, além do tempo de configuração da rede para esses dispositivos. O segundo tempo foi referente ao estabelecimento da vIoTHoneynet na rede, por exemplo, pela configuração do roteamento das conexões do atacante através da implementação de

novas regras no controlador SDN e pelo tempo de aplicação das políticas de segurança em si. O primeiro cenário de experimentação contou com 20 e o segundo com 50 dispositivos IoT remotamente acessíveis, distribuídos em sensores de umidade, temperatura, luz, CO2, RFIDs para a abertura de portas e roteadores RPL. Os cenários não se diferenciaram apenas pela quantidade de dispositivos, mas também pela versão do sistema operacional Contiki utilizado para a simulação dos dispositivos IoT, respectivamente, 2.7 e 3.1. Os autores também consideraram a implantação de diferentes tamanhos de vIoTHoneynets para cada cenário. Esses tamanhos de teste foram de no mínimo 10 dispositivos a no máximo 50.

Os resultados mostraram que para o primeiro cenário, o tempo médio para a configuração da vIoTHoneynets foi de 15 segundos. Para o segundo cenário, o tempo médio de configuração foi de 45 segundos. Quando os tempos de configuração da vIoTHoneynets foram somados ao tempo de aplicação das políticas de segurança para mitigar o ataque, o tempo médio de mitigação do ataque para o primeiro cenário ficou em torno de 51,5 segundos, e para o segundo cenário, 160 segundos. Os autores também aferiram o consumo de CPU e memória RAM das vIoTHoneynets criadas. Em ambos os cenários, o consumo de CPU se manteve em 100% para todos os tamanhos de teste, enquanto que o consumo de memória RAM no primeiro cenário cresceu de acordo com o aumento dos testes, mantendo-se no limite de 20% de uso, diferentemente do segundo cenário, que alcançou um consumo máximo de 25%.

O trabalho de Bagaa *et al.* (2020) propõe um modelo baseado em aprendizado supervisionado desenvolvido sobre o *framework* proposto por Farris, Bernabe *et al.* (2017). Nele, os autores implementam um agente de inteligência virtual que recebe as informações do módulo de monitoramento da rede e as analisa no intuito de detectar padrões referentes a ataques como DDoS, *Probing*, U2R (do inglês, *User to Root*) e R2L (do inglês, *Remote to Local*). Os modelos de aprendizado *Random Forest*, J48, *Bayesian Network* e *Hoeffding Tree* foram desenvolvidos e treinados a partir de conjuntos de dados que incluem os tipos de ataque citados anteriormente.

Os experimentos foram baseados na coleta de dados de sensores de temperatura e CO2 de salas distintas e subdivididos em quatro categorias: (SV), conjunto de dados contendo apenas os valores de medição e data; (P5V), conjunto de dados que inclui data, valor, valor precedente, valor 2º precedente e valor 5º precedente; (PD3V), que inclui data, valor, valor precedente diferente, 2º precedente diferente e 3º precedente diferente; e (CR), conjunto de valores cruzados entre as salas, sendo sua estrutura: data, sala 1, sala 2, sala 3, sala 4 e se são rotulados ou não. Os resultados apresentados mostraram que, para os dados de sensores de temperatura, o modelo desenvolvido teve uma acurácia de 99,71% para SV e P5V, de 99,28% para PD3V e de 99,23% para CR. Para o dados de sensores de CO2, o modelo mostrou uma acurácia de 98,86% para SV, de 99,24% para P5V e de 99,13% para PD3V.

No mesmo contexto referente à aplicação de ML de Bagaa *et al.* (2020), o trabalho de Sairam *et al.* (2019) apresenta uma solução chamada NETRA, que disponibiliza VNFs na borda da rede IoT no intuito de melhorar sua segurança. A solução foi desenvolvida para ser executada em dispositivos de baixa capacidade de processamento computacional. Por isso, utiliza uma estrutura baseada em contêineres Docker[2] para suportar as VNFs de

[2] Docker é uma plataforma que utiliza a virtualização para executar aplicações em pacotes chamados con-

segurança. A Figura 3.3 ilustra como os componentes da solução são distribuídos. A camada de núcleo da rede é composta pela conexão com o provedor de Internet (ISP – *Internet Service Provider*) e pelo Docker Hub[3], responsável por armazenar as imagens utilizadas em cada VNF. A camada de borda contém o *gateway* IoT desenvolvido em Raspberry Pi 3, onde se implementa as VNFs para aquele domínio de segurança. A camada de dispositivos IoT é composta pelos dispositivos inteligentes utilizados na rede. Todo o tráfego de rede realizado na borda é analisado pelo componente vAnalytics, que implementa um modelo de ML baseado em *Random Forest* para classificar os ataques a partir de anomalias identificadas. O modelo utilizado foi treinado a fim de detectar e mitigar ataques do tipo DDoS.

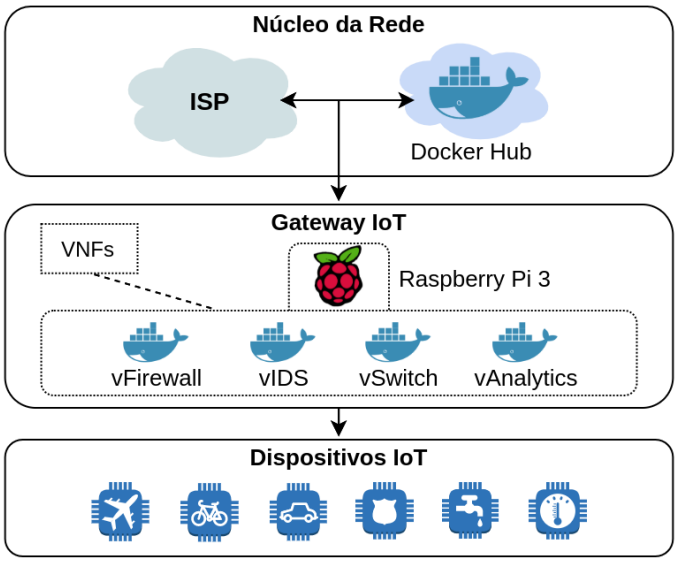


Figura 3.3: *Arquitetura NETRA. Adaptado de* Sairam *et al. (2019).*

Os experimentos foram realizados sob duas perspectivas de implementação, a de máquinas virtuais e a de contêineres, com o objetivo de avaliar os indicadores de desempenho: armazenamento, memória, latência, taxa de transferência e escalabilidade. Sobre o desempenho de armazenamento, as máquinas virtuais requisitaram 72 GB de espaço, enquanto que os contêineres já instanciados com as VNFs requisitaram 465 MB. Sobre o consumo de memória, as máquinas virtuais tiveram um pico de 140 MB, e os contêineres de 110 MB, para quatro instâncias em execução. Os valores apresentados de latência entre VNFs e entre dispositivos IoT e VNFs foram, em média, 0,83 milissegundos para a implementação baseada em máquinas virtuais e 0,53 milissegundos para a implementação em contêineres. Para a taxa de transferência, os resultados mostraram que os contêineres podem suportar até 16,8 Mbit/s para pacotes TCP e aproximadamente 1 Mbit/s para pacotes UDP. Sobre a escalabilidade, a perspectiva baseada em máquinas virtuais apresentou um tempo aproximado de 20 minutos para configurar uma VNF, e de cinco minutos para iniciá-la/pará-la. A perspectiva baseada em contêineres apresentou um tempo aproximado de quatro minutos para configurar uma VNF pela primeira vez, e a partir desse momento, um

têineres. Os contêineres são isolados uns dos outros e agrupam suas próprias aplicações, bibliotecas e arquivos de configuração. Fonte: https://www.docker.com. Acessado em 17 de Julho de 2022.

[3] Docker Hub é o principal repositório de imagens de aplicações disponibilizadas em contêineres. Fonte: https://hub.docker.com. Acessado em 17 de Julho de 2022.

tempo de nove segundos. Além disso, os experimentos mostram que o tempo para iniciar uma VNF em um contêiner é menor que um segundo. Os autores também apresentam uma avaliação do desempenho do modelo de aprendizado para a detecção e mitigação de ataques DDoS. Nesse caso, o modelo apresentou uma acurácia de 94,4% e um tempo médio de resposta da VNF para mitigação menor que um segundo.

O trabalho de Boudi *et al.* (2019) também apresenta uma abordagem baseada em contêineres para arquiteturas de computação de borda. Nele, os autores realizaram uma comparação de desempenho onde o Suricata foi executado em um dispositivo Raspberry Pi 3, que contempla dois cenários diferentes: o primeiro em máquina física (SoBM) e o segundo, em um contêiner Docker (SoDC). Os indicadores de desempenho utilizados na comparação foram: número de pacotes processados, taxa de transferência, consumo de memória RAM e CPU e número de pacotes descartados.

Os resultados mostraram que, em relação ao número de pacotes processados, taxa de transferência e consumo de memória RAM, os valores foram muito próximos, podendo considerar um empate entre as duas implementações. No entanto, os indicadores de consumo de CPU e número de pacotes descartados apontaram que a abordagem SoDC possui uma vantagem sobre SoBM, onde o consumo de CPU foi 10% menor e o número de pacotes descartados foi 4,5% menor.

Em Guizani e Ghafoor (2020), os autores utilizam das vantagens da escalabilidade providas pela NFV para implementar um IDS baseado em modelos de ML para mitigar *malwares* em redes IoT. A solução apresentada é composta por um modelo RNN-LSTM, executado em uma rede centralizada, que detecta os ataques na rede, e cria zonas de vigilância por meio de diferentes NFVs para realizar a aplicação de *patches* de segurança nos dispositivos.

Para demonstrar o desempenho da abordagem, os autores realizaram experimentos considerando o modelo epidêmico SEIR, usado na área de epidemiologia. Nesse modelo, há indivíduos suscetíveis (S), expostos (E), infectados (I) e resistentes (R). No caso de Guizani e Ghafoor (2020), os indivíduos são os dispositivos IoT. A partir da disseminação de um *malware* na rede IoT, o desempenho da abordagem foi aferido de acordo com a escalabilidade da rede, ou seja, a criação de zonas de vigilância em NFV e a quantidade de vírus na rede. Foram consideradas diferentes quantidades de dispositivos na rede, dentre 1000, 10.000 e 100.000. Os resultados mostraram que a criação de zonas de vigilância virtualizadas se mantém constante para 1000 dispositivos na rede. Nesse caso cinco zonas foram criadas. Já o aumento do número de zonas de vigilância, para 50, iniciou-se a partir de 10.000 dispositivos na rede. No entanto, se manteve constante até os 100.000.

Levando em consideração a computação em névoa, Zhou *et al.* (2019) propõem a implementação de VNFs para a realização da detecção de ataques DDoS em redes IIoT. A abordagem de detecção de anomalias proposta pelos autores considera o uso do Snort para implementar máquinas de estado modeladas a partir de protocolos de comunicação TCP e proprietários utilizados em sistemas de controle industrial, nesse caso, o *Modbus*.

Os resultados apresentados mostraram que em uma arquitetura que considera o processamento do sistema de detecção próximo aos dispositivos IoT, em névoa, há um tempo médio de 137 milissegundos e uma taxa de detecção de 93,93%, enquanto que o tempo

médio de resposta em uma arquitetura remota é de aproximadamente 240 milissegundos e uma taxa de detecção de 84,95%.

A complexidade no gerenciamento de segurança de redes IoT também é uma questão a ser abordada quando há mais de um provedor de serviços na rede. As cidades inteligentes são alguns dos exemplos. Nelas, diversos provedores de computação em nuvem podem disponibilizar diferentes serviços aos vários dispositivos IoT conectados pela cidade. No entanto, a oferta desses serviços deve levar em consideração as características de redes IoT, como a utilização de *gateways* de borda e até mesmo domínios específicos para este tipo de conexão. Por isso, Massonet *et al.* (2017) apresentam uma extensão de uma arquitetura de segurança em nuvem federada para redes IoT, baseada na aplicação de políticas de segurança. A segurança em uma arquitetura de nuvem federada é composta por um componente gerenciador central que realiza a gestão das políticas de segurança da rede e as disponibiliza por meio dos demais gerenciadores de nuvens, que são federadas. A proposta apresentada pelos autores considera que a implementação das políticas de segurança de rede, feita por cada provedor de nuvem, seja realizada por meio das cadeias de VNF de segurança e pelo roteamento do tráfego via regras no controlador SDN. No contexto de IoT, a arquitetura proposta se estende para a utilização de *gateways* de borda, onde se realiza a conexão direta com os dispositivos IoT. Além disso, a proposta também considera que o processamento das VNFs de segurança deve ser realizado na borda, devido ao baixo poder computacional e ao possível aumento da latência devido à distância entre o provedor de serviço e os dispositivos. Os autores não avaliaram o desempenho dos recursos propostos.

Em um contexto de soluções domésticas inteligentes, Al-Shaboti *et al.* (2018) apresentam uma proposta da utilização de controladores SDN para reforçar o controle de acesso estático e dinâmico à rede IoT. Além disso, apresentam a implementação de uma VNF para mitigar ataques de falsificação ARP (do inglês, *Address Resolution Protocol*). A implementação do servidor ARP foi realizada de duas formas: uma utilizando a biblioteca Scapy[4] (SC), e a outra utilizando o *Data Plane Development Kit* (DPDK). Esse servidor tem como objetivo ser uma entidade confiável que realiza a gestão das requisições ARP na rede. Desta forma, todos os pacotes ARP na rede são filtrados pelo controlador SDN e são encaminhados ao servidor para resolução.

Para demonstrar a abordagem de mitigação, os experimentos foram executados utilizando o Mininet[5], de acordo com diferentes quantidades de máquinas presentes na rede, começando em dez e finalizando em 50. Os resultados mostraram que o tempo de resposta da VNF implementada em SC foi de mais de 70 milissegundos para dez dispositivos, e de aproximadamente um segundo para 50 dispositivos. Para a implementação em DPDK, o tempo de resposta médio foi de 0,57 milissegundo. Em relação à mitigação do ataque de falsificação ARP, os autores apresentam apenas a taxa de 100% de descarte para os pacotes maliciosos, sem contabilizar o tempo de execução do descarte.

A descrição de um dispositivo IoT e seu comportamento na rede podem ser utilizados

[4] Scapy é uma biblioteca implementada em Python específica para a criação e manipulação de pacotes de redes. Fonte: https://scapy.net. Acessado em 17 de Julho de 2022.

[5] Mininet é um emulador de redes que possibilita a criação virtual de máquinas, *switches*, controladores e conexões. Fonte: http://mininet.org. Acessado em 17 de Julho de 2022.

para a criação de melhorias em mecanismos de segurança. Dessa forma, a IETF (2020) orienta os fabricantes de dispositivos IoT e de sistemas de comunicação sobre a implementação da MUD (do inglês, *Manufacturer Usage Description*). Armazenadas em arquivos, as MUDs consistem em listas de permissões que descrevem domínios e *endpoints* para cada tipo de dispositivo na rede. Assim, logo ao conectar, os dispositivos IoT solicitam a MUD ao servidor dedicado a esse propósito para, então, se integrar com os demais componentes e serviços. Nesse sentido, o trabalho de Afek *et al.* (2020) apresenta uma proposta de utilização de VNFs em nível de ISP para a aplicação de políticas de segurança em redes domésticas em larga escala baseadas em MUD. O objetivo da proposta é garantir que os pacotes que trafegam na rede de ou para um dispositivo IoT sejam analisados e confrontados com suas regras contidas no arquivo MUD. Este mecanismo consiste em dois componentes, o de monitoramento (WLM – *whitelist monitoring*) e o de aplicação (WLE – *whitelist/MUD enforcement*). O componente WLM é responsável por analisar se os pacotes estão ou não de acordo com as regras. O componente WLE recebe as informações da WLM e aplica as políticas de exclusão ou de manutenção da conexão. A proposta dos autores é de que o WLM seja desenvolvido como uma VNF, sendo solicitada pelo WLE que está no caminho principal dos pacotes ou roteadores. Nesse trabalho, os autores apresentam uma prova de conceito. Porém, não analisaram o desempenho dos componentes nem do mecanismo implementados.

Como descrito anteriormente, a NFV permite a criação de uma arquitetura de redes que flexibiliza o processo de aquisição e reconfiguração de funções de rede por meio da virtualização destas funções. Essa flexibilização também permite que VNFs possam ser instanciadas em infraestruturas distintas de acordo com a demanda requisitada. Isso significa que, além dos próprios recursos virtualizados, é necessário que mecanismos sejam implementados para tomarem decisões sobre o posicionamento de VNFs. Por exemplo, uma VNF que implementa um recurso de segurança pode ser instanciada tanto próxima ao dispositivo atacado quanto ao dispositivo atacante para mitigar um ataque na rede. Para tomarem essa decisão, mecanismos de posicionamento precisam ser alimentados com informações sobre o estado atual da rede, ou até mesmo dos dispositivos que nela estão. Nesse sentido, Liu *et al.* (2021) apresentam uma abordagem de posicionamento de VNFs e roteamento de SFC em redes SDN. Os autores desenvolveram um mecanismo dinâmico de posicionamento em programação linear inteira (*ILP – Integer Linear Programming*) que utiliza como variáveis de decisão a capacidade de tabelas de encaminhamento de *switches*, a latência e a taxa de transferência de conexões entre dispositivos, a taxa de pacotes descartados, o *jitter* e o consumo de CPU da VNF.

Para avaliar a proposta, os autores utilizaram a biblioteca Python NetworkX 2.4 para gerar uma topologia de rede composta por 60 nós e 79 conexões entre eles. Os requisitos de CPU das VNFs consideradas foram de 800 a 3200 MIPS (Milhões de Instruções por Segundo). A largura de banda de cada conexão variou de 1 a 10 Gbps, a latência foi entre 2 e 5 milissegundos, a taxa de perda de pacotes esteve entre 0 e 0,0005 e o *jitter* entre 0,5 e 2 milissegundos. Valores inseridos na topologia aleatoriamente.

Nos experimentos, foram aferidos a taxa média de aceitação de pacotes, a taxa de transferência, a contagem de saltos e o tempo de execução do modelo. Em relação a estes indicadores, o modelo demonstra ser uma alternativa viável quando comparado aos demais apresentados no trabalho. Em relação ao tempo de execução, com o aumento do número

de nós da rede (intervalo entre 40 e 80), o tempo de execução do modelo proposto em ILP aumenta exponencialmente, cerca de 90 segundos para 70 nós e de 250 segundos para 80 nós, enquanto que os demais modelos utilizados na comparação aumentam em ordem quadrada e cúbica.

Em Varasteh *et al.* (2021), os autores também apresentam uma formulação em ILP para determinar o número ideal de VNFs e seu mapeamento para implantação em máquinas físicas. Têm como foco a otimização do consumo de energia, tanto da rede, composta pela energia consumida pelos *switches* e por suas conexões ativas, quanto de máquinas físicas, composta pelo consumo de CPU. Também, consideram como variáveis de decisão a taxa de transferência entre dispositivos na rede e o tempo de processamento de requisições à uma determinada VNF.

Cada nó de rede foi equipado com um *switch* e uma máquina física que hospeda as VNFs. As máquinas físicas foram caracterizadas pelo número de núcleos da CPU, definido como 16. O consumo de energia da máquina ociosa e com máxima utilização foi de 299 e 521 *Watts*, respectivamente. Por outro lado, o consumo de energia do *switch* consistiu em uma medida estática e outra de interface de rede. A medida estática foi configurada para 315 *Watts* e o consumo de energia da interface variou de acordo com a configuração da taxa de transferência dos dados: 100 Mbps, 1 Gbps ou 10 Gbps com 26 *Watts*, 30 *Watts* e 55 *Watts* de consumo de energia, respectivamente. A taxa de transferência das conexões foram definidas aleatoriamente entre 6 e 10 Gbps.

Os experimentos se basearam na comparação do modelo proposto pelos autores e outros dois pela literatura. Os autores avaliaram o consumo de energia das máquinas físicas e da rede, além da taxa de aceitação de pacotes de 500 requisições simuladas. Os resultados apresentados mostraram que o modelo proposto possui uma melhoria no consumo médio de energia e na taxa de aceitação de 24,7% e de 31%, respectivamente, em relação aos demais considerados para fim de comparação.

Pei *et al.* (2020) desenvolveram uma abordagem para o posicionamento de VNF, em redes SDN, de acordo com a carga dinâmica da rede. Os autores apresentam a formulação do problema em programação binária inteira (*BIP – Binary Integer Programming*) e utilizam um modelo de aprendizado por reforço profundo (*DRL – Deep Reinforcement Learning*), para resolve-lo de forma eficiente, driblando o tempo consumido por soluções da literatura no estabelecimento de SFC. Consideram como variáveis de decisão a taxa média de transferência da rede, memória RAM, número de processadores e consumo de CPU de nós da rede e de VNFs.

Os autores se basearam em uma topologia de 23 nós e 37 conexões entre eles para a realização dos experimentos. Toda a topologia foi dividida em 8 regiões de rede, cada uma com um nó representando uma função específica. A taxa de transferência das conexões e a memória RAM de cada nó foram de 1 Gbps e de 1 GB, respectivamente. O consumo de CPU de cada VNF foi definida como 0,05 MIPS. Os resultados mostraram que, quando comparados a algoritmos presentes na literatura, o modelo apresenta ganhos significativos em todos os critérios de comparação, desde a taxa de transferência, latência de ponta a ponta, tempo de execução das VNFs, taxa de utilização das VNFs até o balanceamento de carga entre as VNFs.

Quando requisitos de segurança e IoT são incluídos no contexto, é importante também considerar aspectos como a eficácia de uma determinada solução de detecção ou mitigação, seu tempo de resposta, entre outros. No entanto, estes indicadores de desempenho não são trabalhados nos contextos de Liu *et al.* (2021), Varasteh *et al.* (2021) e Pei *et al.* (2020).

3.2 Discussão Entre a Literatura e as Questões de Pesquisa

A Tabela 3.4 traz um panorama dos indicadores de desempenho para o contexto de segurança e a Figura 3.4 contabiliza a quantidade de trabalhos e as categorias das métricas utilizadas na avaliação de desempenho dos recursos implementados de acordo com a Tabela 3.3. Para o contexto de IoT, a Tabela 3.5 também apresenta um panorama dos indicadores de desempenho, porém juntamente com protocolos, arquiteturas e dispositivos. Elas auxiliam a responder as questões de pesquisa QP1 e QP2 da seguinte forma:

Categoria	Métrica
NFV	Consumo de CPU
	Consumo de memória RAM
	Armazenamento
	Escalabilidade
	Tempo de implantação de ambiente
Rede	Latência
	Taxa de transferência
	Número de pacotes processados/descartados
Solução	Tempo de resposta ao ataque
	Tempo de aplicação de políticas de segurança
	Tempo de detecção e mitigação de ataques
	Acurácia do modelo de aprendizado de máquina

Tabela 3.3: *Categorias e métricas de desempenho.*

1. QP1 (*Como os recursos NFV estão sendo utilizados para detectar e mitigar ataques na IoT?*): De acordo com a Figura 3.5, que contabiliza os ataques e as abordagens apresentadas nos trabalhos analisados, é possível constatar que o ataque mais utilizado para demonstrar o uso da NFV é o DDoS (55,55%), à frente do MITM (33,33%) e dos demais (33,33%). Além disso, também é possível observar que 46,15% dos trabalhos levam em consideração a NFV para a prevenção de ataques na rede IoT, 53,85% para detecção e 69,23% para mitigação. Para o contexto de IoT, 46,15% dos trabalhos abordam as características de computação em névoa para a aplicação de NFV. No entanto, apenas Zhou *et al.* (2019) apresentam uma comparação entre o desempenho de detecção e mitigação para arquiteturas distintas;

2. QP2 (*Quais são as métricas, ou indicadores, utilizados para aferir o desempenho do uso de NFV na detecção e mitigação de ataques em redes IoT e como eles impactam o gerenciamento de recursos virtualizados?*): As métricas relacionadas aos componen-

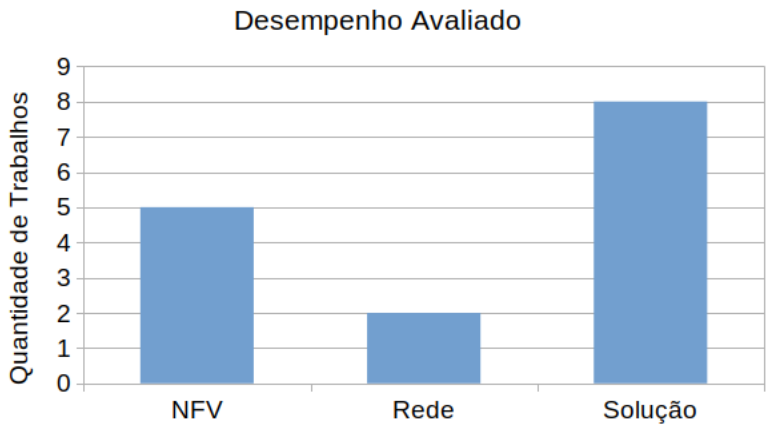
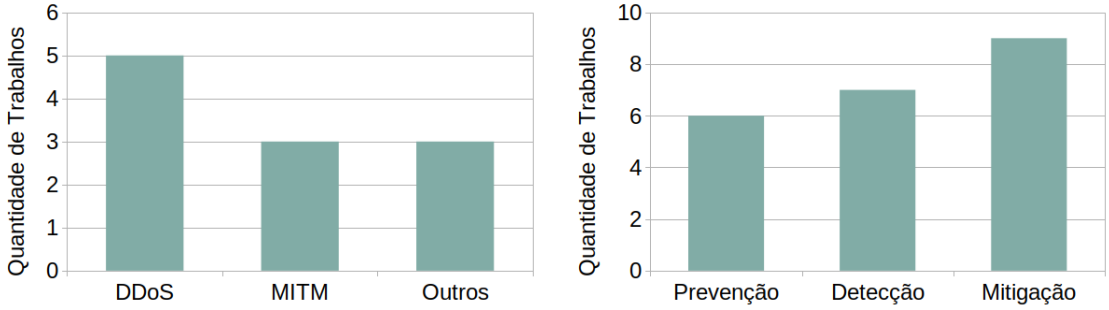


Figura 3.4: *Métricas utilizadas na avaliação de desempenho.*



(a) *Ataque.* **(b)** *Abordagem.*

Figura 3.5: *Tipos de ataques considerados e abordagens apresentadas.*

tes implementados por meio da virtualização (NFV) estão presentes em 50% dos trabalhos que avaliam o desempenho da solução proposta. Já as métricas relacionadas à rede (Rede) estão em 20% dos trabalhos. As que mais se destacam são as métricas que apresentam a eficácia da solução proposta (Solução) que estão em 80% dos trabalhos. Os indicadores de desempenho de NFV e de Rede são essenciais para compor soluções de segurança que visam aumentar o nível de qualidade de serviço (QoS – *Quality of Service*) da rede. No entanto, são menos explorados nas abordagens encontradas na literatura que os utilizados para demonstrar a eficácia da solução. Essa característica impacta negativamente o gerenciamento de recursos virtualizados, pois não basta que a abordagem apresente altos índices de resolução do problema. É necessário considerar, também, as variáveis do contexto do ambiente em que a solução é executada.

A conclusão da QP2 também pode ser encontrada na Tabela 3.6, que apresenta uma comparação dos trabalhos analisados em relação ao apresentado nesta dissertação. Nela é possível observar o diferencial que o projeto de mestrado visou alcançar, que é desenvolver uma abordagem para mitigar ataques DDoS na IoT, utilizando a composição dos indicadores de desempenho, tanto do próprio recurso NFV quanto da Rede e da Solução.

Trabalho	Desempenho avaliado	Segurança	
		Abordagem	Ataque
Farris, Bernabe *et al.* (2017)	—	Prevenção	—
Massonet *et al.* (2017)	—	Prevenção	—
Al-Shaboti *et al.* (2018)	Tempo de resposta ARP	Prevenção e mitigação	MITM
Zarca, Bernabe, Farris *et al.* (2018)	Tempo de aplicação de políticas de segurança	Prevenção e mitigação	MITM
Boudi *et al.* (2019)	Número de pacotes processados, número de pacotes descartados, taxa de transferência, consumo de CPU e de memória RAM	Detecção e mitigação	—
Sairam *et al.* (2019)	Armazenamento, consumo de memória RAM, latência, taxa de transferência, escalabilidade, tempo de implantação de ambiente e acurácia do modelo	Detecção e mitigação	DDoS
Zarca, Bernabe, Trapero *et al.* (2019)	Tempo de detecção e de aplicação de políticas de segurança	Detecção e mitigação	DDoS e *malware*
Zarca, Garcia-Carrillo *et al.* (2019)	Tempo de aplicação de políticas de segurança, consumo de CPU e de memória RAM	Prevenção e mitigação	MITM
Zhou *et al.* (2019)	Tempo de detecção	Detecção e mitigação	DDoS
Afek *et al.* (2020)	—	Prevenção	—
Bagaa *et al.* (2020)	Acurácia do modelo	Detecção	DDoS, Probing Attack, U2R e R2L
Guizani e Ghafoor (2020)	Escalabilidade	Detecção e mitigação	*Malware*
Zarca, Bernabe, Skarmeta *et al.* (2020)	Tempo de implantação de ambiente e de aplicação de políticas de segurança	Detecção e mitigação	DDoS

Tabela 3.4: *Panorama dos indicadores de desempenho para o contexto de segurança.*

Trabalho	Desempenho avaliado	IoT		
		Protocolo	Arquitetura	Dispositivo
Farris, Bernabe *et al.* (2017)	—	—	—	—
Massonet *et al.* (2017)	—	—	Névoa	Dispositivos em geral
Al-Shaboti *et al.* (2018)	Tempo de resposta ARP	—	—	Dispositivos em geral
Zarca, Bernabe, Farris *et al.* (2018)	Tempo de aplicação de políticas de segurança	6LoWPAN	—	Dispositivos em geral
Boudi *et al.* (2019)	Número de pacotes processados, número de pacotes descartados, taxa de transferência, consumo de CPU e de memória RAM	—	Névoa	Raspberry Pi 3
Sairam *et al.* (2019)	Armazenamento, consumo de memória RAM, latência, taxa de transferência, escalabilidade, tempo de implantação de ambiente e acurácia do modelo	—	Névoa	Raspberry Pi 3 e câmeras IP
Zarca, Bernabe, Trapero *et al.* (2019)	Tempo de detecção e de aplicação de políticas de segurança	CoAP e 6LoWPAN	Névoa	Dispositivos em geral
Zarca, Garcia-Carrillo *et al.* (2019)	Tempo de aplicação de políticas de segurança, consumo de CPU e de memória RAM	CoAP e 6LoWPAN	Névoa	Dispositivos em geral
Zhou *et al.* (2019)	Tempo de detecção	TCP e Modbus	Névoa	Dispositivos em geral
Afek *et al.* (2020)	—	—	—	Dispositivos que suportam MUD
Bagaa *et al.* (2020)	Acurácia do modelo	—	—	Sensores de temperatura e CO2
Guizani e Ghafoor (2020)	Escalabilidade	—	—	Dispositivos em geral
Zarca, Bernabe, Skarmeta *et al.* (2020)	Tempo de implantação de ambiente e de aplicação de políticas de segurança	CoAP, RPL e LoWPANs	—	Sensores de umidade, temperatura, luz, CO2 e RFID

Tabela 3.5: *Panorama dos indicadores de desempenho para o contexto de IoT.*

Trabalho	Indicadores de desempenho (Tabela 3.3)			Indicadores de desempenho como variáveis de decisão	Contexto	
	NFV	Rede	Solução		Segurança	IoT
Farris, Bernabe *et al.* (2017)	–	–	–	Não	Sim	Sim
Massonet *et al.* (2017)	–	–	–	Não	Sim	Sim
Al-Shaboti *et al.* (2018)	Não	Não	Sim	Não	Sim	Sim
Zarca, Bernabe, Farris *et al.* (2018)	Não	Não	Sim	Não	Sim	Sim
Boudi *et al.* (2019)	Sim	Sim	Não	Não	Sim	Sim
Sairam *et al.* (2019)	Sim	Sim	Sim	Não	Sim	Sim
Zarca, Bernabe, Trapero *et al.* (2019)	Não	Não	Sim	Não	Sim	Sim
Zarca, Garcia-Carrillo *et al.* (2019)	Sim	Não	Sim	Não	Sim	Sim
Zhou *et al.* (2019)	Não	Não	Sim	Não	Sim	Sim
Afek *et al.* (2020)	–	–	–	Não	Sim	Sim
Bagaa *et al.* (2020)	Não	Não	Sim	Não	Sim	Sim
Guizani e Ghafoor (2020)	Sim	Não	Não	Não	Sim	Sim
Zarca, Bernabe, Skarmeta *et al.* (2020)	Sim	Não	Sim	Não	Sim	Sim
Pei *et al.* (2020)	Sim	Sim	Não	Sim	Não	Não
Varasteh *et al.* (2021)	Sim	Sim	Não	Sim	Não	Não
Liu *et al.* (2021)	Sim	Sim	Não	Sim	Não	Não
Esta dissertação	Sim	Sim	Sim	Sim	Sim	Sim

Tabela 3.6: *Comparação entre os trabalhos analisados e esta dissertação.*

Capítulo 4

Abordagem Baseada no Desempenho do Contexto de Rede para Mitigar Ataques DDoS na IoT

Este capítulo apresenta o objetivo geral desta dissertação, que é uma abordagem baseada no desempenho do contexto de rede para mitigar ataques DDoS na IoT, investigando como as métricas - ou indicadores - de desempenho podem compor, como variáveis de decisão, modelos de ML para o posicionamento adequado de VNF.

A metodologia de pesquisa foi baseada na exploração de modelos de ML e na composição de indicadores de desempenho para o desenvolvimento de uma abordagem de mitigação de ataques DDoS. Os procedimentos foram dirigidos à experimentação e observação a partir de dois estudos em contextos de redes IIoT. Os resultados desta pesquisa têm contribuído com os avanços do projeto de pesquisa MENTORED[1], o qual busca desenvolver sistemas de detecção de intrusão (NIDS – *Network Intrusion Detection System*) diante da natureza distribuída de redes IoT.

A análise da literatura permitiu observar que os indicadores designados para avaliar o desempenho da virtualização de funções de rede são menos explorados que os indicadores utilizados para demonstrar a eficácia dos métodos propostos pelos diversos autores. Um exemplo claro disso é o tempo aferido para a implantação de uma ou mais máquinas virtuais, que é encontrado apenas em dois dos trabalhos (Zarca, Bernabe, Skarmeta *et al.*, 2020; Sairam *et al.*, 2019) que analisam o desempenho da abordagem proposta. Outro ponto importante é que o desempenho da rede também é pouco explorado, mesmo em trabalhos que consideram a implementação de uma arquitetura em névoa. O desempenho da rede é observado pelos autores em apenas dois dos trabalhos (Boudi *et al.*, 2019; Sairam

[1] MENTORED é um projeto cooperativo entre a Universidade Federal de Minas Gerais (UFMG), a Universidade Federal de Pernambuco (UFPE), a Universidade de São Paulo (USP), o Instituto Federal de Santa Catarina (IFSC), a Universidade do Vale do Itajaí (UNIVALI) e a Rede Nacional de Pesquisa (RNP). Fonte: https://www.mentoredproject.org. Acessado em 17 de Julho de 2022.

et al., 2019) analisados. Avaliar o comportamento de recursos virtualizados por meio dessas métricas é essencial, pois impacta diretamente no gerenciamento de desempenho da rede e se torna ainda mais importante quando funções de segurança são implementadas por meio de NFV. Um exemplo de métrica que se destaca em redes IoT é a latência de rede. A busca por sua redução é um dos principais objetivos de paradigmas como a computação em névoa (Yousefpour *et al.*, 2019; Lin *et al.*, 2017; Singh *et al.*, 2016) e estratégias de otimização de rotas por meio de SDN (Okay e Ozdemir, 2018; N. Wang *et al.*, 2008).

Dessa forma, o conteúdo apresentado na Seção 4.1 descreve um modelo de ML que compõe variáveis do contexto de redes IIoT para predição do tempo de resposta de ambientes em névoa. O conjunto de dados utilizado para os experimentos contempla aplicações genéricas que visam responder às requisições específicas de dispositivos industriais, aplicações essas que podem ser modeladas e desenvolvidas como VNFs de segurança. Já a Seção 4.2 apresenta um modelo de ML para posicionamento de VNFs de segurança visando mitigar ataques DDoS na IIoT, após a composição do conjunto de dados anterior para contemplar aplicações de segurança e cenários de rede sob ataques DDoS.

4.1 Predição do Tempo de Resposta em Ambientes SDN-Fog para Aplicações IIoT

Em cenários de aplicação de IoT, o tempo de resposta para uma determinada atividade é um dos atributos que mais requer atenção (Xu *et al.*, 2018) e, por isso, o paradigma de processamento computacional descentralizado ganhou espaço nesse contexto. Um exemplo disso é a computação em névoa, a qual suporta aplicações que requerem tempo de resposta estrito, separando o processamento de uma dada tarefa em nós de capacidade de processamento distintos interconectados por uma rede (Yousefpour *et al.*, 2019). Dessa forma, separando o processamento em diferentes nós de uma rede, outro fator a ser considerado na composição do tempo de resposta é a latência e, para tomar uma decisão ótima sobre a localização dos dispositivos de processamento, o problema de distribuição computacional descentralizada (DCDP – *Decentralized Computation Distribution Problem*) (Choudhury *et al.*, 2019) precisa ser resolvido.

No sentido de reduzir a latência de uma computação em névoa, abordagens de otimização de roteamento podem ser empregadas (N. Wang *et al.*, 2008) usando SDN (Okay e Ozdemir, 2018). No entanto, a latência entre os controladores SDN e os switches também deve ser considerada no cálculo do tempo de resposta entre os dispositivos, ou seja, se os controladores forem colocados de forma a minimizar essa latência, a latência entre os dispositivos da rede será minimizada também. O problema de posicionar o controlador de maneira ideal é conhecido como problema de posicionamento do controlador SDN (CPP – *Controller Placement Problem*) (Das *et al.*, 2020).

A combinação de ambos os problemas, DCDP e CPP, resulta em outro: o problema de implantação SDN-Fog, proposto inicialmente por Herrera *et al.* (2021) para descrever uma abordagem de esforço único para resolvê-los juntos. A abordagem proposta consiste em um *framework* chamado DADO (do inglês, *Distributed Application Deployment Optimization*), que utiliza programação linear inteira mista (MILP – *Mixed Integer Linear Programming*)

em seu núcleo para identificar a implantação de ambos os serviços e controlador SDN a fim de atender a requisitos específicos.

Os resultados apresentados pelos autores mostraram que o *framework* é adequado para pequenas topologias de rede, infraestruturas com menos de 300 nós, nas quais o problema de otimização é resolvido em poucos segundos. No entanto, a solução leva entre 4664 e 20982 segundos e precisa de aproximadamente 18 exabytes de memória em redes maiores, porque o problema de implantação do SDN-Fog é NP-difícil.

MILP envolve problemas em que apenas algumas das variáveis são restritas a valores inteiros, enquanto outras variáveis podem ser não inteiras. Uma maneira de otimizar modelos MILP é utilizar heurísticas como *branch-and-bound* e *branch-and-cut* (Pióro e Medhi, 2004). No entanto, a complexidade da resolução tende a não diminuir quando o ambiente é dinâmico, ou seja, quando há um número excessivo de variáveis e restrições, como ambientes de rede IoT complexos e tráfegos com fluxos do mundo real (Hardegen *et al.*, 2020).

A previsão do comportamento da rede pode trazer vantagens para muitas aplicações que possuem requisitos estritos de QoS (Mirkovic *et al.*, 2018), como posicionamento de dispositivos e alocação de recursos, além de permitir que projetistas de serviço prevejam o desempenho de suas aplicações para diferentes cargas (Bhulai *et al.*, 2007). Por exemplo, para o problema de otimização apresentado anteriormente, prever as variáveis que compõem a rede pode ser útil ao invés de apenas considerar a solução do problema modelada no MILP, que requer um longo tempo de execução e alto poder computacional. Além disso, prever o tempo de resposta da rede pode ser muito útil no planejamento da capacidade da rede, pois pode elucidar o comportamento de uma determinada configuração sem necessariamente implementá-la. Assim, os gerentes de infraestrutura podem atestar sua estratégia e ver se os componentes de rede e os dispositivos são suficientes para garantir uma boa QoS aos usuários finais. No caso específico desta dissertação, isso possibilitaria prever o impacto da virtualização de algumas funções de segurança de rede.

Com base no aprendizado supervisionado e não supervisionado, um *framework* de ML foi criado para prever o tempo de resposta de ambientes SDN-Fog antes de realizar todas as configurações de rede possíveis. A Figura 4.1 ilustra como a abordagem de ML criada pode se integrar com a solução baseada em MILP. Nesse contexto, prever os tempos de resposta de possíveis cenários de rede ajuda a selecionar o mais adequado em relação ao tempo de resposta esperado. Assim, a execução da otimização em uma etapa posterior torna-se menos onerosa e em menos tempo.

4.1.1 IIoT em Névoa

IIoT, ou *Industrial Internet of Things*, é uma especialização da IoT para indústrias. O principal objetivo é agregar inteligência, por meio de sensores, a equipamentos de produção conectados entre si e, até mesmo, à Internet. Através dos dados que coletam, esses sensores ajudam as indústrias a melhorar a produtividade, a eficiência dos serviços, a segurança dos funcionários, entre outros.

O *framework* de ML criado foi focado em prever com alta precisão o tempo de resposta esperado de uma configuração de rede IIoT em névoa específica. O *framework* avaliou como

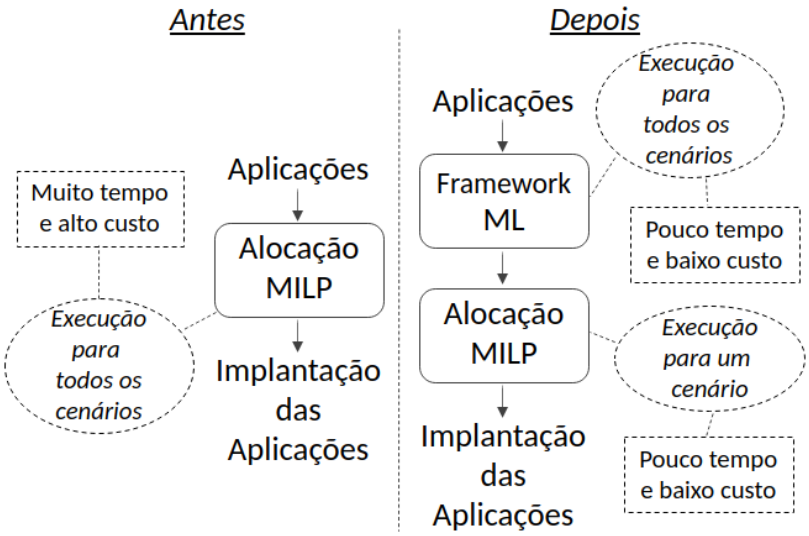


Figura 4.1: *Abordagem baseada em ML.*

diferentes abordagens de ML, como regressão linear e árvores de decisão, podem prever variáveis contínuas, neste caso o tempo de resposta da rede. Além disso, o *framework* também buscou avaliar como a técnica de aprendizado não supervisionado com base na redução de dimensionalidade impacta a previsão.

4.1.2 O Conjunto de Dados

O cenário IIoT em névoa considerado é semelhante ao ambiente proposto por Herrera *et al.* (2021). Esse ambiente é baseado na computação em névoa para suportar os requisitos de QoS dos dispositivos IIoT e descreve os detalhes necessários para prever o tempo de resposta das aplicações neste contexto. É composto por dispositivos IIoT instalados em robôs, além de dez servidores em névoa que fornecem serviços de computação aos dispositivos. Cada servidor em névoa possui uma CPU de 800 MHz e 1 GB de RAM e é conectado diretamente a um *switch* SDN. Em relação aos controladores SDN, o ambiente inclui pelo menos um controlador para cada *switch*, ou seja, um modelo clássico de controle SDN. Um aplicativo que gerencia as informações de robôs foi projetado compondo uma série de microsserviços. Esse aplicativo coleta o status dos robôs por meio de sensores conectados a dispositivos IIoT e, em seguida, processa os dados em servidores em névoa para fornecer comandos adequados.

Os comandos processados pelos microsserviços podem ser solicitados separadamente ou combinados por meio de fluxos de trabalho. Cada fluxo consiste na execução de uma funcionalidade específica, vinculando entre um e seis microsserviços, dependendo do número necessário para executar a funcionalidade. Cada microsserviço possui uma complexidade computacional para realizar o processamento de uma determinada atividade. Este processamento é feito por MCycles (Sun *et al.*, 2018), estando entre 100, 500 ou 1000.

Portanto, a combinação do tempo de execução de um microsserviço e da latência entre a comunicação dos nós da rede resulta no tempo de resposta de uma determinada funciona-

lidade. Essa combinação é diretamente influenciada pela localização dos microsserviços em diferentes servidores em névoa e pelos controladores SDN disponíveis no ambiente. Assim, é necessário que a redução da latência entre os vários nós da rede seja buscada através da combinação de DCDP e CPP para garantir que o tempo de resposta de uma funcionalidade específica seja consistente com os rígidos requisitos das aplicações IIoT.

O conjunto de dados da IIoT em névoa consiste em 17 atributos por cenário: recursos de nós de computação, como ID, velocidade do *clock* (MHz) e RAM (MB); IDs de *switches*; recursos de enlace, como o ID do nó de origem, ID do nó de destino, latência (s) e capacidade (MB); recursos de microsserviços, como ID do microsserviço, ciclos de execução (MCycles), RAM necessária (MB), tamanho de entrada e saída do microsserviço (MB); e recursos de fluxo de trabalho, como ID, a lista de IDs dos microsserviços solicitados no fluxo de trabalho, ID do nó de computação que solicita o fluxo de trabalho e se o fluxo de trabalho deve ter uma resposta.

4.1.3 O *Framework*

Como mencionado anteriormente, a modelagem e execução do problema de implantação SDN-Fog em MILP pode resolver DCDP e CPP juntos para otimizar o tempo de resposta de aplicações IIoT. Porém, a execução dos modelos requer um alto poder computacional, além de consumir um tempo relativamente grande para redes com mais de 300 nós (Herrera *et al.*, 2021). Assim, a fim de evitar esses problemas apresentados, o *framework* baseado em ML foi desenvolvido usando recursos do ambiente de rede para prever o tempo de resposta de cenários IIoT.

A Figura 4.2 apresenta as duas partes principais do *framework*. A primeira parte é responsável por processar os dados de entrada e saída do DADO (Herrera *et al.*, 2021; Herrera *et al.*, 2020), respectivamente, para compor os cenários que são as entradas primárias para os modelos de ML. A segunda parte é composta pela implementação e execução de abordagens de aprendizado supervisionado e não supervisionado para avaliar o comportamento de predição do tempo de resposta dos cenários compostos pela primeira parte.

Para compor os cenários, a etapa de processamento de dados utiliza os recursos do ambiente IIoT em névoa juntamente com uma estimativa do tempo de resposta para diferentes fluxos de trabalho para compor os cenários. Cada cenário teve seus atributos estendidos estatisticamente de acordo com seus valores médios, mínimos, 25%, 50%, 75% e máximos, representando 113 atributos no total.

Como apresentado no Capítulo 3, ML tem auxiliado e contribuído de forma ampla para diversas melhorias propostas relacionadas à segurança de redes IoT, como, por exemplo, na detecção de anomalias. Entretanto, os modelos de ML podem se comportar de forma distinta de acordo com o contexto em que são inseridos. Portanto, avaliar o desempenho desses modelos é o primeiro passo antes de implantá-los em produção.

Assim, o *framework* é composto por Regressão Linear e eXtreme Gradient Boosting (XGBoost) como modelos de ML supervisionados. O objetivo da Regressão Linear é encontrar as melhores estimativas para seus coeficientes, minimizando os erros na previsão de características do alvo. Ele se ajusta a um modelo linear para minimizar a soma residual

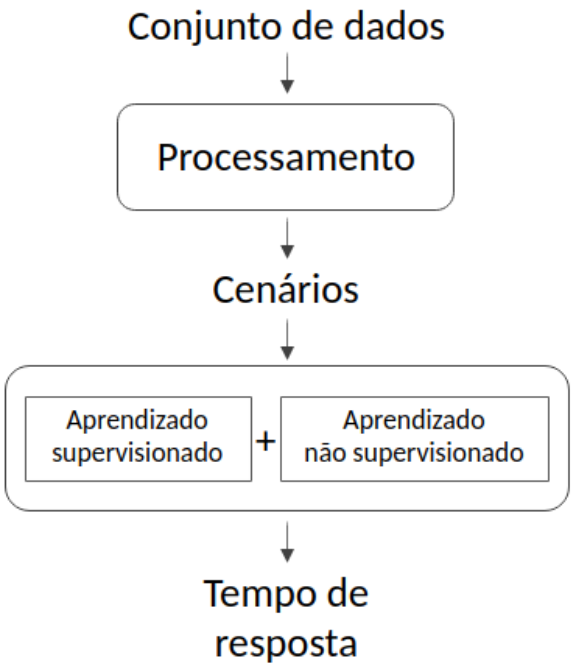


Figura 4.2: *Estrutura do framework.*

dos quadrados entre os alvos observados no conjunto de dados. No contexto IIoT em névoa, o tempo de resposta da rede representa os alvos previstos pela aproximação linear. O XGBoost é uma implementação de Gradient Boosted Decision Trees projetada para velocidade e desempenho proposta por Chen e Guestrin (2016). Esse algoritmo visa minimizar a função de perda, escolhendo valores de ajuste ótimos distintos para cada região da árvore ao invés de um único para toda a árvore (Friedman, 2001). Com base nesse conceito, o XGBoost calcula o ajuste de suas árvores sequenciais de acordo com os erros ponderados das árvores predecessoras para definir a estrutura final da árvore de decisão do modelo. Por exemplo, no contexto de rede considerado, os nós internos da árvore final representam os parâmetros de decisões baseados nos indicadores de desempenho e os nós folha os tempos de respostas da rede a serem previstos.

Para avaliar como a técnica de aprendizado não supervisionado impacta a previsão do tempo de resposta, a abordagem de redução de dimensionalidade utilizando a análise de componentes principais (PCA – *Principal Component Analysis*) foi considerada. A PCA é utilizada para reduzir o tamanho de um conjunto de dados enquanto mantém os recursos que têm a maior contribuição de variação. É um método de análise estatística que realiza uma transformação ortogonal no intuito de transformar linearmente as observações de uma série de variáveis possivelmente relacionadas e, em seguida, projetá-las em uma série de variáveis linearmente não correlacionadas. As variáveis não correlacionadas são chamadas de componentes principais (Alsharif *et al.*, 2020; K. Gao *et al.*, 2020; Rafique e Velasco, 2018). No contexto de rede aqui considerado, as variáveis que contemplam a transformação PCA são os indicadores de desempenho da rede.

4.1.4 Experimentos, Resultados e Discussões

Os experimentos realizados foram baseados na comparação dos modelos de ML implementados por meio das bibliotecas de código aberto Scikit-learn e XGBoost para todos os cenários compostos na etapa de processamento do conjunto de dados. Os Notebooks do Jupyter desenvolvidos para os experimentos podem ser encontrados no Github[2].

Neste estudo, três categorias de topologia foram consideradas: pequena (10 dispositivos IIoT e 10 servidores em névoa), média (25 dispositivos IIoT e 15 servidores em névoa) e grande (50 dispositivos IIoT e 25 servidores em névoa). O universo de dados utilizado para treinar os modelos contém 84 cenários de rede, contabilizando 880 nós para pequenas topologias (44*(10 dispositivos IIoT + 10 servidores em névoa)), 1000 nós para médias (25*(25 dispositivos IIoT + 15 servidores em névoa)) e 1125 nós para as grandes (15*(50 dispositivos IIoT + 25 servidores em névoa)).

Para testar os modelos, o universo de dados incluiu 47 cenários que totalizam 880 nós para topologia pequena (44*(10 dispositivos IIoT + 10 servidores em névoa)) e 240 nós para topologia grande (3*(50 dispositivos IIoT + 30 servidores em névoa)). A diferença na quantidade de servidores em névoa é original do conjunto de dados de origem e foi mantida com o objetivo de avaliar o desempenho dos modelos em estruturas distintas das utilizadas para treinamento.

Antes de compor os modelos com a técnica PCA, foi realizado um estudo dos principais componentes da rede. Nele foi possível verificar que quatro componentes seriam ideais para composição com os modelos de aprendizagem supervisionada. A Figura 4.3 mostra o número de componentes necessários para explicar a variabilidade dos dados originais. O primeiro componente corresponde a 56,87%, o segundo a 21,44%, o terceiro a 13,01% e o quarto a 4,90% da variabilidade. Portanto, selecionar um número de componentes acima de quatro para realizar a redução da dimensionalidade não iria influenciar drasticamente o modelo, uma vez que os demais componentes correspondem a apenas 3,78% da variabilidade. Além disso, de acordo com a matriz de coeficientes calculada, as características que mais contribuíram para a formação do primeiro componente estão relacionadas ao poder computacional dos dispositivos IoT, como processamento e memória. Para o segundo componente, os recursos de número de nós da rede, IoT, servidores em névoa e *switch*, e recursos relacionados aos enlaces, como capacidade e latência, foram os que mais contribuíram. Para aqueles relacionados às etapas do fluxo, eles formaram o terceiro componente. Para o quarto e último componente, os recursos que mais contribuíram estão relacionados aos tamanhos dos fluxos e ao número de dispositivos por fluxos. A validação de desempenho dos modelos foi implementada por meio da validação cruzada *5-fold*, ou seja, cinco medidas diferentes validam os modelos implementados, avaliando sua capacidade de generalização de predição e prevenção de *overfitting*.

As Figuras 4.4a e 4.5a ilustram uma representação gráfica do desempenho da Regressão Linear e do XGBoost na etapa de treinamento, e as Figuras 4.4b e 4.5b na etapa de teste. Ao comparar as etapas de treinamento e teste de ambos os modelos, um fator que impactou na queda de desempenho foi o baixo número de cenários de teste, uma das limitações encontradas. Apesar disso, é possível verificar pela aproximação da linha azul, que representa os

[2] Fonte: https://github.com/rodrigoney/ml-sdn-dado. Acessado em 17 de Julho de 2022.

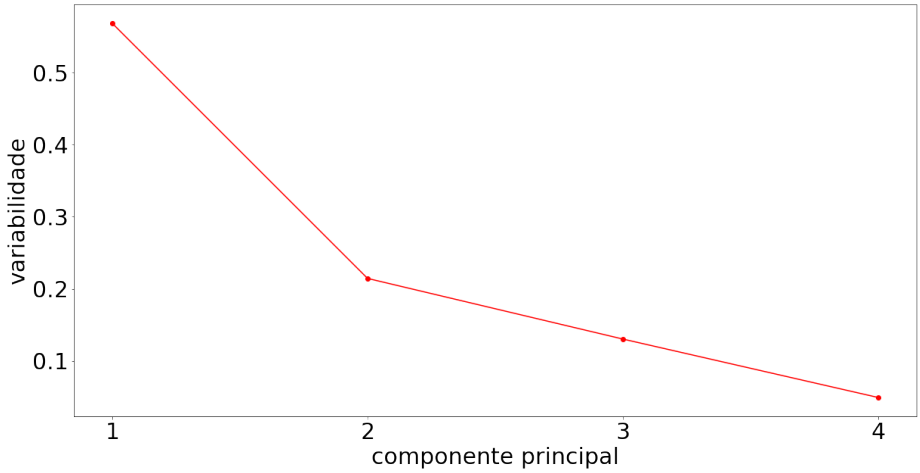


Figura 4.3: *Impacto dos quatro componentes principais.*

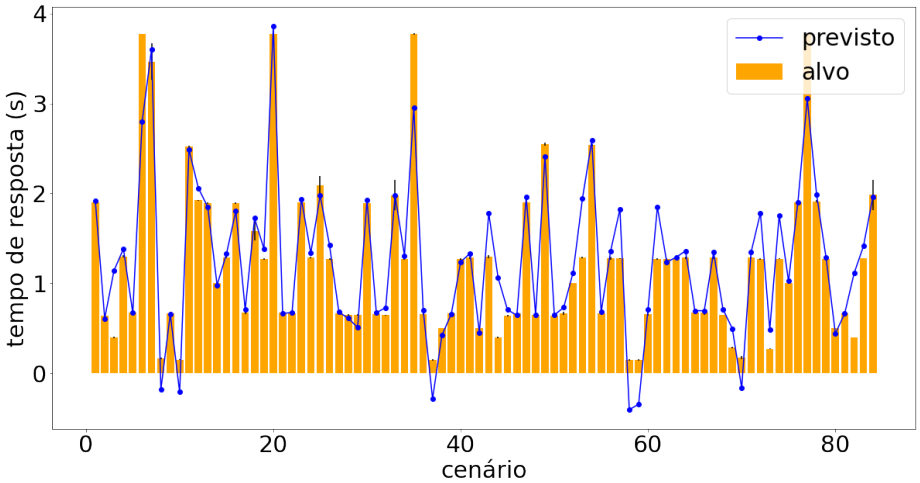
tempos de resposta previstos, que o modelo XGBoost teve o melhor desempenho quando comparado à Regressão Linear.

Os métodos de medição de erro quadrático médio (MSE – *Mean Squared Error*) e R^2, ou R-quadrado, foram utilizados para avaliar a acurácia dos modelos além da representação gráfica. Enquanto que o R^2 representa a proporção da variância para uma variável dependente que é explicada por uma variável independente ou variáveis em um modelo de regressão, o MSE mede a diferença quadrática média entre os valores estimados e o valor real (Skiena, 2017). A precisão resultante dos modelos implementados está descrita na Tabela 4.1.

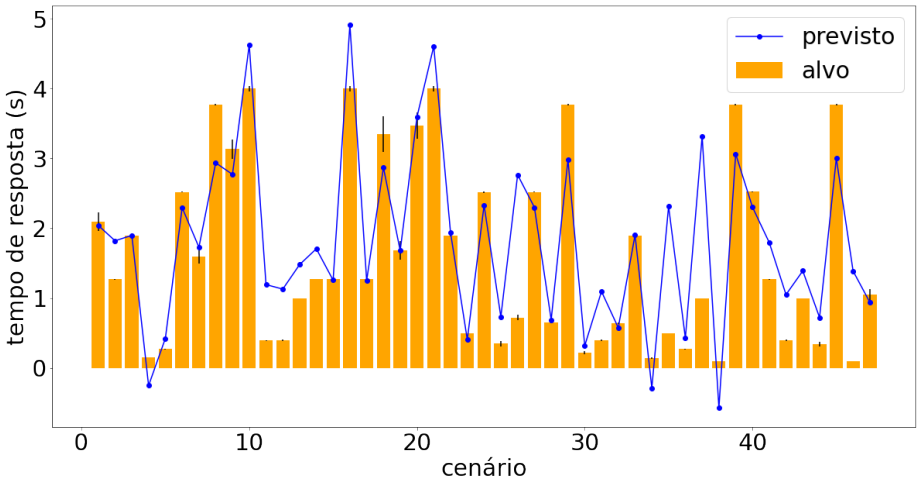
Modelo	**MSE**		R^2	
	Treinamento	**Teste**	**Treinamento**	**Teste**
Regressão Linear	0.0822	0.4999	0.8892	0.7025
Regressão Linear com PCA	0.1547	0.8215	0.7917	0.5111
XGBoost	0.0107	0.0885	0.9854	0.9473
XGBoost com PCA	0.1977	0.9754	0.7338	0.4196

Tabela 4.1: *Acurácia dos modelos implementados.*

Embora os quatro componentes principais representem 96,22% de todos os dados, é possível observar que a estratégia PCA não melhora o desempenho de ambos os modelos, nem pela redução do MSE nem pelo aumento do R^2. Além disso, ao comparar os modelos de Regressão Linear com PCA e XGBoost com PCA, é possível ver um impacto negativo maior no XGBoost. Esse fenômeno ocorre devido à redução ineficaz do erro calculado através da baixa composição de atributos em sucessivas árvores utilizadas no aumento de gradiente. Como o XGBoost utiliza vários atributos para calcular o erro e compô-lo na função de aprendizado, reduzir os atributos do ambiente IIoT mostrados não é uma boa estratégia para prever o tempo de resposta da rede. Além disso, o modelo XGBoost atinge o melhor desempenho a uma taxa média de 49,60 milissegundos segundo o MSE e 96,63%



(a) *Treinamento.*



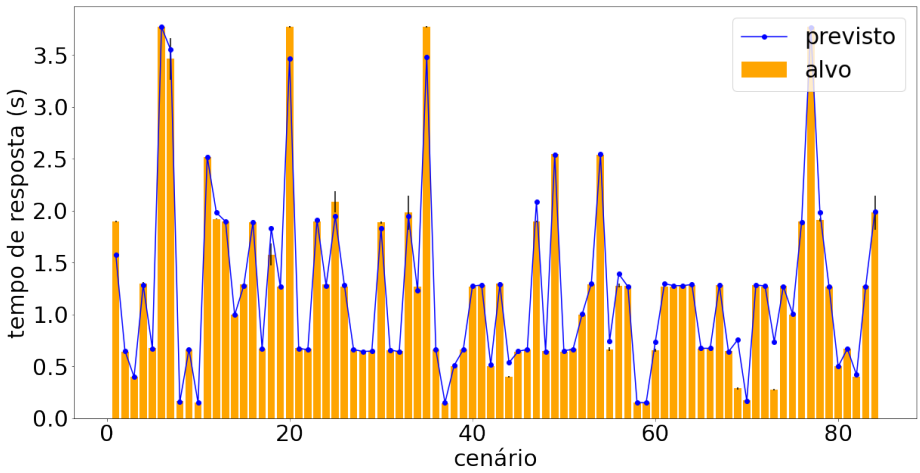
(b) *Teste.*

Figura 4.4: *Desempenho da Regressão Linear para a previsão do tempo de resposta da rede.*

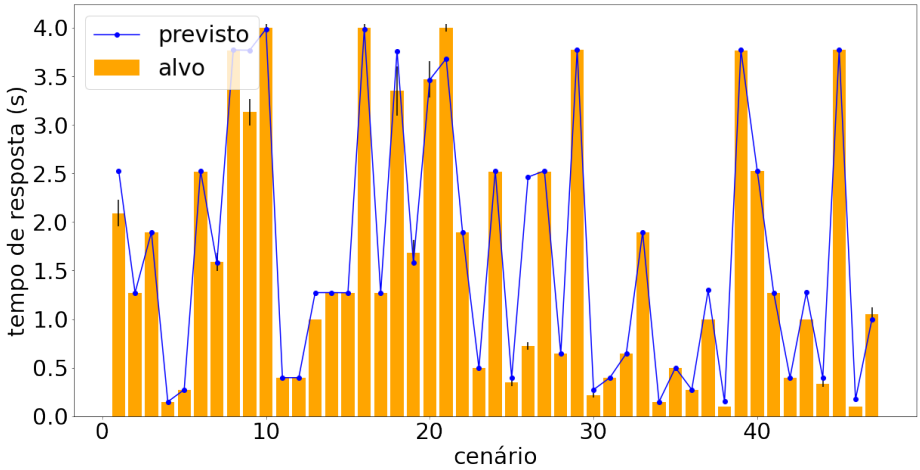
segundo o R^2.

Para aprofundar o estudo dos impactos dos dados em relação aos modelos implementados, foram realizadas duas abordagens: a normalização e a aplicação da técnica *z-score*. A Tabela 4.2 mostra que a aplicação da normalização de dados para o modelo de regressão linear melhora discretamente seu desempenho apenas para R^2, diferentemente do modelo XGBoost, que é aprimorado com a aplicação da técnica *z-score* em ambas métricas de avaliação.

Em geral, os resultados mostram que a previsão do tempo de resposta dos cenários foi próxima aos tempos obtidos pela solução do problema apresentado por Herrera *et al.* (2021), ou seja, quando comparado aos resultados de Herrera *et al.* (2020), o *framework* obteve uma média abaixo de 50 milissegundos de variação de acordo com o modelo de maior precisão e foi executado em menos de um segundo para os modelos de Regressão Linear e XGBoost, o que torna a solução útil para um ambiente de implantação de VNFs de segurança e que precise de um retorno rápido, já que a solução baseada em MILP poderia



(a) *Treinamento.*



(b) *Teste.*

Figura 4.5: *Desempenho do XGBoost para a previsão do tempo de resposta da rede.*

Modelo	**MSE**		R^2	
	Treinamento	**Teste**	**Treinamento**	**Teste**
Regressão Linear Normalizada	0.0857	0.8065	0.8845	0.7392
Regressão Linear *Z-Scored*	0.0924	1.3510	0.8755	0.5632
XGBoost Normalizado	0.1796	1.064	0.7582	0.6558
XGBoost *Z-Scored*	0.0108	0.8904	0.9854	0.7121

Tabela 4.2: *Acurácia dos modelos com dados normalizados.*

levar algumas horas para retornar a estimativa de tempo de resposta.

Outro fator a ser considerado é a composição de modelos de aprendizado supervisionado e não supervisionado para prever o tempo de resposta da rede. Neste caso, a técnica de PCA não supervisionada não se mostrou uma alternativa vantajosa para o contexto IIoT apresentado quando composto com os modelos desenvolvidos supervisionados.

Foi possível observar, por meio dos experimentos, que as informações do ambiente de rede, como latência e capacidade, além das informações de processamento dos serviços, são essenciais para a tomada de decisão quando o objetivo é otimizar a QoS fornecida aos usuários finais. Esses resultados ajudam a mostrar como as variáveis do ambiente de redes IoT podem ser utilizadas para compor soluções de maior eficácia - fato esse que suportou e direcionou a pesquisa a validar a hipótese de que existem vantagens em implementar estratégias baseadas em ML para mitigar ataques DDoS em redes IoT, utilizando como variáveis de decisão as métricas - ou indicadores - de desempenho do próprio recurso NFV e de recursos da rede.

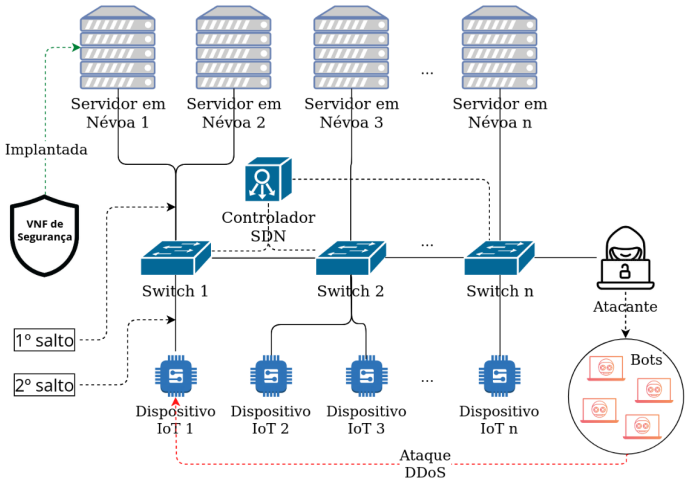
4.2 Posicionamento Inteligente de VNF para Mitigar Ataques DDoS na IIoT

Como mencionado anteriormente, o gerenciamento de desempenho pressupõe como utilizar indicadores ou métricas coletadas da rede para provisionar com eficiência recursos necessários para manter a QoS entregue a seus usuários. Para que a operação de uma VNF seja realizada de forma eficiente, os indicadores de desempenho devem ser bem definidos e medidos a fim de demonstrarem a real demanda, o comportamento e a "saúde" de uma rede. A previsão do comportamento da rede traz vantagens para muitas aplicações que possuem requisitos rigorosos de QoS por antecipar o posicionamento de dispositivos e alocação de recursos, além de permitir que *designers* de serviços prevejam o desempenho de suas aplicações (Oliveira, Ney *et al.*, 2021; Mirkovic *et al.*, 2018; Bhulai *et al.*, 2007).

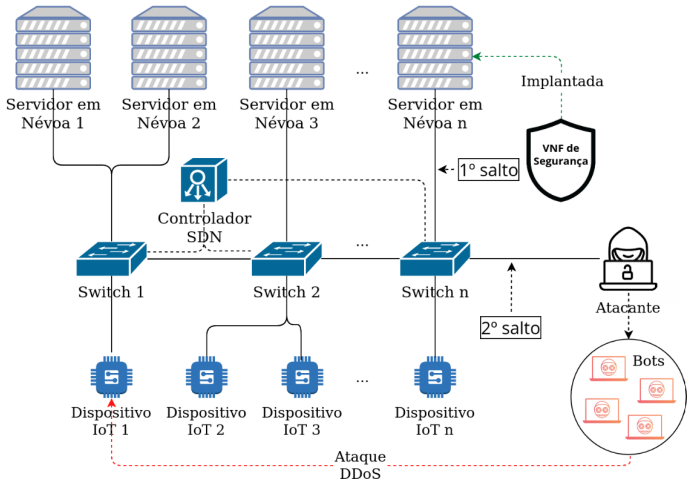
Assim, indicadores de desempenho podem contribuir para a tomada de decisão de modelos de ML sobre o uso ou não de um determinado recurso virtualizado para uma finalidade identificada em locais específicos. Em outras palavras, o gerenciamento de desempenho de NFV pode ser suportado por decisões de ML. Por exemplo, para mecanismos de posicionamento de VNF apresentados por S. Yang *et al.* (2021), Song *et al.* (2020), Shih *et al.* (2019), Z. Jia *et al.* (2018) e Yu *et al.* (2015), rotular, treinar e classificar servidores específicos para implantar uma VNF pode ser útil ao invés de apenas considerar a solução do problema por técnicas de otimização, como time-slot desacoplado, aproximação de arredondamento aleatório, heurística, particionamento de grafos, algoritmos gananciosos e de teoria dos jogos, que pode consumir longo tempo de execução e alto poder computacional.

Muitos dispositivos na IoT podem ser alvos e até mesmo meios para uma série de ataques, pois não oferecem suporte a mecanismos de segurança fortes (Meneghello *et al.*, 2019). Portanto, a Figura 4.6 ilustra o cenário considerado aqui. O tráfego da rede é monitorado e serve como entrada para um mecanismo que analisa o tráfego em busca de ataques DDoS a dispositivos IoT alvos. Caso algo seja detectado, VNFs de segurança são posicionadas na rede para mitigar o ataque. Um bom posicionamento, baseado nas

próprias variáveis que demonstram o estado atual da rede, pode ser alcançado utilizando os mecanismos apresentados nesta dissertação.



(a) *Posicionamento da VNF de segurança próximo ao dispositivo IoT atacado.*



(b) *Posicionamento da VNF de segurança próximo ao dispositivo atacante.*

Figura 4.6: *Formas de posicionamento da VNF de segurança em redes IoT sob ataques DDoS.*

O Algoritmo 4.1 descreve a técnica de posicionamento de VNF de segurança que foi desenvolvida. Ele é baseado em decisões de modelos de ML previamente treinados para definir o melhor servidor no qual uma VNF de segurança pode ser implantada para mitigar ataques DDoS compondo diferentes indicadores de desempenho. Dessa forma, o problema de posicionamento aqui tratado pode ser compreendido como uma classificação n-ária, onde n é a identificação do servidor na rede. Para testar a eficácia dessa abordagem, foram avaliados os desempenhos de modelos supervisionados e não supervisionados, apresentados na Subseção 4.2.2, em cenários IIoT em névoa, compostos por ataques DDoS. Esta composição é descrita com mais detalhes na Subseção 4.2.1.

Programa 4.1 Posicionamento de VNF de Segurança.

```
1   FUNCAO posiciona_vnf(X, Y, m)  ▷ X é o conjunto de treinamento contendo os cenários
        de rede IIoT, Y é o conjunto de teste contendo os cenários de rede IIoT e m é um
        determinado modelo de ML
2   enquanto X ≠ ∅  ▷ Etapa de treinamento
3     x ← X  ▷ x é um cenário de rede IIoT de X
4     x_train ← get_train(x)  ▷ x_train é o conjunto de indicadores de desempenho da rede
          IIoT de x
5     x_target ← get_target(x)  ▷ x_target é o servidor em névoa alvo onde a VNF de
          segurança é posicionada de x
6     train_model(m, x_train, x_target)  ▷ Treinando o modelo de acordo com os indicadores de
          desempenho e o servidor em névoa alvo
7   enquanto Y ≠ ∅  ▷ Etapa de teste
8     y ← Y  ▷ y é um cenário de rede IIoT de Y
9     y_pred ← get_train(y)  ▷ y_pred é o conjunto de indicadores de desempenho da rede IIoT
          de y
10    pred_fog_server ← predict(m, y_pred)  ▷ Predizendo o servidor em névoa alvo de acordo
          com o modelo treinado m
11    Fog_Servers ← pred_fog_server  ▷ Fog_Servers é o conjunto de servidores em névoa onde
          as VNFs de segurança são posicionadas
12  devolva Fog_Servers
13  fim
```

4.2.1 O Conjunto de Dados

Com o objetivo de compor um cenário de rede IoT realista que correspondesse aos indicadores de desempenho considerados na discussão entre a literatura e as questões de pesquisa (Seção 3.2), um novo conjunto de dados foi criado[3] a partir de outros dois já existentes. O primeiro é referente ao cenário apresentado nos experimento anteriores (Subseção 4.1.2) que consiste em dispositivos IIoT instalados em robôs e servidores em névoa que disponibilizam serviços de computação aos dispositivos, contendo indicadores de desempenho da rede, como latência, taxa de transferência, entre outros. O segundo conjunto de dados refere-se ao BoT-IoT (Moustafa, 2019). Este conjunto de dados inclui tráfego de rede IoT legítimo e simulado, juntamente com vários tipos de ataques. Para este trabalho, foram considerados os ataques DDoS de acordo com os três tipos presentes no BoT-IoT: TCP, UDP e HTTP. Inicialmente, todas as variáveis de tráfego de rede foram extraídas de instâncias selecionadas aleatoriamente. Posteriormente, ataques DDoS foram filtrados com seus respectivos tráfegos de rede e incorporados no primeiro conjunto de dados. Além disso, tráfegos benignos também foram extraídos com o intuito de avaliar o comportamento dos modelos implementados na decisão de instanciar ou não uma VNF de segurança.

Para fechar o escopo de rede proposto, indicadores de desempenho relacionados à NFV e à Solução divulgados na literatura e organizados na Tabela 3.3 também foram incluídos

[3] Fonte: https://github.com/werneckg/master/tree/master/experiments/dataset. Acessado em 17 de Julho de 2022.

no novo conjunto de dados. Sobre NFV, três trabalhos relacionados (Zarca, Garcia-Carrillo *et al.*, 2019; Sairam *et al.*, 2019; Boudi *et al.*, 2019) referem-se ao consumo de CPU e RAM e outros dois (Zarca, Bernabe, Skarmeta *et al.*, 2020; Sairam *et al.*, 2019) ao tempo de implantação do recurso virtualizado. Em relação à Solução, dois trabalhos (Bagaa *et al.*, 2020; Sairam *et al.*, 2019) referem-se à acurácia dos modelos de ML e outros seis (Zarca, Bernabe, Skarmeta *et al.*, 2020; Zhou *et al.*, 2019; Zarca, Garcia-Carrillo *et al.*, 2019; Zarca, Bernabe, Trapero *et al.*, 2019; Zarca, Bernabe, Farris *et al.*, 2018; Al-Shaboti *et al.*, 2018) ao tempo de resposta ao ataque da abordagem proposta pelos autores. Essa análise permitiu a definição dos indicadores de desempenho a serem incluídos no novo conjunto de dados. Já o tempo de resposta previsto para cenários de rede IIoT (Oliveira, Ney *et al.*, 2021) permitiu definir suas respectivas posições no conjunto de dados. A Tabela 4.3 apresenta o racional qualitativo usado como referência. Cada indicador de desempenho inserido foi comparado com o tempo de resposta da rede individualmente. Por exemplo, quanto maior o tempo de resposta da rede, maior o tempo de resposta da abordagem. Portanto, são indicadores diretamente proporcionais. Diferentemente do poder computacional, CPU e RAM, que são inversamente proporcionais, ou seja, quanto menor o tempo de resposta da rede, maior o poder computacional requisitado pela solução virtualizada.

Indicador de desempenho	**Tempo de resposta da rede (Oliveira, Ney *et al.*, 2021)**	
	Baixo	**Alto**
Consumo de CPU (MHz)	Alto	Baixo
Consumo de RAM (MB)	Alto	Baixo
Tempo de implantação de ambiente (s)	Baixo	Alto
Acurácia do modelo de ML (%)	Baixa	Alta
Tempo de resposta da abordagem (s)	Baixo	Alto

Tabela 4.3: *Racional qualitativo para inclusão de indicadores de desempenho no conjunto de dados.*

4.2.2 Modelos de ML para Posicionamento de VNF de Segurança

Realizar uma análise de ambos os métodos de aprendizado, supervisionado e não supervisionado, auxilia a compreensão de como o contexto das redes IoT influencia a tomada de decisão. No entanto, a avaliação não pode se limitar a esse aspecto. É necessário, também, avaliar como os algoritmos estabelecidos na literatura se comportam quando os indicadores de desempenho definidos os alimentam. Para o aprendizado supervisionado, os algoritmos de Árvore de Decisão, Random Forest, XGBoost, Regressão Logística, Support Vector Machine, K-Nearest Neighbors e Naive Bayes foram implementados e testados. Para o aprendizado não supervisionado, foram considerados o K-Means, Hierarchical Cluster e Gaussian Mixture Model.

A Árvore de Decisão é uma árvore binária na qual as previsões são realizadas percorrendo a árvore da raiz à folha, onde está associada uma classe, ou um servidor em névoa específico para implantar uma VNF de segurança. Ela implementa o conceito de impureza de Gini para descrever o quão homogêneo é um nó. Um nó é homogêneo se

todas as suas amostras pertencem ao mesmo servidor em névoa, enquanto um nó com muitas amostras de vários servidores em névoa diferentes é menos puro. Diferentemente da Árvore de Decisão, o Random Forest cria pequenas árvores de decisão com vários conjuntos aleatórios de recursos ou indicadores de desempenho. Após essa criação, as previsões são feitas a partir de votos. Cada pequena árvore toma uma decisão com base nos dados apresentados e, a decisão mais votada, é o algoritmo final parametrizado. Como citado anteriormente, o XGBoost visa minimizar a função de perda, escolhendo valores de ajuste ótimo distintos para cada região da árvore ao invés de um único para toda a árvore. Com base nesse conceito, o XGBoost calcula o ajuste de suas árvores sequenciais de acordo com os erros ponderados das árvores predecessoras para prever o servidor em névoa que é o alvo do modelo.

Além das estruturas em árvore, o algoritmo de Regressão Logística estima a probabilidade associada à ocorrência de um determinado evento, o posicionamento da VNF de segurança, diante de um conjunto de variáveis explicativas, os indicadores de desempenho. Seus coeficientes são estimados a partir do conjunto de dados, pelo método da máxima verossimilhança, que encontra uma combinação de coeficientes que maximiza a probabilidade do servidor em névoa ter sido observado. O objetivo da Support Vector Machine é encontrar um hiperplano em um espaço n-dimensional, onde n é o número de indicadores de desempenho, que classificam distintamente o servidor em névoa como alvo. Embora a Regressão Logística e o Support Vector Machine sejam destinados à classificação de variáveis binárias, o comportamento desses algoritmos foram observados para n diferentes servidores em névoa a fim de buscar algum possível padrão na composição dos indicadores de desempenho.

O K-Nearest Neighbors classifica um novo servidor em névoa de acordo com a identificação de seus k vizinhos mais próximos, ou seja, ele busca grupos de servidores em névoa em comum com base em suas semelhanças, os indicadores de desempenho que descrevem a decisão de posicionamento da VNF. É um algoritmo $lazy$, pois memoriza o conjunto de dados de treinamento e classifica o novo servidor em névoa somente se houver proximidade numérica entre os dados aprendidos. O raciocínio básico do Naive Bayes é desconsiderar a correlação entre os indicadores de desempenho da rede, pois trata cada um de forma independente ou ingênua. Com este algoritmo, é possível observar como uma correlação baixa ou alta dos indicadores de desempenho do ambiente IIoT impacta na classificação do modelo.

Em relação aos algoritmos de aprendizado não supervisionado, o K-Means avalia e agrupa os servidores em névoa de acordo com suas características. A partir da definição k, ou o número previsto de servidores em névoa, o algoritmo define centroides no plano e separa os servidores em grupos k de igual variação. O objetivo do Hierarchical Cluster é produzir uma série hierárquica de conjuntos de nós aninhados. Ele começa assumindo cada servidor em névoa como um conjunto de nós separado e, em seguida, identifica dois conjuntos de nós que podem estar mais próximos e, finalmente, mescla-os. Essa hierarquia de conjunto de nós é representada como uma árvore em que a raiz é um conjunto de nós único que reúne todas as amostras e deixa os demais conjuntos de nós com apenas uma amostra. Como modelo probabilístico, o Gaussian Mixture Model considera que os servidores em névoa são gerados a partir de uma mistura de um número finito de distribuições Gaussianas com parâmetros desconhecidos, ou indicadores de desempenho.

Outra perspectiva essencial a ser analisada é a importância dos indicadores de desempenho para os modelos citados. Esta análise demonstra a contribuição proporcional de cada indicador de desempenho.

4.2.3 Experimentos, Resultados e Discussões

Da mesma forma que os experimentos anteriores (Subseção 4.1.4) os modelos de ML foram implementados por meio das bibliotecas de código aberto Scikit-learn e os Notebooks do Jupyter desenvolvidos para os experimentos apresentados a seguir podem ser encontrados no Github[4].

As mesmas três categorias de topologia foram consideradas: pequena (10 dispositivos IIoT e 10 servidores em névoa), média (25 dispositivos IIoT e 15 servidores em névoa) e grande (50 dispositivos IIoT e 25 servidores em névoa), assim como a divisão dos dados entre treinamento (84 cenários de rede) e teste (47 cenários de rede). As métricas de acurácia (Equação 4.1), precisão (Equação 4.2), *recall* (Equação 4.3) e F1-score (Equação 4.4) foram utilizadas para avaliar e comparar o desempenho dos modelos implementados. Para o contexto de redes IIoT, o posicionamento correto da VNF de segurança em um servidor em névoa específico é equivalente ao verdadeiro positivo (TP), um resultado no qual o modelo prevê corretamente a classe que identifica o servidor, e ao verdadeiro negativo (TN), quando o modelo prevê corretamente a não instanciação da VNF de segurança, ou seja, a classe identificada como zero. O falso positivo (FP) representa o posicionamento da VNF de segurança mesmo quando o cenário de rede não sofre ataques DDoS. Da mesma forma, o falso negativo (FN) representa a não instanciação da VNF de segurança quando a rede está sob ataque.

$$Acur = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

$$Prec = \frac{TP}{TP + FP} \tag{4.2}$$

$$Rec = \frac{TP}{TP + FN} \tag{4.3}$$

$$F1 = 2 * \frac{Prec * Rec}{Prec + Rec} \tag{4.4}$$

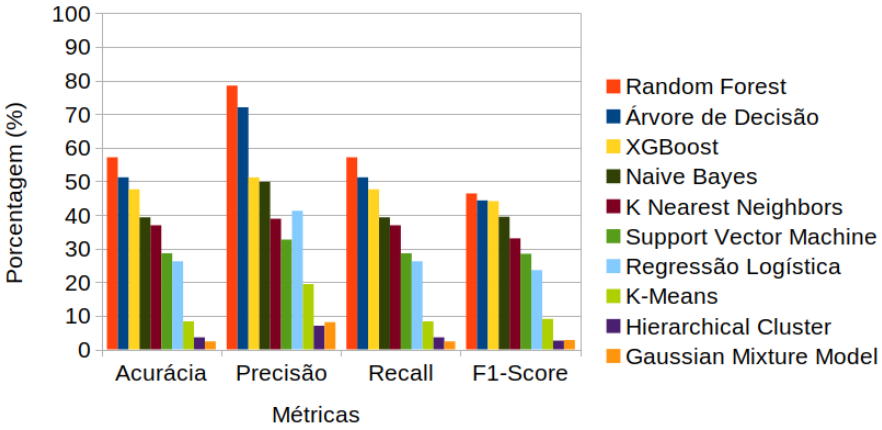
Para avaliar a capacidade de generalização da predição e evitar o *overfitting* dos modelos, a validação de desempenho dos modelos foi implementada por meio da validação cruzada *5-fold*. Além disso, foram definidos dois grandes cenários de testes, cujo posicionamento da VNF de segurança pode ser realizado próximo ao dispositivo IoT atacado ou próximo ao dispositivo atacante.

[4] Fonte: https://github.com/werneckg/master/tree/master/experiments/code. Acessado em 17 de Julho de 2022.

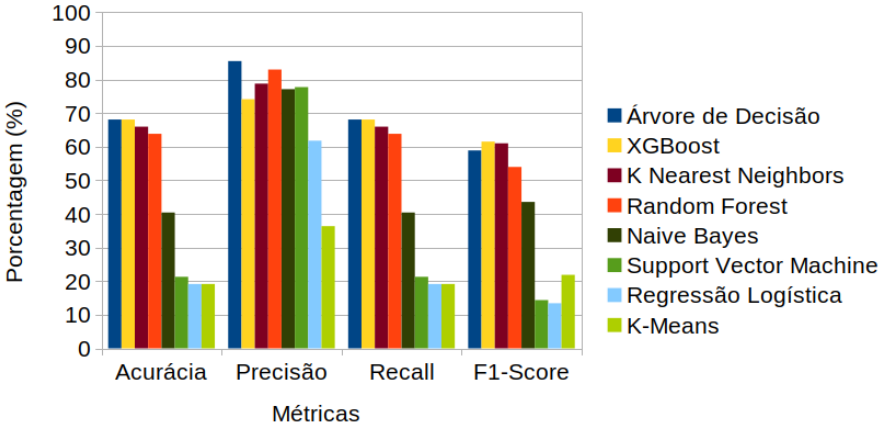
Posicionamento da VNF de Segurança Próximo ao Dispositivo IoT Atacado

Como qualquer dispositivo IoT é suscetível a ataques DDoS na rede definida, os modelos de ML foram avaliados de acordo com a classificação do posicionamento da VNF de segurança implantada especificamente em servidores em névoa próximos ao dispositivo IoT atacado, como ilustra a Figura 4.6a.

A Figura 4.7 resume o desempenho dos modelos de ML supervisionados e não supervisionados. Do ponto de vista da acurácia, todos os modelos apresentaram baixo desempenho, em média 30%. Quanto aos modelos não supervisionados, a média foi ainda menor, em torno de 5%. Esse comportamento é semelhante nos dados de treinamento e teste. Também, esse desempenho é estendido para métricas de precisão, *recall* e F1-score.



(a) *Treinamento.*



(b) *Teste.*

Figura 4.7: *Desempenho dos modelos de ML de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado.*

Focando nos modelos supervisionados, é possível observar nas Figuras 4.8, 4.9 e 4.10, os resultados com mais detalhes. Cada ponto no eixo vertical representa um servidor em névoa específico. As barras laranja representam o servidor em névoa ideal onde instanciar a VNF de segurança. Os pontos/linhas azuis representam o servidor em névoa selecionado pelo modelo de ML para instanciar a VNF de segurança (um modelo ideal teria todos os

pontos azuis no topo de todas as barras laranjas). Cada ponto do eixo horizontal representa um cenário de rede IIoT específico.

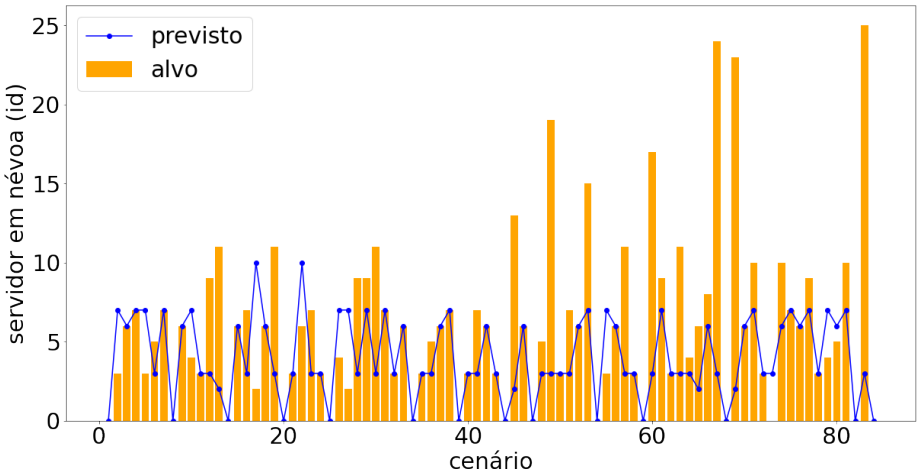
Cenários que possuem apenas tráfego normal, sem ataques DDoS, são cenários sem barras no gráfico, ou seja, o servidor em névoa é identificado pelo número zero. Isso representa o caso em que nenhuma instanciação é necessária. Os modelos obtiveram um baixo desempenho principalmente em cenários de rede que previam maior quantidade de servidores em névoa (Figuras 4.8a, 4.9a e 4.10a) a partir do cenário de rede 45.

Aprofundando e analisando a distribuição dos servidores em névoa, é possível observar um certo desequilíbrio na distribuição dos dados principalmente no conjunto de treinamento (Tabela 4.4). A maioria dos cenários de rede previam o posicionamento da VNF de segurança até o servidor em névoa 11. De fato, esse desequilíbrio contribui para o desempenho negativo dos modelos. No entanto, este não é o único fator que contribui para o baixo desempenho dos modelos não supervisionados. Nesse caso, não rotular o servidor em névoa de destino afeta negativamente a etapa de treinamento, como ilustram as Figuras A.5a, A.6a e A.7a. Consequentemente, isso também afeta a etapa de teste dos modelos (Figura A.5b).

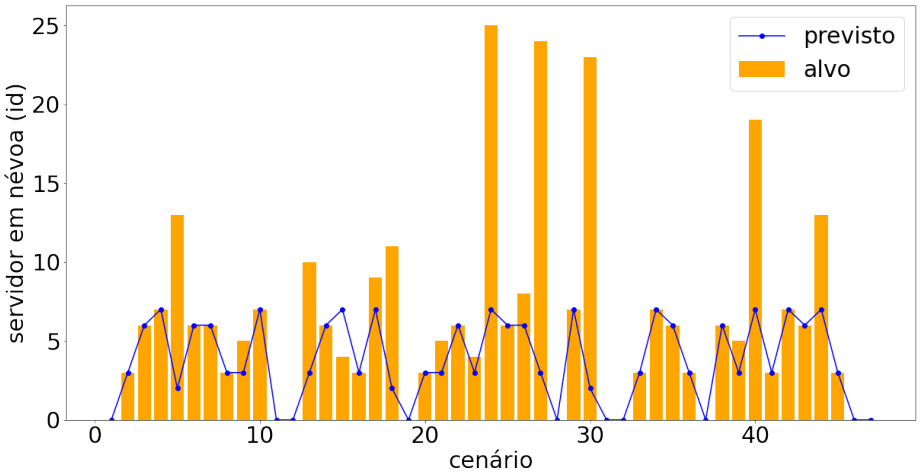
Servidor em névoa (id)	**Cenário de rede**
0	[0, 7, 13, 19, 24, 33, 38, 43, 46, 53, 58, 67, 72, 81, 83]
3	[1, 4, 10, 20, 23, 31, 34, 39, 42, 49, 54, 57, 61, 71, 77]
6	[2, 8, 14, 17, 21, 32, 36, 41, 45, 51, 55, 64, 69, 75]
7	[3, 6, 15, 22, 30, 37, 40, 50, 74]
5	[5, 35, 47, 79]
4	[9, 25, 63, 78]
9	[11, 27, 28, 60, 76]
11	[12, 18, 29, 56, 62]
2	[16, 26]
13	[44]
19	[48]
15	[52]
17	[59]
8	[65]
24	[66]
23	[68]
10	[70, 73, 80]
25	[82]

Tabela 4.4: *Distribuição dos servidores em névoa no conjunto de dados de treinamento de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado.*

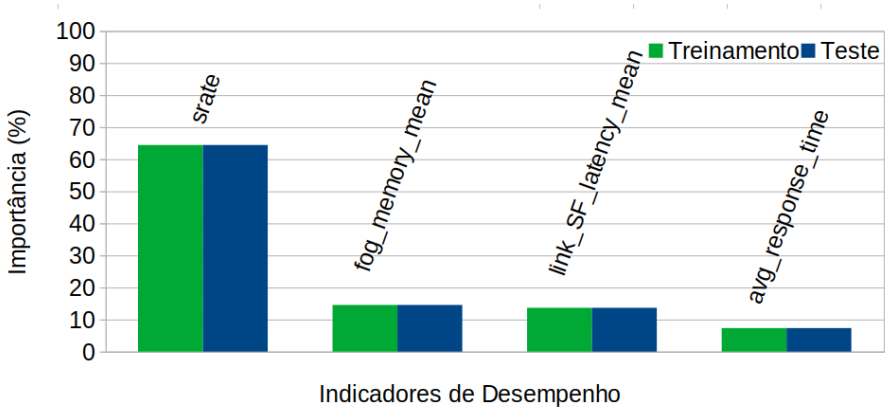
Outra análise realizada foi sobre a importância dos indicadores de desempenho para os modelos considerados. As Figuras 4.8c, 4.9c e 4.10c resumem as importâncias para Árvore de Decisão, Random Forest e XGBoost, respectivamente. No geral, os indicadores que mais contribuíram para os modelos foram relacionadas à rede, como srate (taxa dos pacotes fonte-destino por segundo), latência média do link, desvio padrão e tempo médio



(a) *Treinamento.*

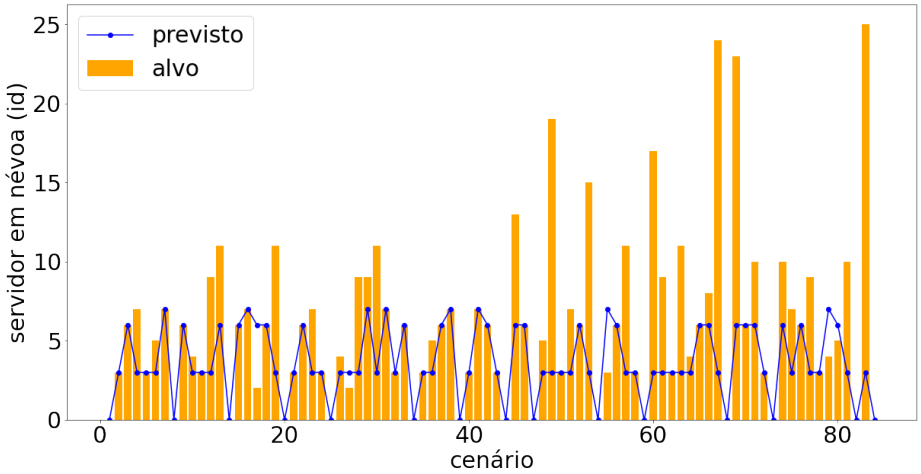


(b) *Teste.*

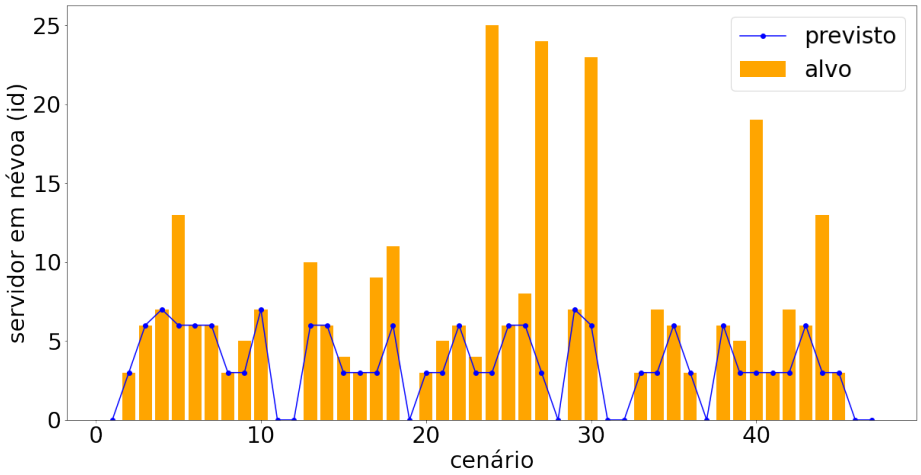


(c) *Importância.*

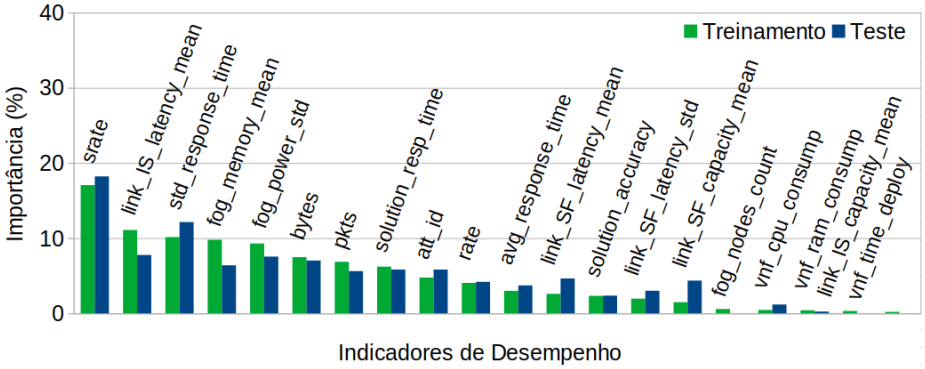
Figura 4.8: *Desempenho e importância dos indicadores para Árvore de Decisão de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado.*



(a) *Treinamento.*

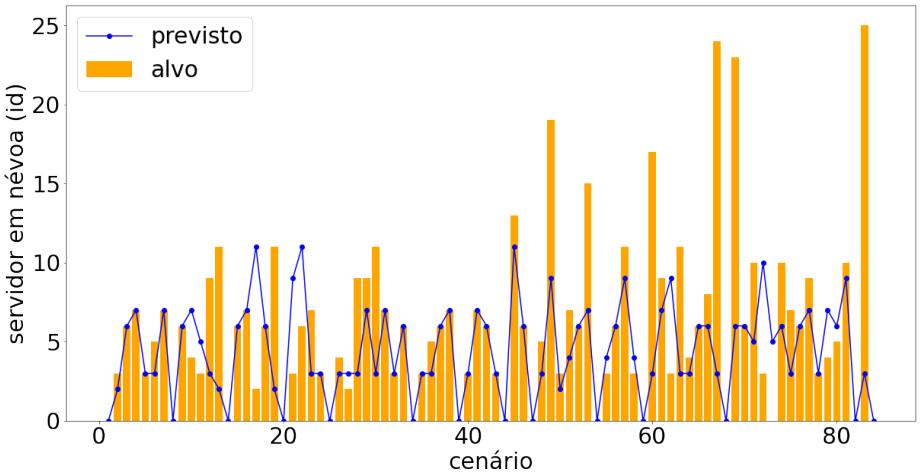


(b) *Teste.*

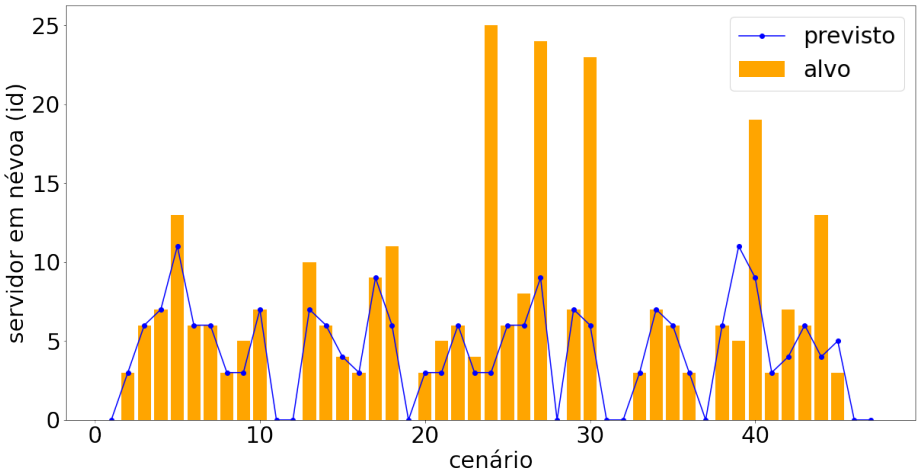


(c) *Importância.*

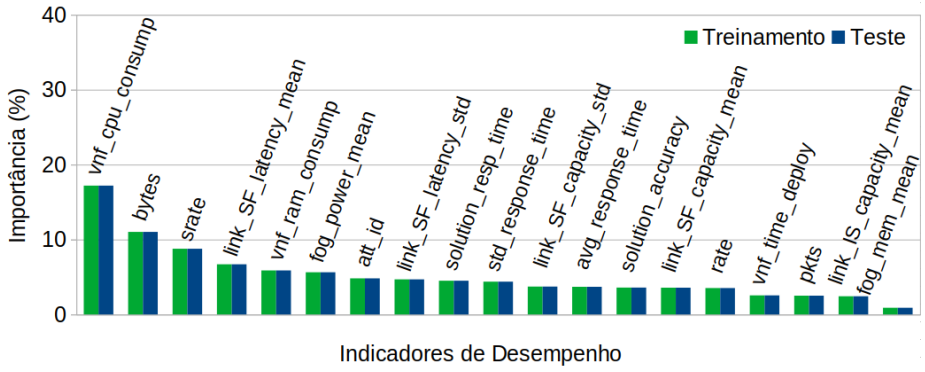
Figura 4.9: *Desempenho e importância dos indicadores para Random Forest de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado.*



(a) *Treinamento.*



(b) *Teste.*



(c) *Importância.*

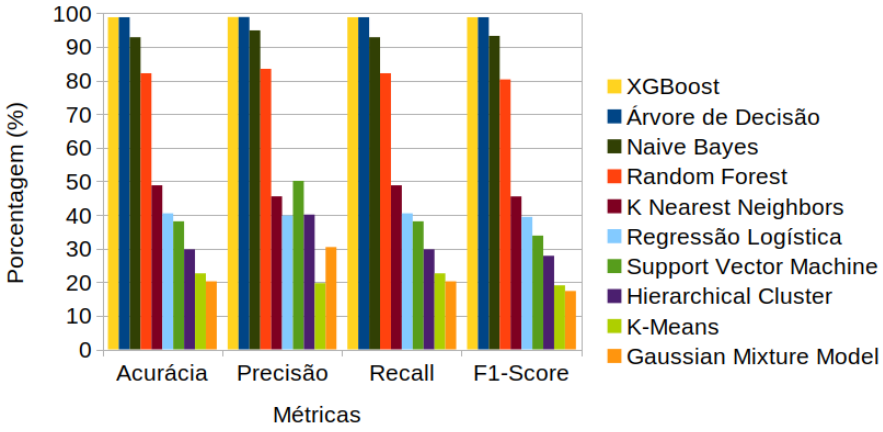
Figura 4.10: *Desempenho e importância dos indicadores para XGBoost de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado.*

de resposta da rede, bytes (número total de bytes na transação) e relacionados à NFV, como consumo de CPU.

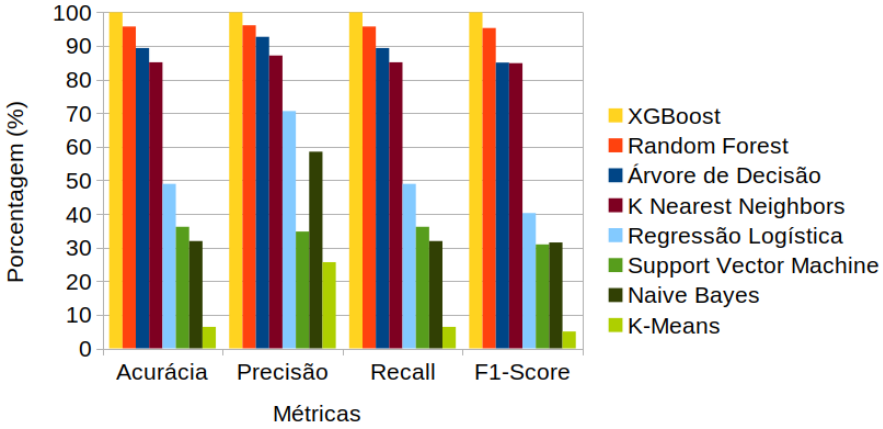
Posicionamento da VNF de Segurança Próximo ao Dispositivo Atacante

Outra estratégia de posicionamento é instanciar a VNF de segurança próximo ao dispositivo atacante, como ilustra a Figura 4.6b. Nessa linha, os modelos de ML foram avaliados de acordo com a classificação do posicionamento da VNF de segurança especificamente em servidores em névoa próximos aos invasores. A Figura 4.11 resume o desempenho dos modelos. Quando comparado com a estratégia de posicionamento anterior, é possível notar uma melhora significativa em todas as métricas: acurácia, precisão, *recall* e F1-score. Esse aumento é de cerca de 19% no geral para dados de treinamento e teste.

Os modelos não supervisionados seguem o mesmo comportamento dos experimentos anteriores, ou seja, apresentam baixo desempenho em todas as métricas (Figuras A.12, A.13 e A.14). Em média, eles obtiveram 24% de acurácia, uma queda significativa em relação aos modelos supervisionados que obtiveram uma média de 71% de acurácia.



(a) *Treinamento.*



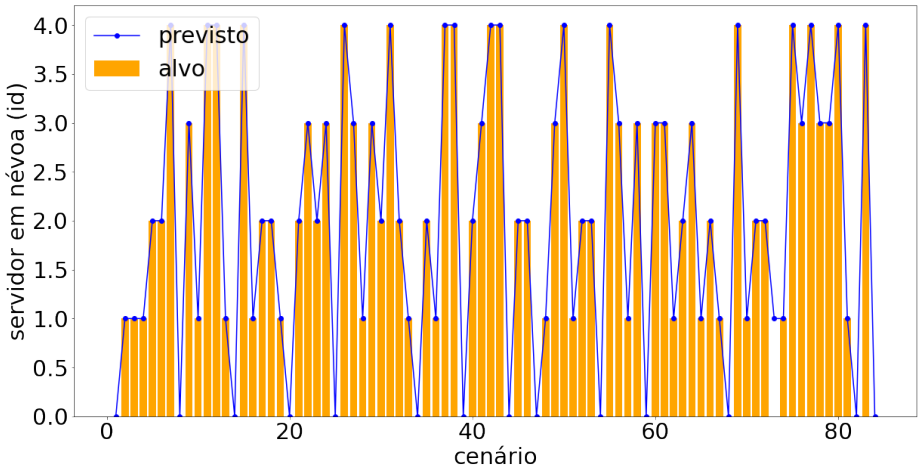
(b) *Teste.*

Figura 4.11: *Desempenho dos modelos de ML de acordo com o posicionamento da VNF de segurança próximo ao atacante.*

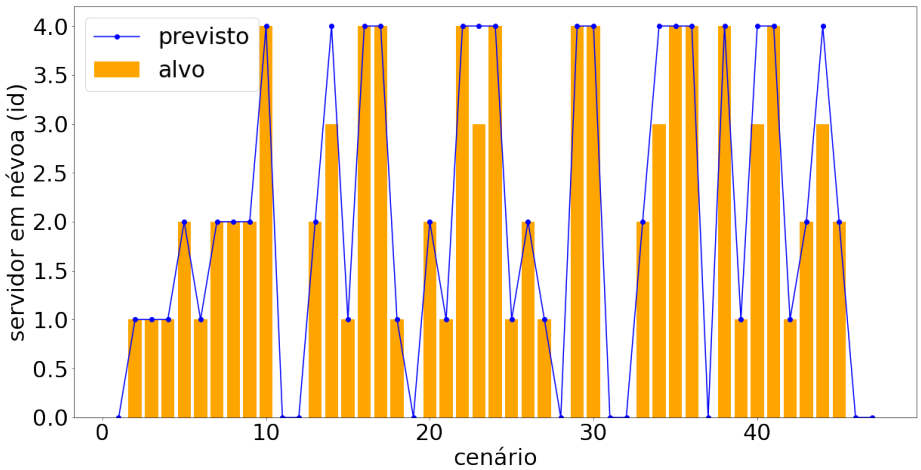
É possível destacar os modelos supervisionados XGBoost, Árvore de Decisão e Random Forest. Eles apresentaram, respectivamente, acurácias médias de 99,40%, 94,08% e 88,94% para as etapas de treinamento e teste. As Figuras 4.12c, 4.13c e 4.14c ilustram a importância dos indicadores de desempenho para eles. Os indicadores que mais contribuíram para os modelos foram relacionados à rede e NFV. Para o XGBoost, modelo com o melhor desempenho, os recursos que mais se destacaram foram o consumo de CPU e o tempo de implantação da VNF, com uma importância total de cerca de 22%. Além disso, as características relacionadas à rede que mais influenciaram o modelo foram srate, média de capacidade do link, desvio padrão e tempo médio de resposta da rede. Esse comportamento que também é observado na estratégia de posicionamento anterior.

Um fator que contribui positivamente para o desempenho alcançado pelos modelos é o número reduzido de servidores em névoa nos quais a VNF de segurança pode ser instanciada. Nesse caso, como as instâncias de ataque foram extraídas do BoT-IoT, apenas quatro invasores estão habilitados a gerar tráfego DDoS. Consequentemente, apenas quatro servidores em névoa foram habilitados para instanciar a VNF de segurança. As Figuras 4.12, 4.13 e 4.14 ilustram em detalhes como os três melhores modelos se comportam.

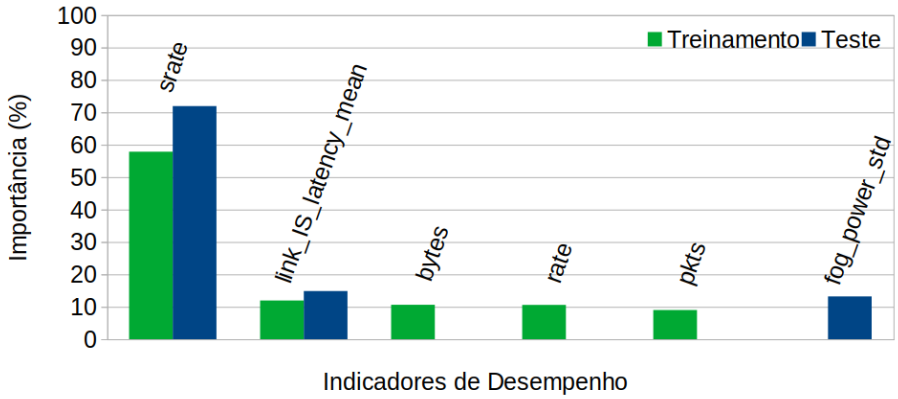
Esses modelos apresentam um desempenho satisfatório para o problema analisado. De acordo com os gráficos e também para a métrica de *recall* (Equação 4.3), os modelos não deixam de instanciar a VNF de segurança quando ataques DDoS estão presentes na rede. Os erros cometidos pelos modelos estão relacionados à instanciação equivocada quando não há ataques na rede, ou seja, falsos positivos. O problema, neste caso, seriam os falsos negativos, pois fariam com que os modelos não instanciassem a VNF de segurança em cenários sob ataques, algo que não é observado nos experimentos.



(a) *Treinamento.*

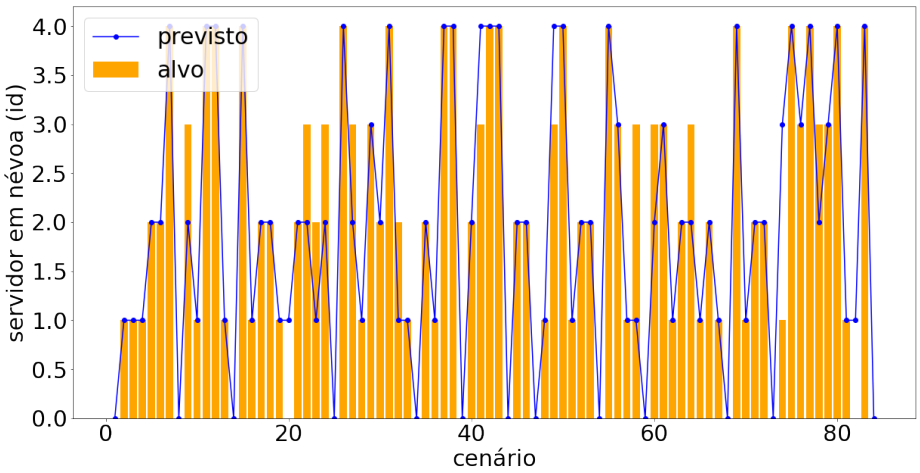


(b) *Teste.*

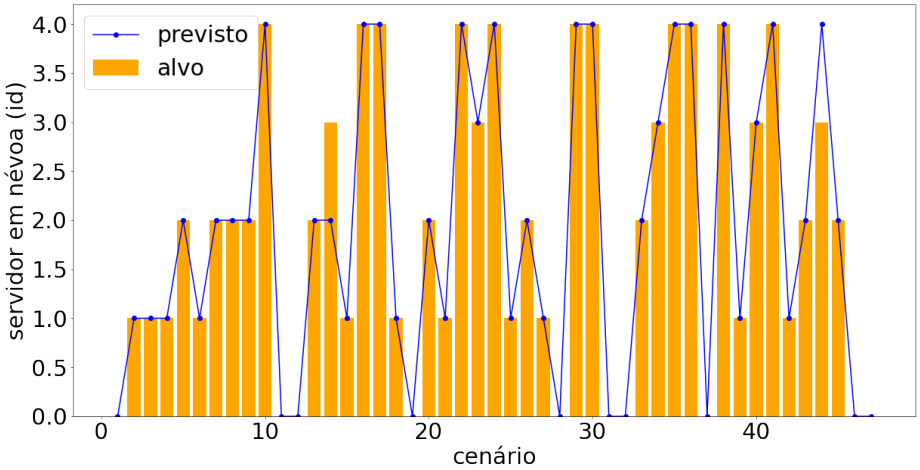


(c) *Importância.*

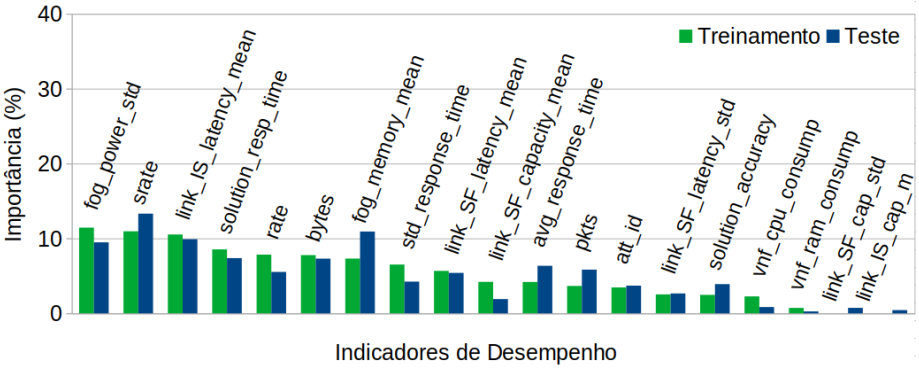
Figura 4.12: *Desempenho e importância dos indicadores para Árvore de Decisão de acordo com o posicionamento da VNF de segurança próximo ao atacante.*



(a) *Treinamento.*

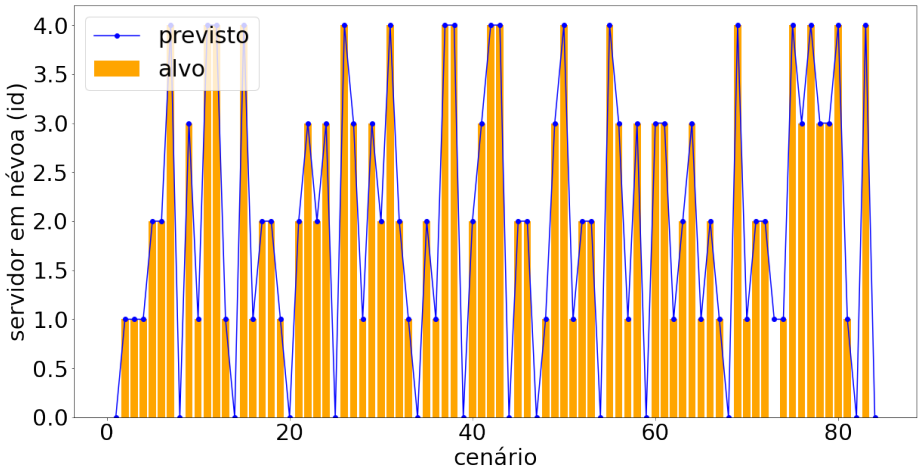


(b) *Teste.*

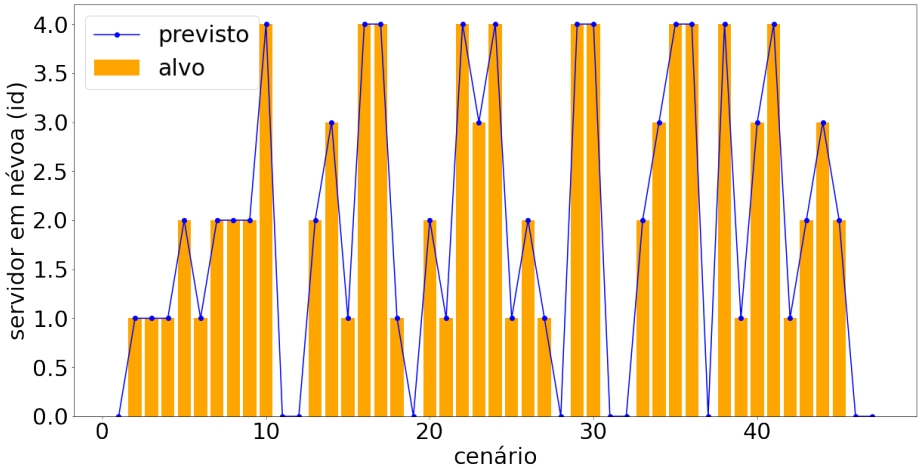


(c) *Importância.*

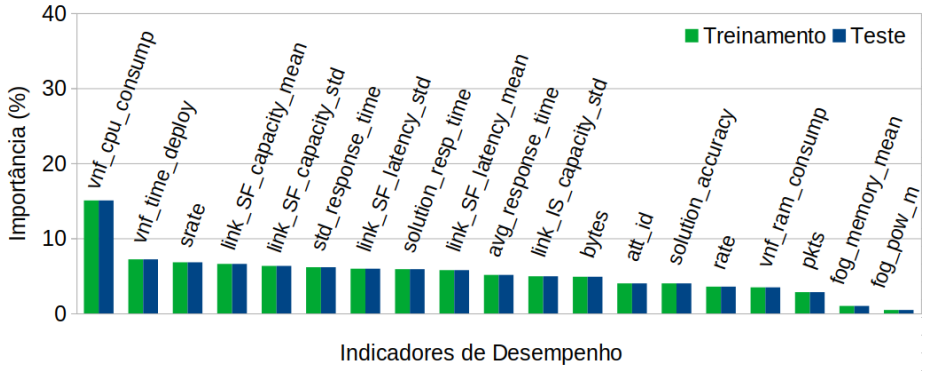
Figura 4.13: *Desempenho e importância dos indicadores para Random Forest de acordo com o posicionamento da VNF de segurança próximo ao atacante.*



(a) *Treinamento.*



(b) *Teste.*



(c) *Importância.*

Figura 4.14: *Desempenho e importância dos indicadores para XGBoost de acordo com o posicionamento da VNF de segurança próximo ao atacante.*

Capítulo 5

Conclusão

A introdução de recursos de rede virtualizados no ecossistema IoT traz diversas vantagens que aumentam o valor agregado da rede, principalmente pelo desacoplamento de funções de segurança de dispositivos com capacidades limitadas. Além disso, a escalabilidade é um dos aspectos mais fortes da virtualização quando comparada a recursos estáticos em uma rede tradicional, permitindo, assim, a alocação dinâmica de funções de rede de acordo com a demanda do contexto, como o volume de tráfego, quantidade de dispositivos, ataques, entre outros.

Aprimorar os mecanismos de segurança levando em consideração a heterogeneidade de dispositivos IoT é um grande desafio a ser encarado e a NFV tem se mostrado uma alternativa viável para auxiliar no cumprimento dos requisitos de segurança em redes IoT. Um dos requisitos apresentados nesta dissertação está relacionado ao gerenciamento e à orquestração de recursos da rede de acordo com sua demanda.

A análise da literatura (Capítulo 3) permitiu identificar que os indicadores designados para avaliar o desempenho da virtualização de funções de redes são menos explorados que os indicadores utilizados para demonstrar a eficácia dos métodos propostos pelos diversos autores. Um exemplo claro disso é o tempo aferido para a implantação de uma ou mais máquinas virtuais que é encontrado apenas em dois dos trabalhos que analisam o desempenho da abordagem proposta. Outro ponto importante é que o desempenho da rede também é pouco explorado, mesmo em trabalhos que consideram a implementação de uma arquitetura em névoa. O desempenho de rede é observado pelos autores em apenas dois dos trabalhos analisados. Avaliar o comportamento de recursos virtualizados por meio dessas métricas é essencial, pois impacta diretamente no gerenciamento de desempenho da rede e se torna ainda mais importante quando funções de segurança são implementadas por meio da NFV.

Com base nas leituras dos trabalhos relacionados, foram propostas técnicas inspiradas em uma abordagem baseada no desempenho do contexto de rede para mitigar ataques DDoS na IoT, investigando como as métricas - ou indicadores - de desempenho podem compor, como variáveis de decisão, modelos de ML para o posicionamento adequado de VNF (Capítulo 4). Essas técnicas validaram a hipótese de que existem vantagens em implementar estratégias baseadas em ML para mitigar ataques DDoS em redes IoT, utilizando como

variáveis de decisão os indicadores de desempenho do próprio recurso NFV e de recursos da rede.

A primeira técnica prevê o tempo de resposta da rede IIoT para diferentes cenários em um contexto SDN-Fog. Regressão Linear, Regressão Linear com PCA, XGBoost e XGBoost com PCA foram os modelos de ML considerados. Experimentos mostraram que a previsão do tempo de resposta dos cenários foi próxima aos tempos obtidos pela resolução do problema. Ou seja, quando comparado a Herrera *et al.* (2021), os experimentos apresentaram uma média menor que 50 milissegundos de variação de acordo com o melhor modelo, o XGBoost. Além disso, a abordagem baseada em modelos de ML foi executada em menos de um segundo, tanto para a etapa de treinamento quanto para a etapa de teste, contra possibilidades de execução de várias horas pela proposta de Herrera *et al.* (2021).

A segunda técnica usa modelos de ML para posicionar VNFs de segurança na rede IIoT, considerando indicadores de desempenho de NFV, da rede e de soluções propostas na literatura. Experimentos realizados mostraram que os modelos supervisionados que implementam estruturas baseadas em árvore (XGBoost, Árvore de Decisão e Random Forest) são os mais adequados para o cenário de rede proposto, alcançando acurácias de 99,40%, 94,08% e 88,94%, respectivamente. De fato, os indicadores de desempenho relacionados à solução são importantes quando o objetivo de uma VNF de segurança é mitigar ataques em redes IoT. No entanto, eles não são únicos. É necessário considerar outros indicadores relacionados à rede e NFV para compor soluções mais eficientes. Além disso, posicionar VNFs de segurança perto do invasor demonstra ser a melhor estratégia para o cenário de rede IIoT apresentado aqui. Não só pelo desempenho dos modelos apresentados, mas também por não permitir que ocorra uma grande inundação de pacotes de rede, ao contrário de posicioná-las próximo ao dispositivo IoT. Em outras palavras, quanto mais longe a mitigação do ataque DDoS estiver do invasor, mais pacotes maliciosos trafegam pela rede até chegarem ao alvo.

Outro fator a ser destacado é a utilização de modelos de aprendizado supervisionado e não supervisionado, tanto para prever o tempo de resposta da rede quanto para classificar o posicionamento de VNFs. Ambos experimentos mostraram que modelos baseados no aprendizado não supervisionado não são os mais adequados para o cenário IIoT considerado, e isso se deve pela não rotulação dos servidores em névoa que são os alvos para a correta classificação dos modelos.

Por fim, esta dissertação contribui para a demonstração de como os indicadores de desempenho do contexto de redes IIoT podem ser utilizados para treinar e testar modelos de ML, seja para a previsão de variáveis contínuas relacionadas a rede, como o tempo de resposta, seja para a classificação inerente ao posicionamento de VNFs em servidores em névoa para a mitigação de ataques DDoS, algo pouco explorado pela atual literatura. A lista a seguir apresenta todos os itens que compõem a contribuição desta dissertação:

- Identificação das abordagens da literatura em que a NFV tem sido utilizada para detectar e mitigar ataques na IoT;
- Análise de como as métricas - ou indicadores - de desempenho são aferidos nas abordagens encontradas;
- Avaliação de diferentes composições de métricas - ou indicadores - de desempenho

como variáveis de decisão em diferentes modelos de ML para mitigar ataques DDoS na IoT, de acordo com o posicionando de VNF;

- Disponibilização de um algoritmo de posicionamento de VNF de segurança baseado em ML;
- Publicização da implementação em Python de modelos de ML juntamente com o conjunto de dados correspondente a redes IIoT sob ataques DDoS.

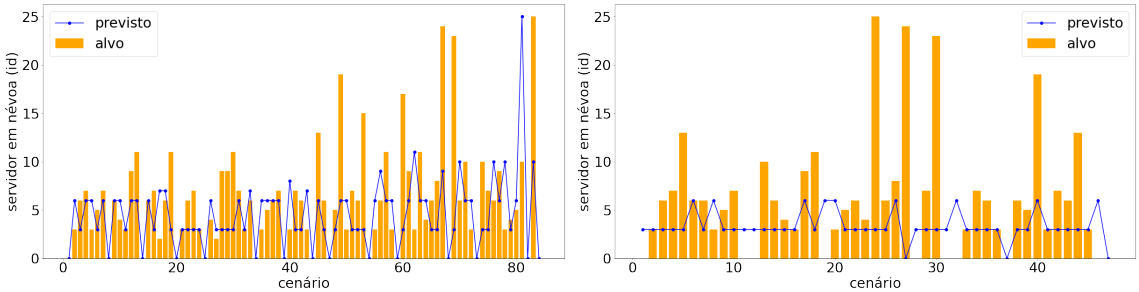
Como trabalhos futuros, busca-se a extensão da exploração dos modelos de ML para incluir os demais paradigmas de aprendizado: semi-supervisionado e aprendizado por reforço. Além disso, outros modelos computacionais poderão ser considerados para a realização do aprendizado, como redes neurais. Pretende-se, também, analisar indicadores de desempenho específicos de redes 5/6G e suas composições, tais como: tipos de dispositivos de rede e seus protocolos de comunicação, frequência e intensidade de sinal, topologias de larga escala, entre outros. Como esta dissertação foca em ataques DDoS a dispositivos IoT alvos, uma próxima análise do mecanismo de posicionamento seria considerar dispositivos IoT como meios, ou seja, dispositivos infectados que atuam como agentes em ataques. Outra abordagem relevante seria a distribuição dos modelos de ML em uma rede baseada em *blockchain*. Rede na qual, a decisão do posicionamento de uma VNF de segurança se daria pelo consenso entre diferentes blocos contidos nela.

Apêndice A

Modelos de Aprendizado de Máquina com Baixo Desempenho

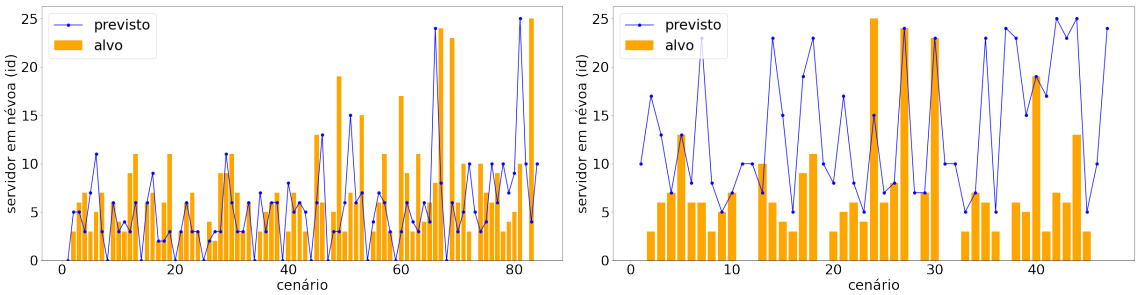
A.1 Posicionamento da VNF de Segurança Próximo ao Dispositivo IoT Atacado

A.1.1 Modelos de Aprendizado Supervisionado



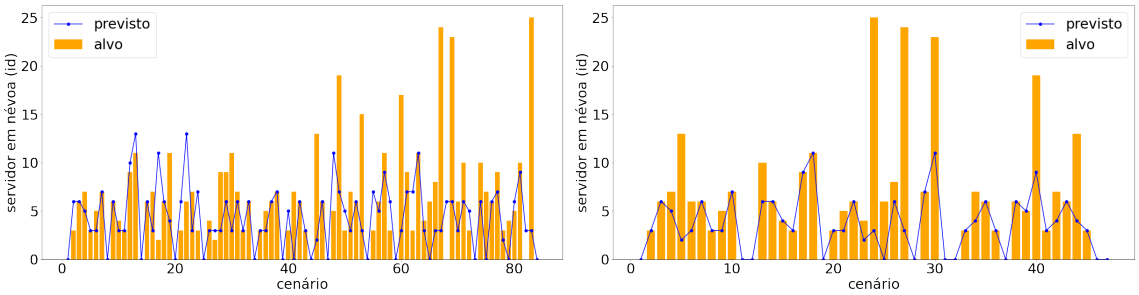
(a) *Treinamento.* **(b)** *Teste.*

Figura A.1: *Desempenho da Regressão Logística de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado.*



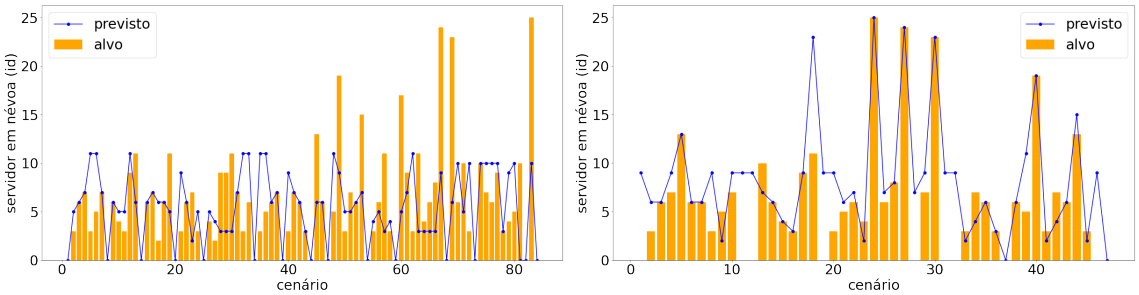
(a) *Treinamento.* **(b)** *Teste.*

Figura A.2: *Desempenho da Support Vector Machine de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado.*



(a) *Treinamento.* **(b)** *Teste.*

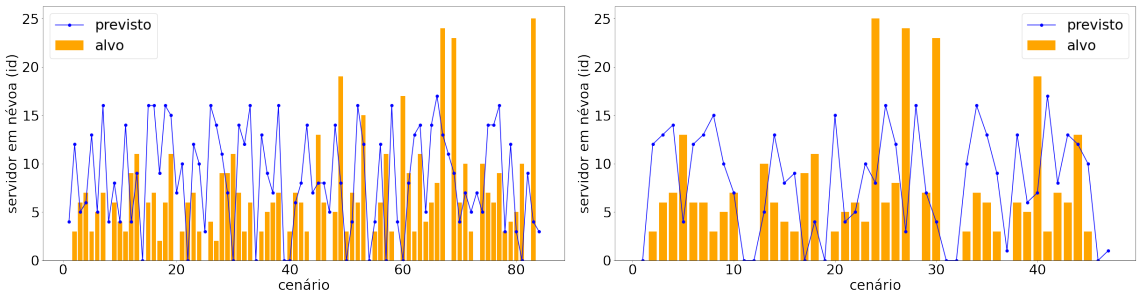
Figura A.3: *Desempenho do K Nearest Neighbors de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado.*



(a) *Treinamento.* **(b)** *Teste.*

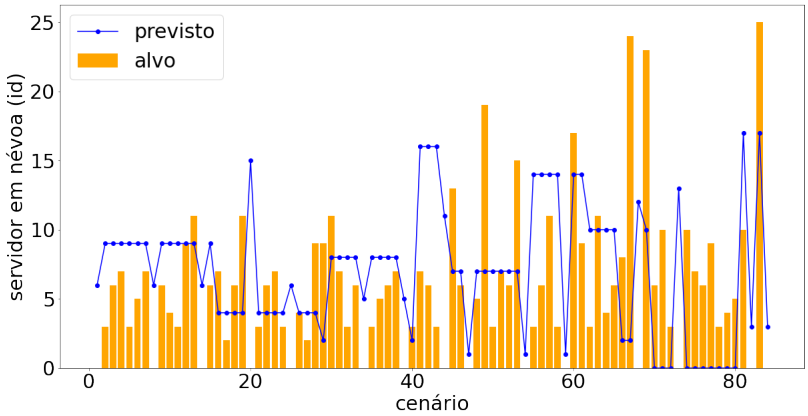
Figura A.4: *Desempenho do Naive Bayes de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado.*

A.1.2 Modelos de Aprendizado Não Supervisionado



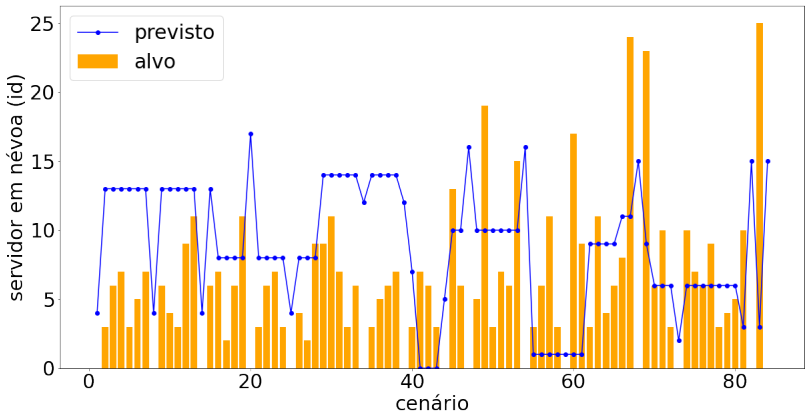
(a) *Treinamento.* **(b)** *Teste.*

Figura A.5: *Desempenho do K-Means de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado.*



(a) *Treinamento.*

Figura A.6: *Desempenho do Hierarchical Cluster de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado.*

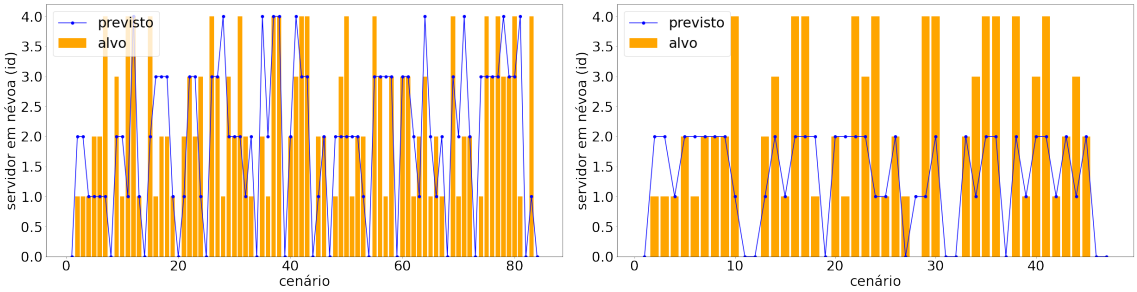


(a) *Treinamento.*

Figura A.7: *Desempenho do Gaussian Mixture Model de acordo com o posicionamento da VNF de segurança próximo ao dispositivo IoT atacado.*

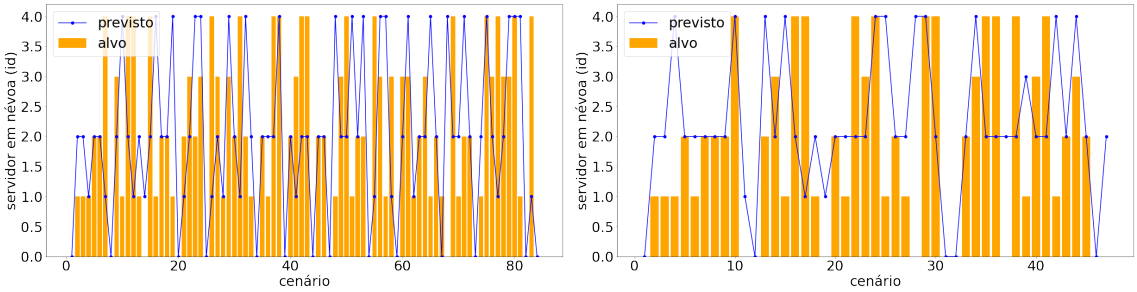
A.2 Posicionamento da VNF de Segurança Próximo ao Dispositivo Atacante

A.2.1 Modelos de Aprendizado Supervisionado



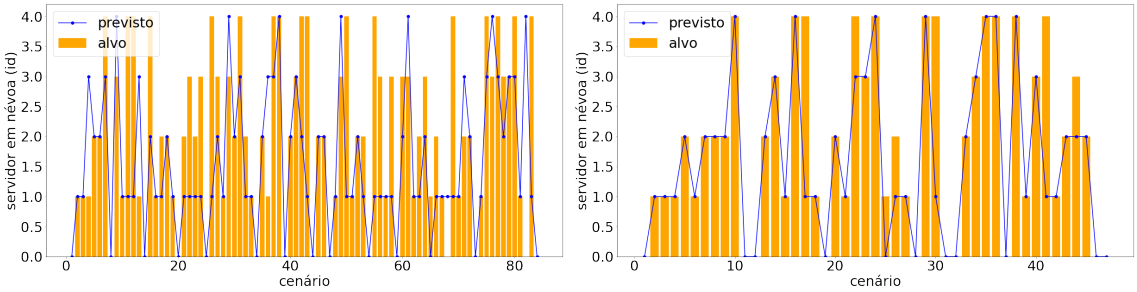
(a) *Treinamento.* **(b)** *Teste.*

Figura A.8: *Desempenho da Regressão Logística de acordo com o posicionamento da VNF de segurança próximo ao atacante.*



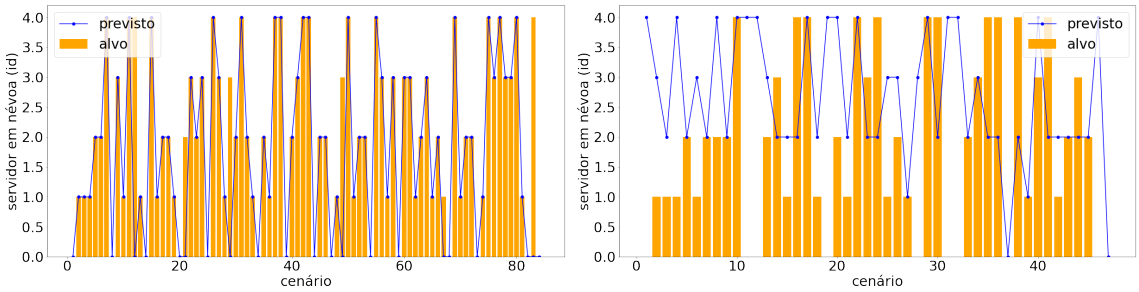
(a) *Treinamento.* **(b)** *Teste.*

Figura A.9: *Desempenho da Support Vector Machine de acordo com o posicionamento da VNF de segurança próximo ao atacante.*



(a) *Treinamento.* **(b)** *Teste.*

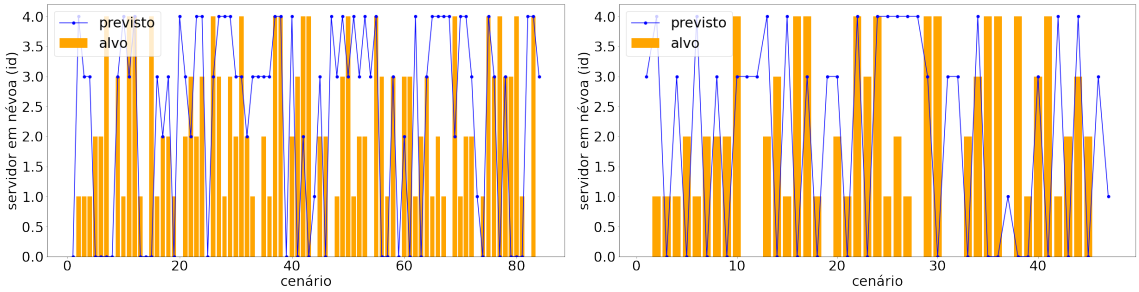
Figura A.10: *Desempenho do K Nearest Neighbors de acordo com o posicionamento da VNF de segurança próximo ao atacante.*



(a) *Treinamento.* **(b)** *Teste.*

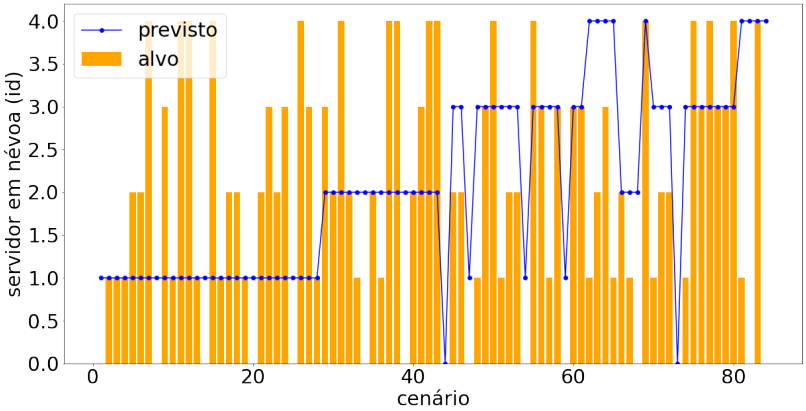
Figura A.11: *Desempenho do Naive Bayes de acordo com o posicionamento da VNF de segurança próximo ao atacante.*

A.2.2 Modelos de Aprendizado Não Supervisionado



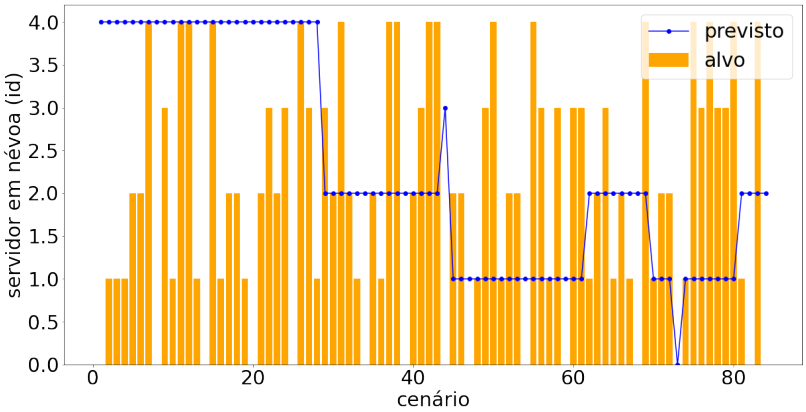
(a) *Treinamento.* **(b)** *Teste.*

Figura A.12: *Desempenho do K-Means de acordo com o posicionamento da VNF de segurança próximo ao atacante.*



(a) *Treinamento.*

Figura A.13: *Desempenho do Hierarchical Cluster de acordo com o posicionamento da VNF de segurança próximo ao atacante.*



(a) *Treinamento.*

Figura A.14: *Desempenho do Gaussian Mixture Model de acordo com o posicionamento da VNF de segurança próximo ao atacante.*

Referências

[5G-Americas 2022] 5G-Americas. *World Nears Three Fourths of a Billion 5G Connections*. Disponível em: https://www.5gamericas.org/world-nears-three-fourths-of-a-billion-5g-connections. Acessado em 25 de Junho de 2022. 2022 (citado na pg. 1).

[Afek *et al.* 2020] Yehuda Afek *et al.* "NFV-based IoT Security for Home Networks using MUD". Em: *Anais do NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*. 2020, pgs. 1–9. doi: 10.1109/NOMS47738.2020.9110329 (citado nas pgs. 31, 35–37).

[Akyildiz *et al.* 2020] Ian F. Akyildiz, Ahan Kak e Shuai Nie. "6g and beyond: the future of wireless communications systems". Em: *IEEE Access* 8 (2020), pgs. 133995–134030. doi: 10.1109/ACCESS.2020.3010896 (citado nas pgs. 1, 3, 17).

[Alam *et al.* 2020] Iqbal Alam *et al.* "A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV". Em: *ACM Computing Surveys* 53.2 (2020), pgs. 1–40. doi: 10.1145/3379444 (citado nas pgs. 1, 2, 8, 14).

[Alsharif *et al.* 2020] Mohammed Alsharif, Anabi Hilary, Khalid Yahya e Shehzad Chaudhry. "Machine learning algorithms for smart data analysis in internet of things environment: taxonomies and research trends". Em: *Symmetry* 12 (jan. de 2020), pg. 88. doi: 10.3390/sym12010088 (citado na pg. 44).

[Bagaa *et al.* 2020] Miloud Bagaa, Tarik Taleb, Jorge Bernal Bernabe e Antonio Skarmeta. "A Machine Learning Security Framework for Iot Systems". Em: *IEEE Access* 8 (2020), pgs. 114066–114077. doi: 10.1109/ACCESS.2020.2996214 (citado nas pgs. 26, 27, 35–37, 52).

[Batista *et al.* 2016] Daniel Macêdo Batista *et al.* "InterSCity: Addressing Future Internet research challenges for Smart Cities". Em: *Anais da 7th International Conference on the Network of the Future (NOF)*. 2016, pgs. 1–6 (citado na pg. 1).

[Bhulai *et al.* 2007] Sandjai Bhulai, Swaminathan Sivasubramanian, Rob van der Mei e Maarten van Steen. "Modeling and predicting end-to-end response times in multi-tier internet applications". Em: *Managing Traffic Performance in Converged Networks*. Ed. por Lorne Mason, Tadeusz Drwiega e James Yan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pgs. 519–532. isbn: 978-3-540-72990-7 (citado nas pgs. 41, 49).

[Boudi *et al.* 2019] Abderrahmane Boudi, Ivan Farris, Miloud Bagaa e Tarik Taleb. "Assessing Lightweight Virtualization for Security-as-a-Service at the Network Edge". Em: *IEICE Transactions on Communications* E102.B.5 (2019), pgs. 970–977. doi: 10.1587/transcom.2018EUI0001 (citado nas pgs. 29, 35–37, 39, 52).

[Boutaba *et al.* 2018] R. Boutaba, M.A. Salahuddin e N. Limam. "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities". Em: *Journal of Internet Services and Applications* 9.16 (2018). doi: 10.1186/s13174-018-0087-2 (citado nas pgs. 3, 17).

[Bremler-Barr *et al.* 2014] Anat Bremler-Barr, Yotam Harchol, David Hay e Yaron Koral. "Deep Packet Inspection as a Service". Em: *Anais da 10th ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT '14)*. Association for Computing Machinery, 2014, pgs. 271–282. isbn: 9781450332798. doi: 10.1145/2674005.2674984. url: https://doi.org/10.1145/2674005.2674984 (citado na pg. 15).

[Bülbül e Fischer 2020] Nurefşan Sertbaş Bülbül e Mathias Fischer. "SDN/NFV-based DDoS Mitigation via Pushback". Em: *Anais da ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. 2020, pgs. 1–6. doi: 10.1109/ICC40277.2020.9148717 (citado na pg. 17).

[Cao *et al.* 2015] Lianjie Cao, Puneet Sharma, Sonia Fahmy e Vinay Saxena. "NFV-VITAL: A Framework for Characterizing the Performance of Virtual Network Functions". Em: *Anais da 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*. 2015, pgs. 93–99. doi: 10.1109/NFV-SDN.2015.7387412 (citado na pg. 15).

[CheckPoint 2020] CheckPoint. *Keeping the gate locked on your IoT devices: Vulnerabilities found on Amazon's Alexa*. Disponível em: https://research.checkpoint.com/2020/amazons-alexa-hacked. Acessado em 14 de Junho de 2021. 2020. url: https://research.checkpoint.com/2020/amazons-alexa-hacked (citado na pg. 2).

[Chen e Guestrin 2016] Tianqi Chen e Carlos Guestrin. "Xgboost: a scalable tree boosting system". Em: KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pgs. 785–794. isbn: 9781450342322. doi: 10.1145/2939672.2939785. url: https://doi.org/10.1145/2939672.2939785 (citado na pg. 44).

[Chi *et al.* 2019] Zicheng Chi, Yan Li, Hongyu Sun, Yao Yao e Ting Zhu. "Concurrent Cross-Technology Communication among Heterogeneous IoT Devices". Em: *IEEE/ACM TON* 27.3 (2019), pgs. 932–947. doi: 10.1109/TNET.2019.2908754 (citado na pg. 1).

[Choudhury *et al.* 2019] Bikash Choudhury, Subhrabrata Choudhury e Animesh Dutta. "A proactive context-aware service replication scheme for adhoc iot scenarios". Em: *IEEE Transactions on Network and Service Management* 16.4 (2019), pgs. 1797–1811. doi: 10.1109/TNSM.2019.2928698 (citado na pg. 40).

[Cloudflare 2021] Cloudflare. *Cloudflare thwarts 17.2M rps DDoS attack — the largest ever reported.* Disponível em: https://blog.cloudflare.com/cloudflare-thwarts-17-2m-rps-ddos-attack-the-largest-ever-reported. Acessado em 18 de Outubro de 2021. 2021 (citado na pg. 2).

[Das *et al.* 2020] Tamal Das, Vignesh Sridharan e Mohan Gurusamy. "A survey on controller placement in sdn". Em: *IEEE Communications Surveys Tutorials* 22.1 (2020), pgs. 472–503. doi: 10.1109/COMST.2019.2935453 (citado na pg. 40).

[Deogirikar e Vidhate 2017] Jyoti Deogirikar e Amarsinh Vidhate. "Security Attacks in IoT: A Survey". Em: *Anais da International Conference on IoT in Social, Mobile, Analytics and Cloud.* 2017, pgs. 32–37. doi: 10.1109/I-SMAC.2017.8058363 (citado nas pgs. 20, 22).

[Ericsson 2022] Ericsson. *6G – Connecting a cyber-physical world.* Disponível em: https://www.ericsson.com/en/reports-and-papers/white-papers/a-research-outlook-towards-6g. Acessado em 25 de Junho de 2022. 2022 (citado na pg. 1).

[ETSI 2014a] ETSI. *Network Functions Virtualisation (NFV): NFV Performance & Portability Best Practises.* Disponível em: https://www.etsi.org/deliver/etsi_gs/NFV-PER/001_099/001/01.01.01_60/gs_NFV-PER001v010101p.pdf. Acessado em 13 de Maio de 2021. 2014 (citado na pg. 12).

[ETSI 2014b] ETSI. *Network Functions Virtualisation (NFV): Service Quality Metrics.* Disponível em: https://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/010/01.01.01_60/gs_NFV-INF010v010101p.pdf. Acessado em 13 de Maio de 2021. 2014 (citado na pg. 12).

[ETSI 2022] ETSI. *Network Functions Virtualisation (NFV): Management and Orchestration Architectural Framework Specification.* Disponível em: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/006/03.06.01_60/gs_NFV006v030601p.pdf. Acessado em 03 de Setembro de 2022. 2022 (citado na pg. 6).

[Farris, Bernabe *et al.* 2017] Ivan Farris, Jorge Bernal Bernabe *et al.* "Towards provisioning of SDN/NFV-based security enablers for integrated protection of IoT systems". Em: *Anais da 2017 IEEE Conference on Standards for Communications and Networking (CSCN).* 2017, pgs. 169–174. doi: 10.1109/CSCN.2017.8088617 (citado nas pgs. 24, 25, 27, 35–37).

[Farris, Taleb *et al.* 2019] Ivan Farris, Tarik Taleb, Yacine Khettab e Jaeseung Song. "A Survey on Emerging SDN and NFV Security Mechanisms for IoT Systems". Em: *IEEE COMST* 21.1 (2019), pgs. 812–837. doi: 10.1109/COMST.2018.2862350 (citado nas pgs. 2, 5, 14, 15).

[Friedman 2001] Jerome H. Friedman. "Greedy function approximation: a gradient boosting machine". Em: *Annals of statistics* (2001), pgs. 1189–1232 (citado na pg. 44).

[K. Gao *et al.* 2020] Kaifeng Gao *et al.* "Julia language in machine learning: algorithms, applications, and open issues". Em: *Computer Science Review* (mai. de 2020), pgs. 100254–100259. doi: 10.1016/j.cosrev.2020.100254 (citado na pg. 44).

[Y. Gao *et al.* 2013] Yang Gao *et al.* "Analysis of security threats and vulnerability for cyber-physical systems". Em: *Anais da 2013 3rd International Conference on Computer Science and Network Technology*. 2013, pgs. 50–55. doi: 10.1109/ICCSNT.2013.6967062 (citado na pg. 20).

[Gember-Jacobson *et al.* 2014] Aaron Gember-Jacobson *et al.* "OpenNF: Enabling Innovation in Network Function Control". Em: *Anais da 2014 ACM Conference on SIGCOMM*. Chicago, Illinois, USA, 2014, pgs. 163–174. isbn: 9781450328364. doi: 10.1145/2619239.2626313. url: https://doi.org/10.1145/2619239.2626313 (citado na pg. 16).

[Giaretta *et al.* 2019] Alberto Giaretta, Nicola Dragoni e Fabio Massacci. "IoT Security Configurability with Security-by-Contract". Em: *Sensors* 19.19 (2019). issn: 1424-8220. doi: 10.3390/s19194121 (citado na pg. 10).

[Griffioen e Doerr 2020] Harm Griffioen e Christian Doerr. "Examining mirai's battle over the internet of things". Em: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. CCS '20. Virtual Event, USA: Association for Computing Machinery, 2020, pgs. 743–756. isbn: 9781450370899. doi: 10.1145/3372297.3417277. url: https://doi.org/10.1145/3372297.3417277 (citado na pg. 2).

[Guizani e Ghafoor 2020] Nadra Guizani e Arif Ghafoor. "A Network Function Virtualization System for Detecting Malware in Large IoT Based Networks". Em: *IEEE Journal on Selected Areas in Communications* 38.6 (2020), pgs. 1218–1228. doi: 10.1109/JSAC.2020.2986618 (citado nas pgs. 29, 35–37).

[Gupta *et al.* 2019] Lav Gupta, Tara Salman, Maede Zolanvari, Aiman Erbad e Raj Jain. "Fault and Performance Management in Multi-Cloud Virtual Network Services using AI: A Tutorial and a Case Study". Em: *Computer Networks* 165 (2019), pg. 106950. issn: 1389-1286. doi: https://doi.org/10.1016/j.comnet.2019.106950 (citado nas pgs. 2, 13).

[Hafeez *et al.* 2016] Ibbad Hafeez, Aaron Yi Ding, Lauri Suomalainen, Alexey Kirichenko e Sasu Tarkoma. "Securebox: Toward Safer and Smarter IoT Networks". Em: *Anais do 2016 ACM Workshop on Cloud-Assisted Networking*. Irvine, California, USA, 2016, pgs. 55–60. isbn: 9781450346733. doi: 10.1145/3010079.3012014. url: https://doi.org/10.1145/3010079.3012014 (citado na pg. 16).

[Hardegen *et al.* 2020] Christoph Hardegen, Benedikt Pfülb, Sebastian Rieger e Alexander Gepperth. "Predicting network flow characteristics using deep learning and real-world network traffic". Em: *IEEE Transactions on Network and Service Management* 17.4 (2020), pgs. 2662–2676. doi: 10.1109/TNSM.2020.3025131 (citado na pg. 41).

[Herrera *et al.* 2020] Juan Luis Herrera, Jaime Galán-Jiménez, Javier Berrocal e Juan Manuel Murillo. *Optimizing response time in sdn-edge environments for time-strict iot applications*. 2020. doi: 10.21227/m9s7-1q10. url: https://dx.doi.org/10.21227/m9s7-1q10 (citado nas pgs. 43, 47).

[Herrera *et al.* 2021] Juan Luis Herrera, Jaime Galán-Jiménez, Javier Berrocal e Juan Manuel Murillo. "Optimizing the response time in sdn-fog environments for time-strict iot applications". Em: *IEEE Internet of Things Journal* (2021), pgs. 1–1. doi: 10.1109/JIOT.2021.3077992 (citado nas pgs. 40, 42, 43, 47, 66).

[IDC 2020] IDC. *IoT Growth Demands Rethink of Long-Term Storage Strategies, says IDC*. Disponível em: https://www.idc.com/getdoc.jsp?containerId=prAP46737220. Acessado em 12 de Junho de 2021. 2020. url: https://www.idc.com/getdoc.jsp?containerId=prAP46737220 (citado na pg. 20).

[IETF 1998] IETF. *RFC 2330: Framework for IP Performance Metrics*. Disponível em: https://datatracker.ietf.org/doc/html/rfc2330. Acessado em 13 de Maio de 2021. 1998 (citado na pg. 11).

[IETF 2011] IETF. *RFC 6390: Guidelines for Considering New Performance Metric Development*. Disponível em: https://datatracker.ietf.org/doc/html/rfc6390. Acessado em 13 de Maio de 2021. 2011 (citado na pg. 11).

[IETF 2017a] IETF. *Benchmarking Methodology for Virtualization Network Performance*. Disponível em: https://tools.ietf.org/id/draft-huang-bmwg-virtual-network-performance-03.html. Acessado em 13 de Maio de 2021. 2017 (citado na pg. 12).

[IETF 2017b] IETF. *RFC 8172: Considerations for Benchmarking Virtual Network Functions and Their Infrastructure*. Disponível em: https://datatracker.ietf.org/doc/html/rfc8172. Acessado em 13 de Maio de 2021. 2017 (citado na pg. 13).

[IETF 2020] IETF. *RFC 8520: Manufacturer Usage Description Specification*. Disponível em: https://datatracker.ietf.org/doc/rfc8520. Acessado em 21 de Junho de 2021. 2020 (citado na pg. 31).

[Jain 1991] Raj Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley, 1991 (citado nas pgs. 11, 19).

[Y. Jia *et al.* 2020] Yizhen Jia, Fangtian Zhong, Arwa Alrawais, Bei Gong e Xiuzhen Cheng. "FlowGuard: An Intelligent Edge Defense Mechanism Against IoT DDoS Attacks". Em: *IEEE Internet of Things Journal* 7.10 (2020), pgs. 9552–9562. doi: 10.1109/JIOT.2020.2993782 (citado na pg. 17).

[Z. Jia *et al.* 2018] Ziye Jia *et al.* "Joint optimization of vnf deployment and routing in software defined satellite networks". Em: *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. 2018, pgs. 1–5. doi: 10.1109/VTCFall.2018.8690972 (citado na pg. 49).

[Kaspersky 2019] Kaspersky. *Brasil é o segundo país com mais ataques a dispositivos IoT*. Disponível em: https://www.kaspersky.com.br/about/press-releases/2019_brasil-e-o-segundo-pais-com-mais-ataques-a-dispositivos-iot. Acessado em 9 de Março de 2021. 2019 (citado na pg. 2).

[Kaspersky 2020] Kaspersky. *Quase 30% das empresas que usam IoT sofreram incidentes de segurança*. Disponível em: https://www.kaspersky.com.br/blog/empresas-iot-seguranca-dicas/14869. Acessado em 9 de Março de 2021. 2020 (citado na pg. 2).

[Kim *et al.* 2015] Taekhee Kim, Taehwan Koo e Eunkyoung Paik. "SDN and NFV Benchmarking for Performance and Reliability". Em: *Anais do 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. 2015, pgs. 600–603 (citado na pg. 13).

[Laghrissi e Taleb 2019] Abdelquoddouss Laghrissi e Tarik Taleb. "A Survey on the Placement of Virtual Resources and Virtual Network Functions". Em: *IEEE COMST* 21.2 (2019), pgs. 1409–1434. doi: 10.1109/COMST.2018.2884835 (citado na pg. 2).

[Lin *et al.* 2017] Jie Lin *et al.* "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications". Em: *IEEE Internet of Things Journal* 4.5 (2017), pgs. 1125–1142. doi: 10.1109/JIOT.2017.2683200 (citado nas pgs. 10, 19–22, 40).

[Liu *et al.* 2021] Liang Liu, Songtao Guo, Guiyan Liu e Yuanyuan Yang. "Joint dynamical vnf placement and sfc routing in nfv-enabled sdns". Em: *IEEE Transactions on Network and Service Management* 18.4 (2021), pgs. 4263–4276. doi: 10.1109/TNSM.2021.3091424 (citado nas pgs. 31, 33, 37).

[Massonet *et al.* 2017] Philippe Massonet *et al.* "Security in Lightweight Network Function Virtualisation for Federated Cloud and IoT". Em: *Anais da 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*. 2017, pgs. 148–154. doi: 10.1109/FiCloud.2017.43 (citado nas pgs. 30, 35–37).

[Medhat *et al.* 2017] Ahmed M. Medhat *et al.* "Service Function Chaining in Next Generation Networks: State of the Art and Research Challenges". Em: *IEEE Communications Magazine* 55.2 (2017), pgs. 216–223. doi: 10.1109/MCOM.2016.1600219RP (citado na pg. 11).

[Meneghello *et al.* 2019] Francesca Meneghello, Matteo Calore, Daniel Zucchetto, Michele Polese e Andrea Zanella. "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices". Em: *IEEE Internet of Things Journal* 6.5 (2019), pgs. 8182–8201. doi: 10.1109/JIOT.2019.2935189 (citado nas pgs. 2, 9, 20, 22, 49).

[Mijumbi *et al.* 2016] Rashid Mijumbi *et al.* "Network Function Virtualization: State-of-the-Art and Research Challenges". Em: *IEEE COMST* 18.1 (2016), pgs. 236–262. doi: 10.1109/COMST.2015.2477041 (citado na pg. 2).

[Mirkovic *et al.* 2018] Djuro Mirkovic, Grenville Armitage e Philip Branch. "A survey of round trip time prediction systems". Em: *IEEE Communications Surveys Tutorials* 20.3 (2018), pgs. 1758–1776. doi: 10.1109/COMST.2018.2816917 (citado nas pgs. 41, 49).

[Moghaddam *et al.* 2019] Sara Kardani Moghaddam, Rajkumar Buyya e Kotagiri Ramamohanarao. "Performance-Aware Management of Cloud Resources: A Taxonomy and Future Directions". Em: *ACM Computing Surveys* 52.4 (ago. de 2019). issn: 0360-0300. doi: 10.1145/3337956. url: https://doi.org/10.1145/3337956 (citado nas pgs. 11, 19).

[Montero e Serral-Gracià 2016] Diego Montero e René Serral-Gracià. "Offloading Personal Security Applications to the Network Edge: A Mobile User Case Scenario". Em: *Anais da 2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*. 2016, pgs. 96–101. doi: 10.1109/IWCMC.2016.7577040 (citado na pg. 16).

[Montero, Yannuzzi *et al.* 2015] Diego Montero, Marcelo Yannuzzi *et al.* "Virtualized Security at the Network Edge: a User-Centric Approach". Em: *IEEE Communications Magazine* 53.4 (2015), pgs. 176–186. doi: 10.1109/MCOM.2015.7081092 (citado na pg. 15).

[Moustafa 2019] Nour Moustafa. *The bot-iot dataset*. 2019. doi: 10.21227/r7v2-x988. url: https://dx.doi.org/10.21227/r7v2-x988 (citado na pg. 51).

[Naik 2017] Nitin Naik. "Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP". Em: *Anais do 2017 IEEE International Systems Engineering Symposium (ISSE)*. 2017, pgs. 1–7. doi: 10.1109/SysEng.2017.8088251 (citado na pg. 10).

[Neshenko *et al.* 2019] Nataliia Neshenko, Elias Bou-Harb, Jorge Crichigno, Georges Kaddoum e Nasir Ghani. "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations". Em: *IEEE Communications Surveys Tutorials* 21.3 (2019), pgs. 2702–2733. doi: 10.1109/COMST.2019.2910750 (citado nas pgs. 20, 22).

[Okay e Ozdemir 2018] Feyza Yildirim Okay e Suat Ozdemir. "Routing in fog-enabled iot platforms: a survey and an sdn-based solution". Em: *IEEE Internet of Things Journal* 5.6 (2018), pgs. 4871–4889. doi: 10.1109/JIOT.2018.2882781 (citado na pg. 40).

[Oliveira, Ney *et al.* 2021] Guilherme Werneck de Oliveira, Rodrigo Ney *et al.* "Predicting Response Time in SDN-Fog Environments for IIoT Applications". Em: *Proceedings of the 2021 IEEE Latin-American Conference on Communications (LATINCOM)*. 2021. doi: 10.1109/LATINCOM53176.2021.9647803 (citado nas pgs. 4, 49, 52).

[Oliveira, Porto *et al.* 2021] Guilherme Werneck de Oliveira, Jonathan Rangel Porto *et al.* "Virtualização de Funções de Rede na IoT: Um Panorama do Gerenciamento de Desempenho x Segurança". Em: *Livro de Minicursos do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*. 2021. Cap. 3. doi: 10.5753/sbc.8184.9 (citado na pg. 4).

[Omdia 2021] Omdia. *Capture growing 5G opportunities with the 2022 5G Trends to Watch report*. Disponível em: https://omdia.tech.informa.com/marketing/campaigns/5G-Trends. Acessado em 25 de Junho de 2022. 2021 (citado na pg. 1).

[Pei *et al.* 2020] Jianing Pei, Peilin Hong, Miao Pan, Jiangqing Liu e Jingsong Zhou. "Optimal vnf placement via deep reinforcement learning in sdn/nfv-enabled networks". Em: *IEEE Journal on Selected Areas in Communications* 38.2 (2020), pgs. 263–278. doi: 10.1109/JSAC.2019.2959181 (citado nas pgs. 32, 33, 37).

[Pióro e Medhi 2004] Michał Pióro e Deepankar Medhi. "Chapter 5 - general optimization methods for network design". Em: *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Ed. por Michał Pióro e Deepankar Medhi. The Morgan Kaufmann Series in Networking. San Francisco: Morgan Kaufmann, 2004, pgs. 151–210. doi: https://doi.org/10.1016/B978-012557189-0/50008-1. url: https://www.sciencedirect.com/science/article/pii/B9780125571890500081 (citado na pg. 41).

[Qazi *et al.* 2013] Zafar Ayyub Qazi *et al.* "SIMPLE-Fying Middlebox Policy Enforcement Using SDN". Em: *SIGCOMM Comput. Commun. Rev.* 43.4 (ago. de 2013), pgs. 27–38. issn: 0146-4833. doi: 10.1145/2534169.2486022. url: https://doi.org/10.1145/2534169.2486022 (citado na pg. 16).

[Rafique e Velasco 2018] Danish Rafique e Luis Velasco. "Machine learning for network automation: overview, architecture, and applications [invited tutorial]". Em: *J. Opt. Commun. Netw.* 10.10 (out. de 2018), pgs. D126–D143. doi: 10.1364/JOCN.10.00D126. url: http://jocn.osa.org/abstract.cfm?URI=jocn-10-10-D126 (citado na pg. 44).

[Rosário *et al.* 2014] Denis Rosário, Zhongliang Zhao, Aldri Santos, Torsten Braun e Eduardo Cerqueira. "A Beaconless Opportunistic Routing based on a Cross-layer Approach for Efficient Video Dissemination in Mobile Multimedia IoT Applications". Em: *Computer Communications* 45 (2014), pgs. 21–31. issn: 0140-3664 (citado na pg. 1).

[Sairam *et al.* 2019] Rishi Sairam, Suman Sankar Bhunia, Vijayanand Thangavelu e Mohan Gurusamy. "NETRA: Enhancing IoT Security Using NFV-Based Edge Traffic Analysis". Em: *IEEE Sensors Journal* 19.12 (2019), pgs. 4660–4671. doi: 10.1109/JSEN.2019.2900097 (citado nas pgs. 27, 28, 35–37, 39, 52).

[Schwengber *et al.* 2020] Bruno Henrique Schwengber, Andressa Vergütz, Nelson G. Prates e Michele Nogueira. "A Method Aware of Concept Drift for Online Botnet Detection". Em: *Anais da 2020 IEEE Global Communications Conference (GLOBECOM)*. 2020, pgs. 1–6. doi: 10.1109/GLOBECOM42002.2020.9347990 (citado na pg. 10).

[Al-Shaboti *et al.* 2018] Mohammed Al-Shaboti, Ian Welch, Aaron Chen e Muhammed Adeel Mahmood. "Towards Secure Smart Home IoT: Manufacturer and User Network Access Control Framework". Em: *Anais da 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*. 2018, pgs. 892–899. doi: 10.1109/AINA.2018.00131 (citado nas pgs. 30, 35–37, 52).

[Shih *et al.* 2019] Yuan-Yao Shih, Hsin-Peng Lin, Ai-Chun Pang, Ching-Chih Chuang e Chun-Ting Chou. "An nfv-based service framework for iot applications in edge computing environments". Em: *IEEE TNSM* 16.4 (2019), pgs. 1419–1434. doi: 10.1109/TNSM.2019.2948764 (citado na pg. 49).

[Singh *et al.* 2016] Jatinder Singh, Thomas Pasquier, Jean Bacon, Hajoon Ko e David Eyers. "Twenty Security Considerations for Cloud-Supported Internet of Things". Em: *IEEE Internet of Things Journal* 3.3 (2016), pgs. 269–284. doi: 10.1109/JIOT.2015.2460333 (citado nas pgs. 21, 40).

[Skiena 2017] Steven S. Skiena. *The Data Science Design Manual*. 1. Springer, Cham, 2017, pg. 445. isbn: 978-3-319-55444-0. doi: 10.1007/978-3-319-55444-0 (citado nas pgs. 17, 46).

[Sobin 2020] C.C. Sobin. "A Survey on Architecture, Protocols and Challenges in IoT". Em: *Wireless Personal Communications* 112 (2020), pgs. 1383–1429. doi: https://doi.org/10.1007/s11277-020-07108-5 (citado na pg. 20).

[Song *et al.* 2020] Sooeun Song, Changsung Lee, Hyoungjun Cho, Goeun Lim e Jong-Moon Chung. "Clustered virtualized network functions resource allocation based on context-aware grouping in 5g edge networks". Em: *IEEE Transactions on Mobile Computing* 19.5 (2020), pgs. 1072–1083. doi: 10.1109/TMC.2019.2907593 (citado na pg. 49).

[Sun *et al.* 2018] Wen Sun, Jiajia Liu, Yanlin Yue e Haibin Zhang. "Double auction-based resource allocation for mobile edge computing in industrial internet of things". Em: *IEEE Transactions on Industrial Informatics* 14.10 (2018), pgs. 4692–4701. doi: 10.1109/TII.2018.2855746 (citado na pg. 42).

[V. D. da Silva *et al.* 2021] Marcus de V. D. da Silva, Alexandre Rocha, Rafael L. Gomes e Michele Nogueira. "Lightweight Data Compression for Low Energy Consumption in Industrial Internet of Things". Em: *Anais da IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*. 2021 (citado na pg. 1).

[Varasteh *et al.* 2021] Amir Varasteh, Basavaraj Madiwalar, Amaury Van Bemten, Wolfgang Kellerer e Carmen Mas-Machuca. "Holu: power-aware and delay-constrained vnf placement and chaining". Em: *IEEE Transactions on Network and Service Management* 18.2 (2021), pgs. 1524–1539. doi: 10.1109/TNSM.2021.3055693 (citado nas pgs. 32, 33, 37).

[N. Wang *et al.* 2008] Ning Wang, Kin Hon Ho, George Pavlou e Michael Howarth. "An overview of routing optimization for internet traffic engineering". Em: *IEEE Communications Surveys Tutorials* 10.1 (2008), pgs. 36–56. doi: 10.1109/COMST.2008.4483669 (citado na pg. 40).

[T. Wang *et al.* 2021] Tianfu Wang *et al.* "Drl-sfcp: adaptive service function chains placement with deep reinforcement learning". Em: *Anais da ICC 2021 - 2021 IEEE International Conference on Communications (ICC)*. 2021 (citado na pg. 16).

[Xu *et al.* 2018] Hansong Xu, Wei Yu, David Griffith e Nada Golmie. "A survey on industrial internet of things: a cyber-physical systems perspective". Em: *IEEE Access* 6 (2018), pgs. 78238–78259. doi: 10.1109/ACCESS.2018.2884906 (citado na pg. 40).

[S. Yang *et al.* 2021] Song Yang *et al.* "Delay-aware virtual network function placement and routing in edge clouds". Em: *IEEE Transactions on Mobile Computing* 20.2 (2021), pgs. 445–459. doi: 10.1109/TMC.2019.2942306 (citado na pg. 49).

[Y. Yang *et al.* 2017] Yuchen Yang, Longfei Wu, Guisheng Yin, Lijie Li e Hongbin Zhao. "A Survey on Security and Privacy Issues in Internet-of-Things". Em: *IEEE Internet of Things Journal* 4.5 (2017), pgs. 1250–1258. doi: 10.1109/JIOT.2017.2694844 (citado nas pgs. 20, 22).

[Yousefpour *et al.* 2019] Ashkan Yousefpour *et al.* "All one needs to know about fog computing and related edge computing paradigms: A complete survey". Em: *Journal of Systems Architecture* 98 (2019), pgs. 289–330. ISSN: 1383-7621. DOI: https://doi.org/10.1016/j.sysarc.2019.02.009. URL: https://www.sciencedirect.com/science/article/pii/S1383762118306349 (citado nas pgs. 21, 22, 40).

[Yu *et al.* 2015] Tianlong Yu, Vyas Sekar, Srinivasan Seshan, Yuvraj Agarwal e Chenren Xu. "Handling a Trillion (Unfixable) Flaws on a Billion Devices: Rethinking Network Security for the Internet-of-Things". Em: *Anais do 14th ACM Workshop on Hot Topics in Networks (HotNets-XIV)*. 2015. ISBN: 9781450340472. DOI: 10.1145/2834050.2834095. URL: https://doi.org/10.1145/2834050.2834095 (citado nas pgs. 15, 49).

[Zarca, Bagaa *et al.* 2020] Alejandro Molina Zarca, Miloud Bagaa, Jorge Bernal Bernabe, Tarik Taleb e Antonio Skarmeta. "Semantic-Aware Security Orchestration in SDN/NFV-Enabled IoT Systems". Em: *Sensors* 20.13 (2020). ISSN: 1424-8220. DOI: 10.3390/s20133622. URL: https://www.mdpi.com/1424-8220/20/13/3622 (citado na pg. 2).

[Zarca, Bernabe, Farris *et al.* 2018] Alejandro Molina Zarca, Jorge Bernal Bernabe, Ivan Farris *et al.* "Enhancing IoT security through network softwarization and virtual security appliances". Em: *International Journal of Network Management* 28.5 (2018). e2038 nem.2038, e2038. DOI: https://doi.org/10.1002/nem.2038. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nem.2038. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.2038 (citado nas pgs. 2, 14, 24, 25, 35–37, 52).

[Zarca, Bernabe, Skarmeta *et al.* 2020] Alejandro Molina Zarca, Jorge Bernal Bernabe, Antonio Skarmeta e Jose M. Alcaraz Calero. "Virtual IoT HoneyNets to Mitigate Cyberattacks in SDN/NFV-Enabled IoT Networks". Em: *IEEE Journal on Selected Areas in Communications* 38.6 (2020), pgs. 1262–1277. DOI: 10.1109/JSAC.2020.2986621 (citado nas pgs. 26, 35–37, 39, 52).

[Zarca, Bernabe, Trapero *et al.* 2019] Alejandro Molina Zarca, Jorge Bernal Bernabe, Ruben Trapero *et al.* "Security Management Architecture for NFV/SDN-Aware IoT Systems". Em: *IEEE Internet of Things Journal* 6.5 (2019), pgs. 8005–8020. DOI: 10.1109/JIOT.2019.2904123 (citado nas pgs. 25, 35–37, 52).

[Zarca, Garcia-Carrillo *et al.* 2019] Alejandro Molina Zarca, Dan Garcia-Carrillo *et al.* "Enabling Virtual AAA Management in SDN-Based IoT Networks". Em: *Sensors* 19.2 (2019). ISSN: 1424-8220. DOI: 10.3390/s19020295. URL: https://www.mdpi.com/1424-8220/19/2/295 (citado nas pgs. 25, 26, 35–37, 52).

[Zhang *et al.* 2016] Wei Zhang, Jinho Hwang, Shriram Rajagopalan, K.K. Ramakrishnan e Timothy Wood. "Performance Management Challenges for Virtual Network Functions". Em: *2016 IEEE NetSoft Conference and Workshops*. 2016, pgs. 20–23. DOI: 10.1109/NETSOFT.2016.7502435 (citado nas pgs. 2, 13).

[Zhou *et al.* 2019] Luying Zhou, Huaqun Guo e Gelei Deng. "A fog computing based approach to DDoS mitigation in IIoT systems". Em: *Computers & Security* 85 (2019), pgs. 51–62. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2019.04.017. URL: https://www.sciencedirect.com/science/article/pii/S0167404818311349 (citado nas pgs. 29, 33, 35–37, 52).