

---

# Framework para autocura cognitiva de redes de banda larga

---

Enock Cabral Almeida Vieira



UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uberlândia  
2024



**Enock Cabral Almeida Vieira**

**Framework para autocura cognitiva de redes de  
banda larga**

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Flávio de Oliveira Silva

Uberlândia

2024

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU  
com dados informados pelo(a) próprio(a) autor(a).

V658 2024	<p>Vieira, Enock Cabral Almeida, 1982- Framework para autocura cognitiva de redes de banda larga [recurso eletrônico] / Enock Cabral Almeida Vieira. - 2024.</p> <p>Orientador: Flávio de Oliveira Silva. Dissertação (Mestrado) - Universidade Federal de Uberlândia, Pós-graduação em Ciência da Computação. Modo de acesso: Internet. Disponível em: <a href="http://doi.org/10.14393/ufu.di.2024.568">http://doi.org/10.14393/ufu.di.2024.568</a> Inclui bibliografia.</p> <p>1. Computação. I. Silva, Flávio de Oliveira, 1970-, (Orient.). II. Universidade Federal de Uberlândia. Pós- graduação em Ciência da Computação. III. Título.</p> <p style="text-align: right;">CDU: 681.3</p>
--------------	---

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:

Gizele Cristine Nunes do Couto - CRB6/2091  
Nelson Marcos Ferreira - CRB6/3074



**UNIVERSIDADE FEDERAL DE UBERLÂNDIA**  
Coordenação do Programa de Pós-Graduação em Ciência da  
Computação

Av. João Naves de Ávila, 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG,  
CEP 38400-902

Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpgfacom@ufu.br



## ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Dissertação de Mestrado, 22/2024, PPGCO				
Data:	31 de julho de 2024	Hora de início:	09:33	Hora de encerramento:	12:11
Matrícula do Discente:	12122CCP005				
Nome do Discente:	Enock Cabral Almeida Vieira				
Título do Trabalho:	Framework para autocura cognitiva de redes de banda larga				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Sistemas de Computação				
Projeto de Pesquisa de vinculação:	Gerenciamento Inteligente de Fatias de Rede Autônomas entre Múltiplos Domínios (INTERCESSION)				

Reuniu-se por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Rodrigo Sanches Miani - FACOM/UFU, Daniel Nunes Corujo - Departamento de Eletrônica e Telecomunicações da Universidade de Aveiro e Flávio de Oliveira Silva - Universidade do Minho, orientador do candidato.

Os examinadores participaram desde as seguintes localidades: Flávio de Oliveira Silva - Braga/Portugal e Daniel Nunes Corujo - Aveiro/Portugal . Os outros membros da banca e o aluno participaram da cidade de Uberlândia.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Flávio de Oliveira Silva, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir ao candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

**Aprovado**

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos,

conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Flávio de Oliveira Silva, Professor(a) do Magistério Superior**, em 13/08/2024, às 14:06, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Rodrigo Sanches Miani, Professor(a) do Magistério Superior**, em 13/08/2024, às 17:54, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [https://www.sei.ufu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **5534784** e o código CRC **25F267AD**.

*Este trabalho é dedicado à minha família, que sempre me apoiou nos meus sonhos. A meus pais, que por inúmeras vezes se sacrificaram para que eu pudesse seguir minha carreira acadêmica e profissional. Por fim, à minha esposa e filhos, que compreenderam e me suportaram na difícil missão de dividir vida acadêmica, trabalho profissional e vida pessoal.*



---

# Agradecimentos

Gostaria de agradecer aqui a muitas pessoas que tornaram esse trabalho possível. Ao meu colega de trabalho Régis Caixeta, à época líder do Centro de Operações de Rede da Algar Telecom, que me indicou que um tema com alto potencial para estudo seria o de redes de banda larga. Aos meus colegas de trabalho e universidade Daniel Ricardo e Natal Neto, que abriram meus olhos ao indicar caminhos pelos quais eu poderia estruturar minha dissertação. Aos colegas da equipe de marketing da Algar Telecom, Henrique Omena e Felipe Franco, com os quais aprendi bastante sobre a ligação entre tecnologia e negócio, sempre com foco no usuário final do serviço. Ao meu orientador Flávio Silva, que me mostrou com extrema gentileza e paciência, a direção para a criação do tema de pesquisa, elaboração da dissertação e publicação do meu artigo técnico internacional. Gostaria de também agradecer à operadora Algar Telecom, por disponibilizar a base de dados anônima com os parâmetros necessários para a realização deste trabalho. Um agradecimento especial aos amigos do BRAIN (Instituto de pesquisa e desenvolvimento tecnológico do grupo Algar), nas pessoas do Luiz Cláudio Theodoro, que sempre acreditou em unir a pesquisa acadêmica à indústria, e da Gessyca Bernardes, que topou meu desafio de realizar pesquisas de inteligência artificial aplicadas à banda larga, e cujo trabalho inspirou minha dissertação. Agradecimento ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) sob bolsa número 421944/2021-8 (chamada CNPq/MCTI/FNDCT 18/2021) e Centro Algoritmi, financiado pela Fundação para a Ciência e Tecnologia (FCT) no âmbito do Projecto Unidades de ID 2020-2023 (UIDB/00319/2020) por apoiar parcialmente este trabalho. E a Deus, o mais importante, que sempre foi a luz que me guiou em toda minha vida.



*“Atingir o equilíbrio é a meta suprema.”*  
*(Alziro Zarur)*



---

# Resumo

Com o crescente volume de tráfego de dados exigido pelos consumidores corporativos, empresariais e de varejo, os operadores de telecomunicações estão se tornando cada vez mais importantes na economia mundial. No entanto, os operadores devem preparar-se com soluções que permitam lidar mais rapidamente com os incidentes ou mesmo evitá-los, apostando sempre na manutenção de um nível aceitável de serviço ao cliente. Neste contexto, soluções Self-Healing (SH), apoiadas em mecanismos Machine Learning (ML), surgem como possibilidades para enfrentar este desafio. Este trabalho apresenta um framework de autocura cognitiva para ser implantado em operadoras de telecomunicações. Ele abrange autodiagnóstico, análise e atuação automática para mitigação de falhas em redes de banda larga de fibra óptica, por exemplo o Gigabit Passive Optical Network (GPON). Além disso, foi feita uma avaliação experimental utilizando um conjunto de dados anonimizados de usuários de uma operadora extraídos de seu Network Management System (NMS) e Customer Relationship Management (CRM), trazendo mais confiabilidade aos nossos resultados. Este trabalho mostra que o uso de ML em redes de banda larga é viável, e pode mudar a forma como as operadoras de telecomunicações gerenciam e melhoram a experiência do cliente. Neste trabalho, mostramos que um modelo inteligente pode aprender com os dados de uma rede de telecomunicações e tomar decisões sem intervenção humana. Três modelos cognitivos automáticos foram testados como prova experimental de conceito, com precisão média acima de 96%.

**Palavras-chave:** Aprendizado de máquina; Autocura; Banda Larga; Redes; Framework; Redes Neurais Artificiais.



---

# Abstract

With the growing volume of data traffic demanded by corporate, business, and retail consumers, telecommunications operators are becoming an increasingly important player in the world economy. However, the operators must prepare themselves with solutions that allow dealing with incidents more quickly or even avoid them, always focusing on maintaining an acceptable customer service level. In this context, SH solutions, supported by ML mechanisms, emerge as possibilities to address this challenge. This work presents a cognitive self-healing framework for telecommunications operators. This framework encompasses self-diagnosis, analysis, and automatic actuation for failure mitigation in fiber broadband telecommunications based on GPON. In addition, we did an experimental evaluation using an anonymized dataset from the operators' users, extracted from its NMS and CRM, bringing more reliability to our results. This work shows that using ML in telecommunication broadband networks is viable and can change how telecom operators manage and improve customer experience. We show that an intelligent model could do machine learning in telecom networks and make decisions without human intervention. Three automatic cognitive models were tested as experimental proof of concept with an average accuracy above 96%.

**Keywords:** Machine Learning; Self-Healing; Broadband; Network; Framework; ANN.



---

## Lista de ilustrações

Figura 1 – Rede de Banda Larga de Fibra . . . . .	32
Figura 2 – Redes ópticas passivas . . . . .	34
Figura 3 – Modem . . . . .	35
Figura 4 – NMS . . . . .	36
Figura 5 – Tela de um CRM real . . . . .	40
Figura 6 – Posição do Auto-Configuration Server (ACS) na rede de banda larga . . . . .	41
Figura 7 – Pilha do protocolo TR069 . . . . .	42
Figura 8 – Fluxo de mensagens inicial entre MODEM e ACS . . . . .	44
Figura 9 – Arquitetura genérica do DataLake . . . . .	45
Figura 10 – Tipos de aprendizado e seus algoritmos . . . . .	47
Figura 11 – Rede Neural Artificial . . . . .	48
Figura 12 – Fluxo genérico de Self-Healing . . . . .	53
Figura 13 – Aspectos de uma arquitetura de Self Healing . . . . .	54
Figura 14 – Domínios de ação do Self Healing . . . . .	55
Figura 15 – Framework de SH Cognitivo em redes de Banda Larga GPON . . . . .	60
Figura 16 – Módulos do Framework . . . . .	61
Figura 17 – Módulo de Rede e seus componentes . . . . .	61
Figura 18 – Módulo de Sistema . . . . .	63
Figura 19 – Módulo de Coleta de Dados . . . . .	64
Figura 20 – Módulo Cognitive Engine . . . . .	65
Figura 21 – Módulo de SH . . . . .	66
Figura 22 – Exemplo de chamada de Application Programming Interfaces (API) do módulo SH . . . . .	67
Figura 23 – Módulo de Correção . . . . .	69
Figura 24 – Diagrama de Sequência . . . . .	69
Figura 25 – Média dos parâmetros de rede dos usuários que reclamaram . . . . .	74
Figura 26 – Relação entre N-distant-devices e Perda de Pacotes . . . . .	75
Figura 27 – Frequência de N-Distant-Devices por volume de devices . . . . .	75

Figura 28 – Comparativo da média de dispositivos com mais de 10m do modem x QoS WiFi 5GHz . . . . .	76
Figura 29 – Jitter e Latência máximos x Reclamações . . . . .	76
Figura 30 – Árvore de Decisão Implementada - DT . . . . .	78
Figura 31 – Topologia da RNA Implementada . . . . .	79
Figura 32 – Rede Neural Artificial Implementada . . . . .	79
Figura 33 – Gradient Boosting Implementado . . . . .	80
Figura 34 – Matriz de confusão dos algoritmos implementados . . . . .	82

---

## Lista de tabelas

Tabela 1 – Velocidades médias por tipo de acesso banda larga . . . . .	32
Tabela 2 – Descrição elementos BroadBand Forum - Technical Report 069 (TR069)	42
Tabela 3 – Trabalhos Relacionados . . . . .	57
Tabela 4 – Parâmetros do dataset utilizado . . . . .	73
Tabela 5 – Resumo Estatístico dos Parâmetros do Dataset . . . . .	73
Tabela 6 – Resultados dos Algoritmos . . . . .	81
Tabela 7 – Trabalhos Relacionados x Framework proposto . . . . .	84



---

## Lista de siglas

**3G** Third Generation of Cellular Mobile Network

**3GPP** 3rd Generation Partnership Project

**4G** Fourth Generation Cellular Mobile Network

**5G** Fifth Generation Cellular Mobile Network

**ACS** Auto-Configuration Server

**ADSL** *Assimetric Digital Subscriber Line*

**ANN** Artificial Neural Network

**API** Application Programming Interfaces

**BNG** Broadband Network Gateway

**BSS** Business Support System

**CAPEX** Capital Expenditure

**CNN** Convolutional Neural Networks

**CPE** Customer Premises Equipment

**CRM** Customer Relationship Management

**CWMP** CPE WAN Management Protocol

**CX** Customer Experience

**DL** Deep Learning

**DT** Decision Tree

**DTL** Data Lake

**EML** Element Management Layer

**ETL** Extract Transform Load

**FTTx** Fiber to the x

**GB** Gradient Boosting

**GenAI** Generative Artificial Intelligence

**GPON** Gigabit Passive Optical Network

**HFC** Hybrid Fiber-Coaxial

**HTTP** HyperText Transfer Protocol

**HTTPS** HyperText Transfer Protocol Secure

**IA** Inteligencia Artificial

**IEEE** Institute of Electrical and Electronics Engineers

**IoT** Internet of Things

**IP** Internet Protocol

**IPTV** Internet Protocol Television

**ISP** Internet Service Provider

**ITSM** Information Technology Service Management

**ITU-T** International Telecommunication Union - Telecommunication

**LAN** Local Area Network

**MAC** Media Access Control

**ML** Machine Learning

**MLP** Multi-Layer Perceptron

**MODEM** Modulador-Demodulador

**MTTF** Mean Time To Fail

**MTTR** Mean Time To Repair

**NS** Networking Slicing

**NFV** Network Function Virtualization

**NFV** Network Functions Virtualization

**NGMN** Next Generation Mobile Networks

**NML** Network Management Layer

**NMS** Network Management System

**NOC** Network Operations Center

**NRT** Near Real Time

**NS** Network Slicing

**OAM** Operations and Maintenance

**OLT** Optical Line Terminal

**ONT** Optical Network Terminal

**ONU** Optical Network Unit

**OPEX** Operacional Expenditure

**OS** Ordem de Serviço

**OSS** Operations Support System

**PL** Packet Loss

**PON** Passive Optical Network

**QoE** Quality of Experience

**QoS** Quality of Service

**RNA** Rede Neural Artificial

**RPC** Remote Procedure Call

**SH** Self-Healing

**SLA** Service Level Agreement

**SNMP** Simple Network Management Protocol

**SOAP** Simple Object Access Protocol

**SOM** Service Order Management

**SON** Self-Organizing Network

**SVM** Support Vector Machines

**TCP** Transmission Control Protocol

**TCP/IP** Transmission Control Protocol/Internet Protocol

**TDMPON** Time Division Multiplex Passive Optical Network

**TI** Tecnologia de Informação

**TLS** Transport Layer Security

**TR069** BroadBand Forum - Technical Report 069

**TR369** BroadBand Forum - User Service Platform

**VDSL** Very-high-bit-rate Digital Subscriber Line

**VOIP** Voice over IP - Voz sobre IP (Internet protocol)

**Wi-Fi** Wireless Fidelity

**xDSL** Everything over Digital Subscriber Line

**XML** eXtensible Markup Language

---

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>27</b>
1.1	Motivação . . . . .	27
1.2	Objetivos e Desafios da Pesquisa . . . . .	29
1.3	Hipótese . . . . .	29
1.4	Contribuições . . . . .	29
1.5	Organização da Dissertação . . . . .	30
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> . . . . .	<b>31</b>
<b>2.1</b>	<b>Redes de Banda Larga</b> . . . . .	<b>31</b>
2.1.1	Conceitos e características . . . . .	32
2.1.2	Redes Ópticas e GPON . . . . .	33
2.1.3	Modulador-Demodulador (MODEM) . . . . .	35
2.1.4	Gerência de Redes - NMS . . . . .	36
2.1.5	Parâmetros de qualidade . . . . .	37
<b>2.2</b>	<b>Business Support System (BSS) e Operations Support System (OSS)</b> . . . . .	<b>38</b>
2.2.1	CRM . . . . .	39
2.2.2	ACS e TR069 . . . . .	40
2.2.3	Data Lake . . . . .	44
<b>2.3</b>	<b>Inteligência Artificial</b> . . . . .	<b>45</b>
2.3.1	Conceitos de Machine Learning . . . . .	46
2.3.2	Bibliotecas de Inteligência Artificial em Python . . . . .	49
<b>2.4</b>	<b>Self-Organizing Networks</b> . . . . .	<b>51</b>
2.4.1	Self-Healing . . . . .	52
2.4.2	Aspectos de uma arquitetura de Self-Healing . . . . .	54
<b>2.5</b>	<b>Trabalhos Relacionados</b> . . . . .	<b>55</b>

<b>3</b>	<b>FRAMEWORK PARA AUTOCURA COGNITIVA DE RE-</b>	
	<b>DES DE BANDA LARGA . . . . .</b>	<b>59</b>
<b>3.1</b>	<b>Visão Geral . . . . .</b>	<b>59</b>
<b>3.2</b>	<b>Módulo de Rede . . . . .</b>	<b>61</b>
3.2.1	Sub-Módulo MODEM . . . . .	61
3.2.2	Sub-Módulo NMS . . . . .	62
<b>3.3</b>	<b>Módulo de Sistema . . . . .</b>	<b>63</b>
<b>3.4</b>	<b>Módulo de Coleta de Dados - Data Lake (DTL) . . . . .</b>	<b>64</b>
<b>3.5</b>	<b>Módulo Cognitive Engine . . . . .</b>	<b>65</b>
<b>3.6</b>	<b>Módulo <i>Self-Healing</i> . . . . .</b>	<b>66</b>
<b>3.7</b>	<b>Módulo Correção . . . . .</b>	<b>68</b>
<b>3.8</b>	<b>Visão Comportamental da Arquitetura . . . . .</b>	<b>69</b>
<b>4</b>	<b>DISCUSSÃO E ANÁLISE DOS RESULTADOS . . . . .</b>	<b>71</b>
<b>4.1</b>	<b>Método para a Avaliação . . . . .</b>	<b>71</b>
<b>4.2</b>	<b><i>Dataset</i> . . . . .</b>	<b>72</b>
<b>4.3</b>	<b>Discussões gerais sobre os algoritmos . . . . .</b>	<b>77</b>
4.3.1	Decision Tree (DT) . . . . .	77
4.3.2	Rede Neural Artificial (RNA) (Multi-Layer Perceptron (MLP)) . . . . .	78
4.3.3	Gradient Boosting (GB) . . . . .	80
<b>4.4</b>	<b>Avaliação dos Resultados . . . . .</b>	<b>81</b>
4.4.1	<i>Resultados dos Algoritmos implementados</i> . . . . .	81
<b>4.5</b>	<b>Visão deste trabalho e comparação com o estado da arte . . . . .</b>	<b>83</b>
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>85</b>
<b>5.1</b>	<b>Principais Contribuições . . . . .</b>	<b>85</b>
<b>5.2</b>	<b>Trabalhos Futuros . . . . .</b>	<b>86</b>
<b>5.3</b>	<b>Contribuições em Produção Bibliográfica . . . . .</b>	<b>86</b>
<b>REFERÊNCIAS</b>	<b>. . . . .</b>	<b>87</b>

---

# Introdução

Este trabalho propõe um novo Framework para implantação de autocura de redes de banda larga com o uso de técnicas de ML.

## 1.1 Motivação

Com o crescente volume de tráfego de dados demandado pelos consumidores corporativos, empresariais e de varejo, impulsionado pelo crescente nível de aderência das pessoas e empresas a meios digitais conectados pela internet, as operadoras de telecomunicações se tornam cada vez mais importantes na economia mundial. Naturalmente, a demanda por conectividade vem acompanhada de uma exigência de nível de serviço, conhecido como *Service Level Agreement (SLA)*, cada vez mais alto, e isso por si só já demanda uma nova forma de trabalho nas operadoras para garantir esse maior nível de serviço.

O surgimento do Fifth Generation Cellular Mobile Network (5G) e das aplicações de Internet of Things (IoT), trás a perspectiva de um crescimento exponencial do número de dispositivos conectados às redes móveis. Segundo Uzoka, Cadet e Ojukwu (2024), a infraestrutura de telecomunicações fornecerá os caminhos de comunicação essenciais que facilitarão a transmissão de dados entre dispositivos IoT.

Segundo Xie, Wei e Wang (2021), o surgimento de demandas cada vez maiores de dados é também um reflexo da adoção de aplicações como vídeos de altíssima resolução (4K/8K) pelo usuário da rede, além do crescente uso de realidades virtual e aumentada, que fazem mudar de patamar o perfil de consumo do cliente das operadoras de telecomunicações.

Sobre fatores tecnológicos que estão revolucionando as telecomunicações, nota-se o crescimento da adoção de serviços em nuvem, que criam cada vez mais dependência da conectividade, aumenta o desafio de prestação de serviços de alta confiabilidade por parte das operadoras de telecomunicações.

Tais fatos indicam que as operadoras precisam se preparar, com soluções que possibilitam tratar mais rapidamente incidentes ou mesmo evitar os mesmos, sempre com o

foco de manter o nível de serviço em um patamar aceitável pelos clientes, cada vez mais exigentes.

Sabe-se que as operadoras, além das exigências de mercado já citadas, ainda possuem desafios internos, como a operação de redes cada vez mais complexas e multi-fabricantes (fenômeno muitas vezes ocasionado pelas frequentes fusões e aquisições). Tal fenômeno, ainda leva ao desafio de que as empresas cresçam sem crescer sua estrutura de pessoal na mesma proporção, de maneira a equilibrar custos. Além disso, é sabido que o mercado de telecomunicações sofre com o conceito de "*Lock-in*", que significa que os fabricantes de rede tendem a dificultar a interoperabilidade de soluções e elementos de rede, o que pode gerar mais custos para as operadoras e limitações de inovação.

Ainda dentro das operadoras, tecnologias que demandam tratamentos diferentes por cliente como o fatiamento de rede - *Network Slicing (NS)*, e o *Quality of Service (QoS)*, aumentam a complexidade de gerenciamento das redes.

Considerando essa complexidade operacional crescente, seja pelas fusões e aquisições ou por tecnologias como o NS e serviços em nuvem, uma demanda por profissionais capacitados ("especialistas") é cada vez maior, ao passo que essa demanda é da mesma forma muito difícil de ser resolvida devido a escassez destes profissionais no mercado.

A frequência de falhas de rede - *Mean Time To Fail (MTTF)* e a quantidade de tempo de inatividade do sistema são alguns dos fatores mais importantes que um provedor de serviços de comunicação usa para medir a eficácia de sua rede. Estes parâmetros devem ser controlados para os provedores de telecomunicações se manterem competitivos e manterem seus clientes satisfeitos — especialmente os clientes corporativos, dos quais as operadoras dependerão cada vez mais para cumprir suas metas de receita.

Soluções de autocura (SH), usando uma arquitetura baseada em nuvem com Inteligência Artificial (IA), se propõem a garantir que as operadoras possam ser preditivas em sua manutenção de rede, a partir do uso de tecnologias como *Analytics* e IA. A rede pode se tornar tão automatizada que requer "*toque zero*" dos humanos e pode se auto-corrigir. Os benefícios disso são: a rede pode prever uma falha e atenuá-la antes mesmo de acontecer.

Com a crescente implantação de novas tecnologias de Automação, Inteligência Artificial, *Analytics* e API, criam-se possibilidades de aplicações de SH em vários tipos de redes, como por exemplo as redes de acesso Fiber to the x (FTTx), onde o X designa a aplicação final da rede de fibra óptica (Home, Curb, Building); redes metropolitanas (anéis metro), redes de backbone interurbano, redes móveis (Fourth Generation Cellular Mobile Network (4G), 5G) e data centers.

Todos esses tipos de redes de telecomunicações possuem uma necessidade em comum: níveis de serviço elevados. Apesar da particularidade de cada uma, todas necessitam de ações de identificação de falhas/anomalias, análise dessas falhas, e propostas de correção das mesmas. Hoje, grande parte dessas atividades é feita manualmente pelos analistas das operadoras de telecomunicações. Isso pode gerar mais tempo para se resolver um

problema, aumentando o *Mean Time To Repair (MTTR)*, gerando impacto no SLA e consequente aumento da insatisfação dos usuários do serviço.

Pelo apresentado, soluções de SH podem atuar diretamente nestes problemas, pois pode oferecer identificação automática de falhas, além de análise e correção automáticas de anomalias nas redes.

## 1.2 Objetivos e Desafios da Pesquisa

O presente trabalho tem como objetivo apresentar um framework genérico de SH, para ser implementado em operadoras de telecomunicações, e que seja capaz de utilizar técnicas de ML. Além disso, apresentaremos uma análise experimental com dados reais de uma rede de banda larga.

## 1.3 Hipótese

Segundo NETO (2021), espera-se que um sistema com propriedade de autocura (SH), seja capaz de monitorar e reconhecer anomalias, localizar a fonte de uma falha, responder a condições variáveis e executar mecanismos para levar o sistema de volta ao estado operacional *Normal*.

Considerando todo o exposto, naturalmente o conceito de SH, ou autocura de redes, parece se tornar, num futuro próximo, um grande objetivo das operadoras de telecomunicações para manter o nível de serviço desejado pelos seus clientes.

Entendemos que é possível criar uma solução que englobe autodiagnóstico, análise, aplicação de inteligência e, que a partir de automações, consiga tratar falhas de rede de maneira a reduzir o tempo de indisponibilidade das redes de telecomunicações.

Espera-se que, ao final do trabalho, consigamos validar a hipótese de que é possível utilizar um framework de SH inteligente genérico em qualquer operadora de banda larga, usando ML, com precisão adequada para autocura da rede.

## 1.4 Contribuições

Entre as contribuições científicas deste trabalho podemos citar:

- ❑ Criação de um framework genérico e padronizado para autocura de redes de banda larga que pode ser implementado pelas operadoras de telecomunicações
- ❑ Criação de um módulo de IA que pode ser reutilizado em várias arquiteturas de rede

- ❑ Implementação e análise comparativa de algoritmos de ML aplicados em SH para redes banda larga
- ❑ Análise detalhada do comportamento de parâmetros do usuário de redes de banda larga, identificando causas de problemas de performance e má experiência de uso do serviço
- ❑ Publicação de um conjunto de dados (*dataset*) anonimizado com dados reais de usuários de uma operadora de telecomunicações

## 1.5 Organização da Dissertação

Esta dissertação está organizada em cinco capítulos. O Capítulo 2 detalha e discute os aspectos teóricos e práticos que esclarecem os diversos temas abordados no trabalho. As tecnologias e conceitos de redes de banda larga são detalhadas, tais como redes ópticas, equipamentos do usuário, elementos de gestão da rede, parâmetros de qualidade da rede do usuário. Também são conceituados os sistemas de suporte ao negócio e à operação necessários para a prestação do serviço de banda larga. Por fim, nivelamos conceitos de IA e redes autônomas necessários para a compreensão do capítulo seguinte. Ainda neste capítulo descrevemos os trabalhos relacionados ao tema de pesquisa. O Capítulo 3 descreve a proposta de dissertação, bem como detalhes da estrutura do framework proposto e de seus módulos. O Capítulo 4 apresenta uma análise detalhada do *dataset* colhido na operadora visitada, além de três avaliações experimentais e seus resultados. Ainda neste capítulo apresentamos uma análise comparativa das contribuições deste trabalho em relação às propostas anteriores. O Capítulo 5 discute a conclusão alcançada na construção e experimentação da solução proposta. Além disso, este capítulo aponta novos casos de uso e abordagens que podem ser alcançados pelo framework proposto.

---

## Fundamentação Teórica

Nesse capítulo são apresentados os conceitos básicos que suportam o nosso trabalho. Discorreremos sobre os conceitos de redes de banda larga e seus elementos, os parâmetros de qualidade desta rede, assim como explicamos quais são os sistemas de suporte à operação e ao negócio de telecomunicações. Também são definidos os conceitos de inteligência artificial, e de redes auto organizáveis, entre eles a autocura. Por fim, apresentamos os trabalhos relacionados ao tema dessa dissertação.

### 2.1 Redes de Banda Larga

A rede de banda larga fixa residencial é um elemento fundamental para a sociedade moderna. Ela desempenha um papel crucial na conexão de indivíduos, comunidades e empresas, permitindo o acesso a uma infinidade de serviços e informações online.

A banda larga fixa residencial tem um impacto significativo em várias áreas da vida cotidiana. Ela permite o acesso a uma ampla gama de informações e serviços online, desde entretenimento e jogos até educação e trabalho remoto. Além disso, a banda larga fixa residencial também facilita a comunicação e a interação social, permitindo chamadas de vídeo, redes sociais e outras formas de comunicação digital.

A evolução tecnológica da banda larga fixa residencial tem sido notável. Considerando a parte de acesso final ao usuário, começamos com a tecnologia *Asymmetric Digital Subscriber Line* (ADSL), que oferecia velocidades de download muito mais rápidas do que as velocidades de upload. Isso era ideal para a maioria dos usuários domésticos, pois a maioria das atividades online era de navegação na web.

No entanto, à medida que a demanda por velocidades de internet mais rápidas e a necessidade de uploads mais rápidos (por exemplo, para jogos online e videoconferências) aumentou, a tecnologia evoluiu. Passamos para a tecnologia Very-high-bit-rate Digital Subscriber Line (VDSL), que oferecia velocidades de download e upload significativamente mais rápidas do que o ADSL.

Outra tecnologia que ainda existe, mas que está cada vez mais em desuso é o Hybrid Fiber-Coaxial (HFC). A tecnologia HFC combina cabos de fibra óptica e coaxiais para fornecer serviços de internet. A fibra óptica é usada para transmitir dados até um ponto próximo às residências, onde os cabos coaxiais assumem a distribuição final.

Finalmente, chegamos à tecnologia GPON, que é a tecnologia de fibra óptica mais recente e mais rápida disponível para banda larga fixa residencial em larga escala. O GPON oferece velocidades de download e upload incrivelmente rápidas, permitindo que os usuários aproveitem ao máximo a internet de alta velocidade.

A Tabela 1 faz uma comparação das velocidades médias das tecnologias de acesso de banda larga existentes.

Tabela 1 – Velocidades médias por tipo de acesso banda larga

Tecnologia de acesso	Velocidade média (Mbps)	Velocidade Relativa em relação ao ADSL
ADSL	8	1x
VSDL	50	6.25x
HFC	100	12.5x
GPON	1000	125x

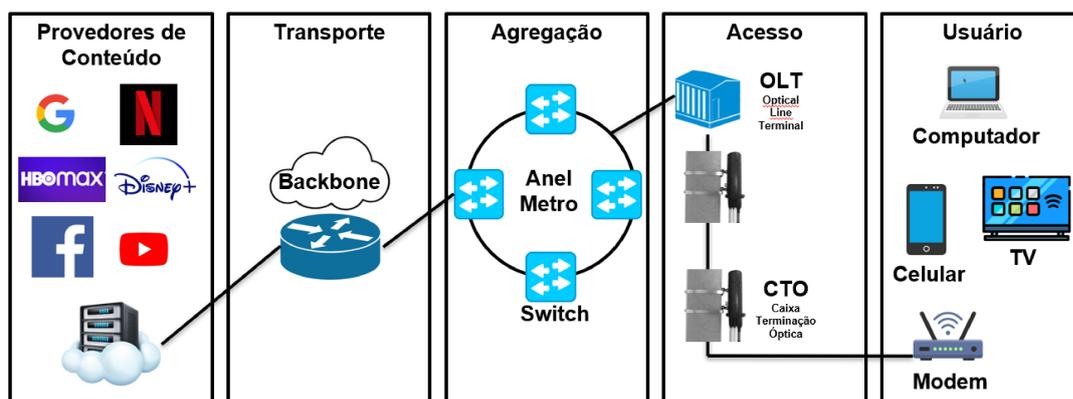
Fonte: O autor

### 2.1.1 Conceitos e características

Para existir uma comunicação entre um computador pessoal na residência de uma pessoa e um site de internet, por exemplo, um provedor de conteúdo como o Netflix, existe um caminho composto de vários elementos diferentes por onde os dados passam antes de a comunicação ter uma resposta.

A Figura 1 ilustra os elementos de uma rede de banda larga em fibra.

Figura 1 – Rede de Banda Larga de Fibra



Fonte: O autor

De acordo com NTIA (2022), os elementos de uma Rede de Banda Larga de Fibra são:

- ❑ **Acesso:** O acesso é a primeira milha da rede, onde os usuários finais se conectam a partir do seu MODEM. Isso pode ser feito através de uma conexão direta com fibra óptica (GPON por exemplo) ou através de tecnologias legadas, como Everything over Digital Subscriber Line (xDSL) ou Cabo Coaxial (HFC, que depois se conectam a uma rede de fibra mais ampla.
- ❑ **Agregação:** Os dados dos usuários são então enviados a um ponto de agregação, onde são coletados e encaminhados para o núcleo da rede. Esses pontos de agregação são cruciais para consolidar o tráfego de dados e otimizar o uso da infraestrutura.
- ❑ **Transporte:** A rede de transporte é a espinha dorsal da rede de banda larga de fibra, conectando diferentes pontos de agregação e levando os dados a longas distâncias. Essa parte da rede é composta por cabos de fibra óptica de alta capacidade e é projetada para ser altamente confiável e eficiente.
- ❑ **Provedores de Conteúdo:** Finalmente, os dados chegam aos provedores de conteúdo, que são os serviços e plataformas acessados pelos usuários finais. Esses provedores podem ser serviços de streaming, sites de notícias, plataformas de mídia social e outros.

### 2.1.2 Redes Ópticas e GPON

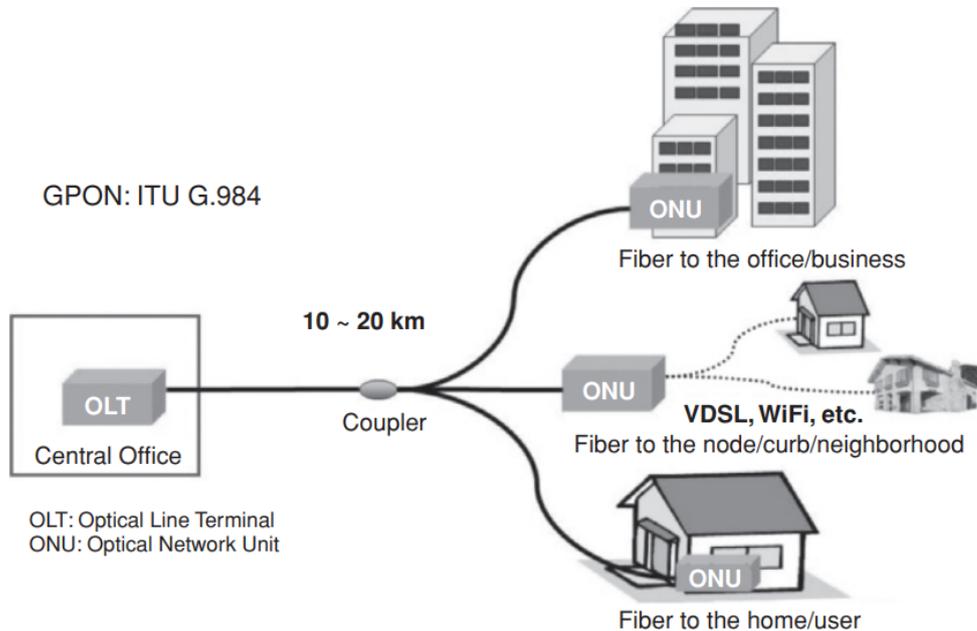
Nesta seção explicamos em mais detalhes uma das tecnologias de acesso de banda larga, o GPON. Segundo Kazovsky et al. (2011), devido à sua largura de banda ultra-alta e baixa atenuação, as fibras ópticas têm sido amplamente implantadas para redes metropolitanas e nos últimos anos no acesso direto a residências e empresas.

Com o advento das aplicações multimídia como stream de vídeo, surgiram demandas de largura de banda tão grandes que as tecnologias de fio de cobre atingiram seus limites de largura de banda. Enquanto isso, componentes ópticos de baixo custo e arquitetura de rede óptica passiva tornaram a fibra uma solução muito atraente. Nos últimos anos, várias arquiteturas e tecnologias Passive Optical Network (PON) foram estudadas pela indústria de telecomunicações, e alguns padrões PON foram aprovados pela International Telecommunication Union - Telecommunication (ITU-T) e Institute of Electrical and Electronics Engineers (IEEE). Hoje FTTx torna-se uma tecnologia madura que vem desbancando os fios de cobre.

A figura 2 ilustra a arquitetura de uma rede óptica passiva.

Como o nome indica, não há nenhum componente ativo entre o escritório central e as instalações do usuário. Entende-se por componente ativo um equipamento alimentado

Figura 2 – Redes ópticas passivas



Fonte: Adaptado de Kazovsky et al. (2011)

por energia elétrica diretamente. Nesta arquitetura, os dispositivos ativos existem apenas no escritório central (central office) e nas instalações do usuário. A partir do escritório central, uma fibra óptica padrão (fibra alimentadora) funciona como um divisor de energia óptica passiva de 1 : N perto das instalações do usuário. As portas de saída do divisor passivo se conectam aos usuários através de fibras individuais (fibras de distribuição).

A distância de transmissão em uma rede óptica passiva é limitada a 20 km, conforme especificado na norma atual (G.984 - ITU - International Telecommunication Union, 2008). As fibras e componentes passivos entre o escritório central e as instalações dos usuários são comumente chamados de rede óptica de distribuição. O número de usuários suportados por um PON pode ser de 2 a 128, dependendo do balanceamento de potência, mas normalmente, 16, 32 ou 64.

No escritório central, um terminal de linha óptica (Optical Line Terminal (OLT)) transmite dados usando comprimento de onda de 1490 nm, já o vídeo é enviado através de comprimento de onda de 1550 nm. O fluxo de *Downstream* usa um esquema de transmissão e seleção; ou seja, os dados e o vídeo em downstream são transmitidos para cada usuário com endereços Media Access Control (MAC) e o usuário seleciona os endereços MAC baseados em pacotes de dados.

No ambiente do usuário, uma unidade de rede óptica (Optical Network Unit (ONU)), também chamada de terminal de rede óptica (Optical Network Terminal (ONT)), transmite dados *upstream* em comprimento de onda de 1310 nm. Para evitar colisão, a transmissão upstream usa um protocolo de acesso múltiplo (ou seja, acesso múltiplo de divisão de tempo) para atribuir slots de tempo a cada usuário. Esse tipo de rede óptica pas-

siva é chamado de Time Division Multiplex Passive Optical Network (TDMPON). A ONU poderia estar localizada em uma casa, escritório, um gabinete na calçada ou em outro lugar. Assim surge a chamada fibra para casa/escritório/empresa/bairro/meio-fio/usuário/instalações/nó, todos comumente referidos como fibra para o x, ou FTTx. No caso de fibra até a vizinhança/meio-fio/nó, os pares trançados são normalmente implantados para conectar os usuários finais às ONUs, fornecendo assim uma solução híbrida de acesso de fibra/metálico.

### 2.1.3 MODEM

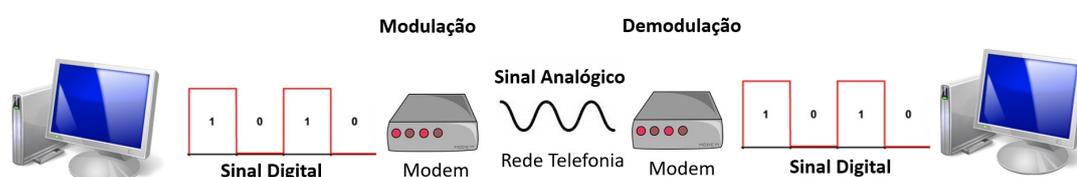
Segundo Karam e Mounsef (2008), um MODEM é um dispositivo de hardware que converte dados para permitir a transmissão entre um computador e uma rede.

No contexto de uma rede de banda larga fixa, o modem desempenha várias funções importantes:

- ❑ **Conexão à Internet:** O modem conecta sua rede doméstica à Internet, geralmente através de um provedor de serviços de Internet (Internet Service Provider (ISP)). Ele recebe sinais da linha de comunicação (que pode ser via cabo coaxial (HFC), par metálico (xDSL), fibra óptica (GPON), etc.) e os converte em dados digitais.
- ❑ **Modulação e Demodulação:** O modem modula os sinais digitais do computador em sinais analógicos que podem ser transmitidos pela linha de comunicação. Da mesma forma, ele demodula os sinais analógicos recebidos de volta em sinais digitais que seu computador pode entender.
- ❑ **Provisão de Endereço Internet Protocol (IP):** O modem geralmente obtém um endereço IP do seu ISP, que é usado para identificar sua rede na Internet.
- ❑ **Segurança:** Muitos modems modernos também possuem recursos de segurança embutidos, como firewalls, para proteger a rede de ameaças online.

A figura 3 ilustra o papel do modem durante a modulação e demodulação de sinais em uma rede de computadores.

Figura 3 – Modem



Fonte: O autor

Como mostrado por suas funções, o MODEM desempenha um papel vital em permitir e gerenciar a conexão com a Internet em uma rede de banda larga fixa. Ele atua como a "porta de entrada" para a Internet em casa ou no escritório.

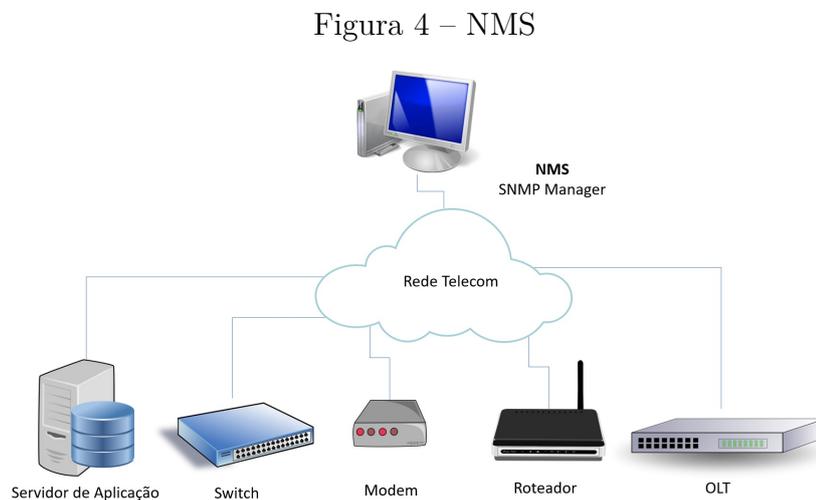
No contexto das redes de banda larga atuais, alguns dos principais fornecedores de MODEMs GPON são: Huawei, FiberHome, Nokia, Zhone e Calix.

### 2.1.4 Gerência de Redes - NMS

Um NMS aplicado a uma rede de banda larga de fibra é uma plataforma de software que permite aos operadores de rede monitorar, gerenciar e otimizar a infraestrutura de rede de fibra óptica. O NMS fornece uma visão abrangente do desempenho da rede, ajudando a identificar e resolver problemas rapidamente, garantindo assim a qualidade do serviço e a experiência do usuário. Normalmente é denominado Gerência de Rede (RAMAN, 1999).

O NMS geralmente se integra a diferentes camadas da rede, como a camada de gerenciamento de rede (Network Management Layer (NML)) e a camada de gerenciamento de elementos (Element Management Layer (EML)) e usa o protocolo Simple Network Management Protocol (SNMP).

Na figura 4, temos um exemplo de topologia de um NMS controlando elementos de uma rede genérica.



Segundo Clemm (2006), esses sistemas são essenciais para a operação eficiente de redes de banda larga de fibra, pois fornecem ferramentas para:

- ❑ Monitoramento em tempo real: Verificar o status dos dispositivos de rede e o tráfego de dados.
- ❑ Gerenciamento de falhas: Detectar e isolar problemas na rede, facilitando a rápida resolução.

- ❑ Configuração e provisionamento de rede: Configurar dispositivos de rede e serviços para novos usuários ou alterar configurações existentes.
- ❑ Desempenho e otimização: Analisar o desempenho da rede para identificar gargalos e otimizar o fluxo de tráfego.
- ❑ Segurança: Proteger a rede contra acessos não autorizados e ataques cibernéticos.

A implementação de um NMS em uma rede de banda larga de fibra óptica é um passo crucial para garantir a alta disponibilidade e confiabilidade dos serviços de telecomunicações, especialmente em um cenário onde a demanda por dados é crescente e a infraestrutura de rede se torna cada vez mais complexa.

### 2.1.5 Parâmetros de qualidade

Os parâmetros de medição de qualidade de uma rede de banda larga são indicadores que mostram às operadoras como está a experiência de uso do serviço de internet dos seus usuários. Os principais são:

- ❑ Latência: Refere-se ao tempo que um pacote de dados leva para viajar de um ponto a outro na rede. É medida em milissegundos (ms) e um valor baixo indica uma rede mais rápida e responsiva. Em redes de banda larga, uma latência baixa é essencial para aplicações como jogos online e chamadas de vídeo, onde o atraso pode ser perceptível e prejudicial à experiência (BRISCOE et al., 2016).
- ❑ Jitter: É a variação no tempo de atraso entre pacotes de dados enviados pela mesma fonte, medido em milissegundos (ms). Em outras palavras, é a inconsistência no tempo de entrega dos pacotes. Valores baixos de jitter são preferíveis para manter a qualidade da transmissão, especialmente em aplicações em tempo real como chamadas de vídeo ou jogos online (BRISCOE et al., 2016).
- ❑ Perda de pacotes (Packet Loss (PL)): Ocorre quando pacotes de dados enviados pela rede não alcançam seu destino final. Isso pode ser causado por congestionamento na rede, falhas no hardware ou sinal fraco. A perda de pacotes é geralmente expressa como uma porcentagem e, para uma boa qualidade de rede, essa porcentagem deve ser a mais baixa possível. Uma alta perda de pacotes é particularmente problemático para streaming de vídeo ou jogos online (BRISCOE et al., 2016).
- ❑ N distant devices (Dispositivos distantes): Refere-se ao número de dispositivos que estão fisicamente distantes do MODEM ou roteador. Quanto maior a distância, maior a possibilidade de interferências e perda de sinal, o que pode afetar a qualidade da conexão.

- Qualidade do canal Wireless Fidelity (Wi-Fi): Avalia a qualidade da conexão sem fio. Canais Wi-Fi podem sofrer interferências de outros dispositivos eletrônicos, o que pode diminuir a qualidade do sinal. A qualidade é geralmente avaliada em uma escala e valores mais altos indicam um canal mais limpo e uma melhor qualidade de conexão. A nota de qualidade do canal Wi-Fi do MODEM, que varia de 0 a 5, é determinada com base em vários fatores que afetam a performance da rede, como a quantidade de interferência de outros dispositivos e redes Wi-Fi próximas, o número de dispositivos conectados ao mesmo canal, a intensidade do sinal Wi-Fi medida em dBm (decibel-miliwatts) e a quantidade de erros de transmissão de dados. Menos interferência e congestionamento, sinais mais fortes e menos erros de transmissão geralmente resultam em uma nota mais alta, que é atribuída automaticamente pelo MODEM.

## 2.2 BSS e OSS

Segundo Sathyan (2016), os OSS e os BSS são sistemas de Tecnologia de Informação (TI) de missão crítica, que oferecem suporte a uma ampla variedade de serviços de telecomunicações.

O OSS é uma coleção de programas de computador projetados para ajudar os provedores a monitorar, analisar e gerenciar redes de telecomunicações. Ele se refere a ferramentas de hardware e software usadas nas operações de uma rede de telecomunicações, incluindo computadores, servidores, roteadores e outros componentes principais. O OSS permite que as organizações de telecomunicações supervisionem uma variedade de funções operacionais, como inventário de rede, gerenciamento de falhas, provisionamento de serviços, configuração, garantia de serviço e planejamento de rede.

Ainda de acordo com Sathyan (2016), por outro lado, o BSS refere-se à variedade de programas de software que ajudam as organizações de telecomunicações a gerenciar e simplificar todas as atividades voltadas para o cliente. Esses sistemas abrangem o lado comercial das telecomunicações em vez do lado tecnológico. O BSS permite interações fáceis entre provedores e clientes de telecomunicações, desde a solicitação de serviços até a resolução de problemas de cobrança e a atualização de assinaturas. As aplicações BSS aceleram a rentabilização de produtos, melhoram a gestão de clientes, melhoram a gestão de receitas e melhoram a eficiência operacional global e o desempenho empresarial.

Embora o OSS e o BSS tenham finalidades diferentes em áreas diferentes de telecomunicações, ambos os sistemas ajudam as organizações de telecomunicações a simplificar os processos do dia a dia, aumentar a produtividade e superar uma série de desafios da indústria.

### 2.2.1 CRM

O conceito de CRM, para facilitar o desenvolvimento e a manutenção de relacionamentos de longo prazo com os clientes, surgiu na década de 1990 (GALBREATH; ROGERS, 1999). Já Cheng e Shiu (2019) destaca uma ligação entre a implementação do CRM e o desempenho da empresa, mostrando que existe uma evolução significativa na maturidade de gestão do cliente.

Um sistema CRM ou Gestão de Relacionamento com o Cliente, é uma ferramenta tecnológica projetada para gerenciar todas as interações de uma empresa com clientes atuais e potenciais. Ele centraliza dados, automatiza processos de vendas, marketing e atendimento ao cliente, e ajuda a fortalecer o relacionamento com o cliente (PEREZ-VEGA et al., 2022). Esses sistemas ajudam as empresas a entender melhor seus clientes, aprimorar a eficiência operacional e impulsionar o crescimento.

A seguir alguns exemplos de informações armazenadas por este sistema:

1. Dados de Contato: Nome, endereço, telefone, e-mail.
2. Histórico de Interações: Registros de chamadas, e-mails, visitas ao site e reuniões.
3. Histórico de Compras: Detalhes das transações anteriores.
4. Reclamações de Clientes: Registros de problemas relatados e soluções oferecidas pela empresa.
5. Preferências e Comportamento: Informações sobre preferências de produtos/serviços e padrões de compra.

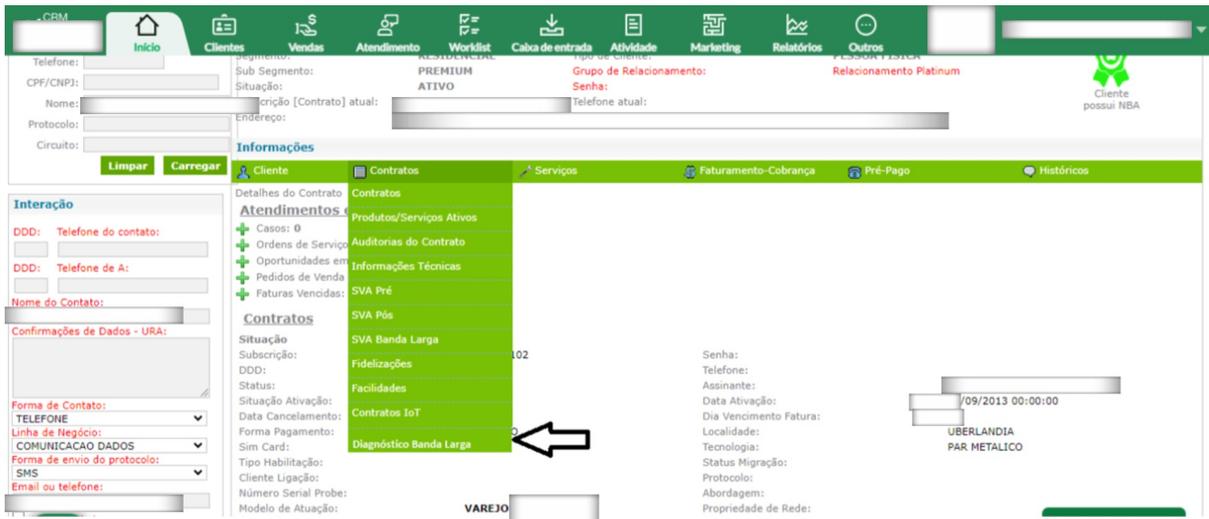
Com relação ao acesso e propósito de um CRM, podemos citar que as equipes de vendas o usam para gerenciar leads, oportunidades e fechar negócios. Já as equipes de marketing podem adotá-lo para segmentar clientes, gerenciar campanhas e analisar a eficácia do marketing realizado. As equipes de atendimento ao cliente trabalham com o CRM para resolver reclamações e fornecer suporte. Por fim, equipes de gestão usam este sistema para tomar decisões estratégicas baseadas em dados.

A figura 5 ilustra um exemplo de tela de um sistema CRM.

Em uma arquitetura OSS/BSS, que é comum em empresas de telecomunicações, o CRM desempenha um papel crucial no BSS, gerenciando o lado comercial das operações. Ele interage com o OSS para garantir que os serviços sejam provisionados corretamente e que as necessidades dos clientes sejam atendidas, influenciando diretamente a experiência do cliente e a receita da empresa.

Entre os principais fornecedores de CRM no mercado podemos citar Salesforce, HubSpot, SAP, Zendesk Sell, Oracle, ou até sistemas feitos pelas equipes de desenvolvimento da própria operadora.

Figura 5 – Tela de um CRM real



Fonte: O autor

### 2.2.2 ACS e TR069

Segundo o BroadBandForum (2020), TR069, também conhecido como CPE WAN Management Protocol (CWMP), é um protocolo de gerenciamento criado pela mesma instituição, que é utilizado para facilitar a comunicação entre dispositivos de rede, como roteadores Wi-Fi, MODEMS, e um servidor; permitindo o envio de comandos remotos para gerar e receber relatórios da rede, atualizações de firmware, fazer correções de parâmetros, entre outros.

O BroadBandForum (2020) relatou no segundo semestre de 2020 mais de 1 bilhão de instalações TR069, mostrando a robustez do protocolo que iniciou sua massificação em 2004 em todo o mundo.

Desde a sua criação, o protocolo era mais usado em MODEMS, roteadores e gateways. Mas, com o tempo, novos dispositivos compatíveis com o TR069 foram surgindo, como Internet Protocol Television (IPTV), e telefones com Voice over IP - Voz sobre IP (Internet protocol) (VOIP).

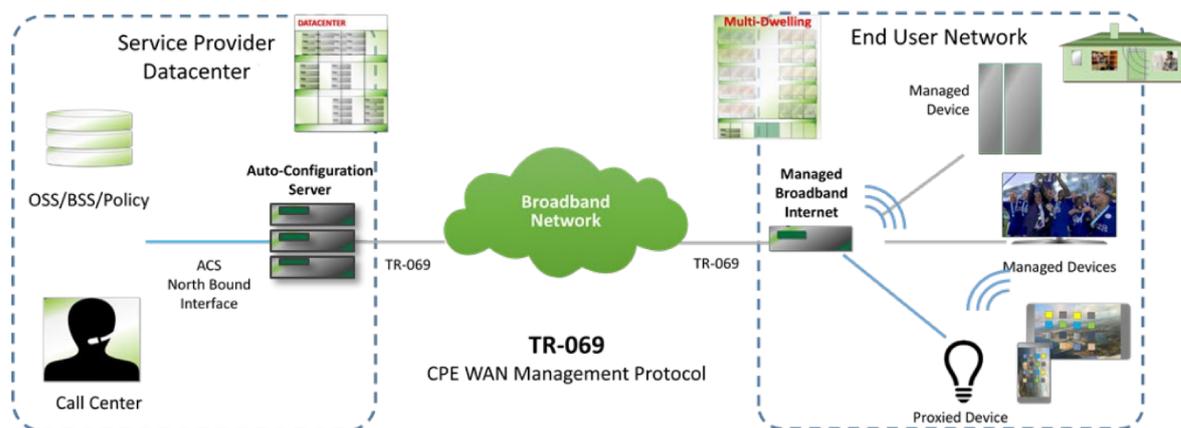
Na rede de telecomunicações, o TR069 desempenha várias funções importantes:

- ❑ Configuração automática e provisionamento dinâmico de serviços: Permite que os dispositivos sejam configurados e os serviços provisionados automaticamente sem intervenção manual.
- ❑ Gerenciamento de software/firmware: Facilita a atualização e manutenção do software dos dispositivos remotamente.
- ❑ Monitoramento de status e performance: Ajuda a monitorar o desempenho e o estado dos dispositivos na rede.

- ❑ Diagnósticos: Oferece ferramentas para diagnosticar problemas na rede.
- ❑ Testes de velocidade: Permite realizar testes para avaliar a velocidade da conexão.

A figura 6, ilustra a topologia de um servidor ACS na rede de banda larga.

Figura 6 – Posição do ACS na rede de banda larga



Fonte: (BROADBANDFORUM, 2020)

O ACS é um componente central no uso do TR069 e que faz parte da família de sistemas de OSS. Ele atua como o servidor responsável pela execução das funções do protocolo, gerenciando os dispositivos de rede, conhecidos como Customer Premises Equipment (CPE)s. O ACS pode ser responsável por um ou mais CPEs e é essencial para a configuração automática, gerenciamento e diagnóstico dos dispositivos. O ACS é agnóstico ao fabricante do MODEM, pois usa o protocolo padrão TR069, dando flexibilidade para a operadora na escolha dos fabricantes de MODEMs e redes. Além disso, o ACS permite que os ISPs gerenciem e configurem os dispositivos de rede de seus clientes de forma remota, coletando dados de desempenho e realizando atualizações de software e firmware conforme necessário.

Algumas premissas foram adotadas na definição do protocolo TR069:

- ❑ Todos os CPEs (MODEMs) independente do tipo (bridge, router ou outro) obtêm um endereço IP para se comunicarem com um ACS.
- ❑ Um MODEM pode interagir com um único ACS por vez. A qualquer momento, um MODEM reconhece exatamente um ACS com o qual pode se conectar. (Observação: uma coleção de ACSs atrás de um balanceador de carga é considerada um único ACS de acordo com a norma BroadBandForum (2020).)

De acordo com BroadBandForum (2020), a pilha do protocolo TR069 é a mostrada na Figura 7. Uma breve descrição de cada camada é fornecida na Tabela 2. Observe que o MODEM e o ACS devem aderir aos requisitos dos protocolos padrão subjacentes.

Figura 7 – Pilha do protocolo TR069

CPE/ACS Management Application
RPC Methods
SOAP
HTTP
SSL/TLS
TCP/IP

Fonte: (BROADBANDFORUM, 2020)

Tabela 2 – Descrição elementos TR069

Camada	Descrição
<b>CPE/ACS Applic.</b>	A aplicação usa o protocolo de gerenciamento TR069 no MODEM e no ACS. A aplicação é definida localmente e não especificada como parte do protocolo.
<b>RPC Methods</b>	Os métodos Remote Procedure Call (RPC) específicos definidos pelo protocolo de gerenciamento TR069 (ex.: Modem Reboot).
<b>SOAP</b>	Simple Object Access Protocol (SOAP) é uma sintaxe padrão baseada em XML usada para codificar chamadas de procedimento remoto.
<b>HTTP</b>	HyperText Transfer Protocol (HTTP) 1.1 - linguagem de hipertexto que atua como uma abstração das camadas inferiores, facilitando a programação
<b>TLS</b>	O protocolo padrão de segurança da camada de transporte da Internet. Especificamente, Transport Layer Security (TLS) 1.2 .
<b>TCP/IP</b>	Transmission Control Protocol/Internet Protocol (TCP/IP) é um protocolo de transporte de dados em redes IP que confirma a entrega dos pacotes de um ponto a outro da rede.

Fonte: Adaptado de BroadBandForum (2020)

Uma RPC é uma operação entre um ACS e o CPE. Ela é usada para comunicação bidirecional entre o CPE e um ACS. Isso inclui métodos iniciados por um ACS e enviados a um CPE, bem como métodos iniciados por um CPE e enviados a um ACS. Alguns métodos RPCs comuns estão incluídos abaixo:

- ❑ **GetParameterValues:** O ACS usa esse RPC para obter o valor de um ou mais parâmetros de um CPE
- ❑ **SetParameterValues:** O ACS define o valor de um ou mais parâmetros de um CPE
- ❑ **GetParameterNames:** O ACS usa esse RPC para descobrir os parâmetros acessíveis em um CPE
- ❑ **Inform:** Um CPE envia essa mensagem a um ACS para iniciar uma sessão e enviar periodicamente informações locais

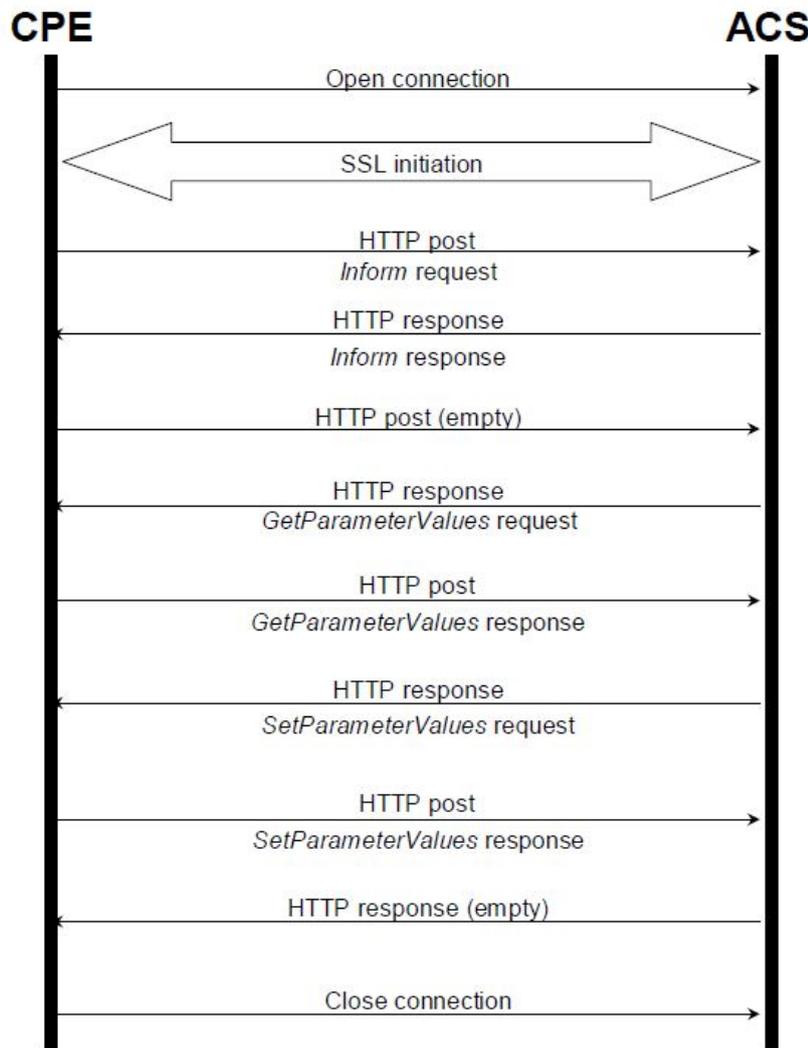
- ❑ **Download:** Quando o ACS requer um CPE para baixar um arquivo especificado para atualizar o firmware e baixar um arquivo de configuração
- ❑ **Upload:** quando o ACS requer um CPE para carregar um arquivo especificado em um local designado.
- ❑ **AutonomousTransferComplete:** informa o ACS sobre a conclusão (bem-sucedida ou malsucedida) de uma transferência de arquivo que não foi especificamente solicitada pelo ACS.
- ❑ **Reboot:** um ACS reinicializa um CPE remotamente quando se encontra uma falha ou precisa de uma atualização de software.
- ❑ **AddObject:** permite que o ACS crie instâncias de objetos disponíveis no CPE, por exemplo, entradas de mapeamento de porta. O ACS também cria os parâmetros e subobjetos associados.
- ❑ **DeleteObject:** permite que o ACS exclua instâncias existentes de objetos disponíveis no CPE. Ele também exclui os parâmetros e subobjetos associados.

Outro ponto importante de se citar é que sempre é o CPE que inicia a conexão após ser "despertado" pelo ACS. Neste ponto são trocadas mensagens de autenticação e segurança para estabelecimento da conexão, conforme a Figura 8. Após este processo, o ACS consegue solicitar os comandos necessários (ex.: Reboot).

Após obter os parâmetros básicos do ACS, o CPE inicia uma conexão Transmission Control Protocol (TCP) com o ACS. Se HyperText Transfer Protocol Secure (HTTPS) for usado, o CPE e o ACS inicializarão o SSL para uma conexão HTTP segura. O CPE envia uma mensagem Inform em HTTP ou HTTPS para iniciar uma sessão CWMP. Após o CPE passar na autenticação, o ACS retorna uma resposta Inform para estabelecer a sessão. Após enviar todas as solicitações, o CPE envia uma mensagem HTTP vazia. Após isso o ACS esta livre para aplicar as configurações na CPE, seja de forma automática, seja de forma manual.

Em resumo, o TR069 e o ACS são fundamentais para o gerenciamento eficiente e automatizado de dispositivos em uma rede de telecomunicações, proporcionando benefícios como redução de custos, aumento da eficiência operacional e melhoria da experiência do usuário final. Cada fornecedor de plataforma ACS fornece sua API, seguindo o modelo padrão da especificação do BroadBandForum (2020) e adicionando customizações em campos livres (se assim julgar necessário). Iremos dar exemplos de chamada de API no próximo capítulo seguinte, onde detalhamos nosso framework.

Figura 8 – Fluxo de mensagens inicial entre MODEM e ACS



Fonte: Adaptado de BroadBandForum (2020)

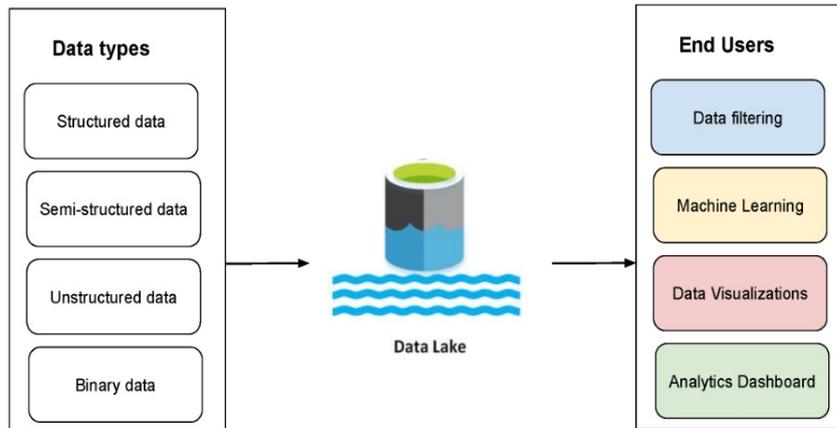
### 2.2.3 Data Lake

Segundo Nambiar e Mundra (2022), com a enorme quantidade de dados disponíveis, tornou-se evidente a necessidade de melhores soluções para armazenar e analisar grandes quantidades de dados semiestruturados e não estruturados para obter informações relevantes e insights valiosos. As abordagens tradicionais de esquema na gravação de dados, como o processo de extração, transformação e carregamento, ou Extract Transform Load (ETL), são muito ineficientes para esses requisitos de gerenciamento de dados. Isso deu origem a outro esquema popular de gerenciamento de dados corporativos modernos, conhecido como DTL

Ainda segundo Nambiar e Mundra (2022), um DTL é um repositório centralizado projetado para armazenar uma grande quantidade de dados em seu estado bruto, sejam eles estruturados, semiestruturados ou não estruturados. A ideia é que se possa armazenar seus dados como estão, sem a necessidade de estruturá-los previamente, e realizar

diferentes tipos de análise - desde visualizações simples até processamento de *big data* e ML. Uma representação de um DTL é dada na Figura 9.

Figura 9 – Arquitetura genérica do DataLake



Fonte: Adaptado de Nambiar e Mundra (2022)

No contexto de ML, os DTLs são particularmente importantes porque fornecem a flexibilidade necessária para armazenar e analisar grandes volumes de dados variados. Isso é crucial para treinar modelos ML, que requerem acesso a vastos conjuntos de dados para aprender e fazer previsões precisas. Além disso, os DTLs suportam a análise em tempo real, o que pode ser usado para orientar decisões de negócios mais informadas e ágeis.

A importância dos DTLs em ML também reside na sua capacidade de:

- ❑ Integrar dados de diversas fontes, o que é essencial para obter uma visão holística e criar modelos mais robustos.
- ❑ Escalar para armazenar e processar quantidades massivas de dados, o que é fundamental para lidar com o crescente volume de informações geradas diariamente.
- ❑ Facilitar o acesso aos dados para cientistas de dados e analistas, permitindo-lhes experimentar e inovar com diferentes técnicas e algoritmos de ML.

Os DTLs permitem que as organizações tirem proveito do potencial dos seus dados, impulsionando avanços em ML e, conseqüentemente, promovendo inovação e competitividade no mercado.

## 2.3 Inteligência Artificial

Segundo Shinde e Shah (2018), IA é um ramo da ciência da computação que busca criar máquinas capazes de simular a inteligência humana, permitindo que sistemas interajam

com o ambiente, aprendam com experiências passadas, se adaptem a mudanças e executem tarefas de forma eficiente.

De acordo com Shinde e Shah (2018), a ideia de criar máquinas inteligentes remonta a muitos anos atrás. No século XVII, René Descartes sugeriu que os animais poderiam ser vistos como máquinas, estabelecendo uma conexão entre mente e corpo. No século XIX, Charles Babbage projetou a Máquina Analítica, considerada um precursor dos computadores modernos. No início do século XX, Alan Turing propôs o conceito de uma “máquina universal” e formulou o famoso “Teste de Turing”.

A IA começou como um campo experimental nos anos 50 com pioneiros como Allen Newell e Herbert Simon. Nas últimas décadas, a IA experimentou um crescimento significativo, impulsionado pelo desenvolvimento de tecnologias cada vez mais avançadas como reconhecimento de fala, visão computacional, processamento de linguagem natural e aprendizado de máquina.

Segundo LeCun, Bengio e Hinton (2015), o avanço rápido da IA nos tempos recentes deve-se ao aumento da capacidade computacional, à disponibilidade de grandes volumes de dados e ao desenvolvimento de novos algoritmos como o algoritmo de aprendizado profundo, ou Deep Learning (DL). Esses fatores têm permitido avanços notáveis em áreas como saúde, finanças, automação e robótica.

A IA está transformando a maneira como vivemos, trabalhamos e interagimos, e continua a ser um campo de rápido desenvolvimento e aplicação em diversas áreas da sociedade.

### 2.3.1 Conceitos de Machine Learning

Segundo Bazazeh e Shubair (2016) ML é um campo da IA que utiliza algoritmos para permitir que as máquinas aprendam a partir de dados e melhorem suas tarefas com experiência. As aplicações abrangem várias áreas, como visão computacional, previsão, análise semântica, processamento de linguagem natural e recuperação de informações. Exemplos incluem reconhecimento de gestos manuais, detecção de intrusões na rede, diagnóstico médico, e sistemas de recomendação.

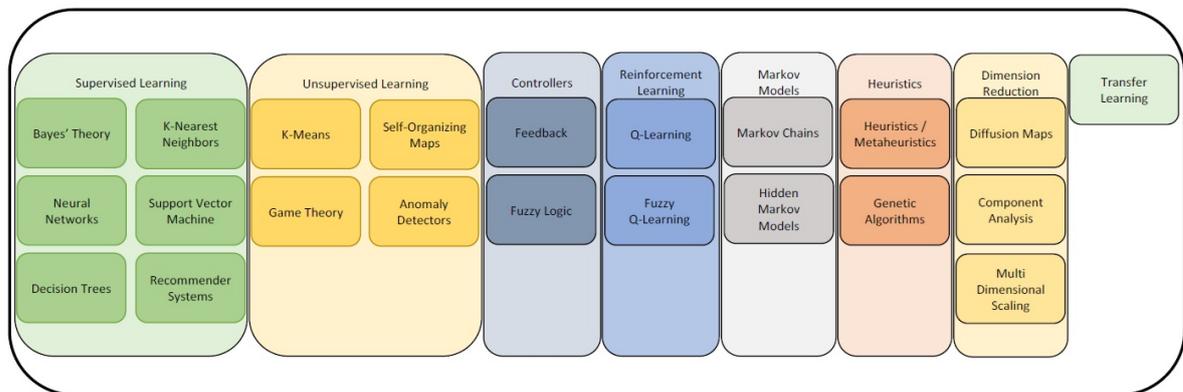
Os conjuntos de dados, conhecidos como *datasets*, são uma coleção de amostras caracterizadas por propriedades (*features*), sendo uma dessas a informação de interesse, chamada de rótulo (*label*). Posto de forma simplificada, o objetivo de um algoritmo de ML é encontrar uma equação, cujas variáveis são as *features*, que prevê o valor do *label*. Percorrendo todo o *dataset*, o processo de treinamento realiza essa tarefa ajustando os pesos (*weights*) da equação por meio de um método iterativo, geralmente minimizando uma função de perda que mede a diferença entre as previsões do modelo e os valores reais dos *labels*. Segundo Valim e Ishii (2023), o resultado deste processo é denominado modelo.

Antes do treinamento, os dados passam por uma fase de pré-processamento, onde são transformados em um formato mais adequado para a modelagem. Ainda segundo Valim e Ishii (2023), existem quatro categorias principais de aprendizado. O aprendi-

zado supervisionado treina modelos com dados rotulados para aprender a relação entre *features* e *labels*, como na classificação de e-mails em *spam* ou não. O aprendizado não supervisionado, por outro lado, trabalha com dados não rotulados, empregando métodos como agrupamento e redução de dimensionalidade para descobrir padrões nos dados. O aprendizado semi-supervisionado mescla elementos dos métodos supervisionados e não supervisionados, usando ambos os tipos de dados. Finalmente, o aprendizado por reforço foca em treinar modelos para tomar decisões, aprendendo mediante recompensas baseadas em suas ações.

A Figura 10 ilustra alguns exemplos de algoritmos de ML classificados por tipo de aprendizado. Em nosso trabalho detalhamos os conceitos de aprendizado supervisionado, não supervisionado e por reforço.

Figura 10 – Tipos de aprendizado e seus algoritmos



Fonte: (KLAINE et al., 2017)

No processo de ML, cada iteração do modelo envolve etapas cruciais de otimização e regularização. Durante estas fases, técnicas são aplicadas para minimizar os erros do modelo. Um aspecto importante nesta etapa é a prevenção do sobreajuste ou *overfitting*, que ocorre quando um modelo se ajusta excessivamente aos dados de treinamento, prejudicando sua habilidade de generalizar para novos dados. Para evitar isso, o modelo é treinado, validado e testado usando conjuntos de dados distintos: treinamento e teste. O conjunto de treinamento é usado para ajustar os parâmetros do modelo. Já o conjunto de teste é usado para avaliar o desempenho do modelo após o treinamento, fornecendo uma medida da eficácia do modelo em condições reais.

Na subseções 2.3.1.1, 2.3.1.2 e 2.3.1.3, são mostrados alguns exemplos de tipos de aprendizado e seus algoritmos.

### 2.3.1.1 Aprendizado Supervisionado

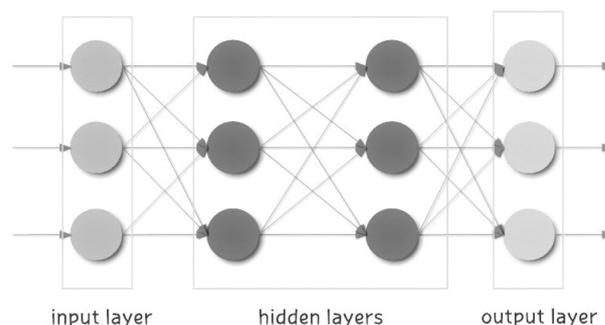
No aprendizado supervisionado, o algoritmo é treinado em um conjunto de dados rotulado, o que significa que cada exemplo de treinamento é emparelhado com a saída

correta. O algoritmo faz previsões e se ajusta com base na precisão de suas previsões em relação aos rótulos pré-fornecidos (BAZAZEH; SHUBAIR, 2016). A seguir são mostrados 3 exemplos de algoritmos de aprendizado supervisionado.

- ❑ **Árvore de Decisão:** Um algoritmo que modela decisões e suas possíveis consequências, criando uma estrutura em forma de árvore. É útil para problemas de classificação e regressão.
- ❑ **Gradient Boosting:** Uma técnica de ML para problemas de regressão e classificação, que produz um modelo de previsão na forma de um conjunto de modelos de previsão fracos, tipicamente árvores de decisão.
- ❑ **Artificial Neural Network (ANN):** As Redes Neurais Artificiais (RNA) são inspiradas pelo funcionamento do cérebro humano e consistem em neurônios e sinapses organizados em camadas. O DL é um subcampo de ML que utiliza redes neurais com muitas camadas ocultas, permitindo que o modelo aprenda representações de dados em vários níveis de abstração. O processo de aprendizado se desdobra em duas fases cruciais: *Forward Propagation*, onde os dados de entrada são processados sequencialmente pelas camadas da rede e o *Backpropagation* onde o erro calculado é propagado de volta pela rede, permitindo o ajuste dos pesos para minimizar o erro de predição.

Um exemplo de rede neural artificial é o MLP, é uma classe de rede neural que pode modelar problemas não-lineares complexos ao aprender uma função não-linear aproximada para a classificação ou regressão. O DL, neste caso, é particularmente poderoso na resolução de problemas não-lineares, pois pode capturar relações complexas nos dados através de suas múltiplas camadas e grande quantidade de parâmetros, tornando-o adequado para tarefas como reconhecimento de imagem e processamento de linguagem natural. A Figura 11 ilustra um exemplo de arquitetura de uma RNA.

Figura 11 – Rede Neural Artificial



Fonte: Adaptado de Tang (2019)

### 2.3.1.2 Aprendizado Não Supervisionado

Diferentemente do supervisionado, no aprendizado não supervisionado, os dados não são rotulados (BAZAZEH; SHUBAIR, 2016). O algoritmo identifica os padrões e a estrutura dos dados sem qualquer orientação explícita. O algoritmo *K-means*, que agrupa os dados em K número de conjuntos distintos com base na distância dos pontos de dados para o centro do cluster é um exemplo deste tipo.

### 2.3.1.3 Aprendizado por reforço

No aprendizado por reforço, o algoritmo aprende a realizar uma tarefa simplesmente tentando e errando. Ele recebe recompensas por ações corretas e penalidades por ações erradas, e seu objetivo é maximizar a recompensa total (VALIM; ISHII, 2023). Como exemplo de algoritmo, podemos citar o *Q-learning*, que aprende a qualidade das ações, indicando quão útil uma ação é para alcançar um objetivo futuro.

## 2.3.2 Bibliotecas de Inteligência Artificial em Python

Python é uma linguagem de programação de alto nível, interpretada e de tipagem dinâmica. Segundo Sarkar, Bali e Sharma (2018), a linguagem é conhecida por sua simplicidade e legibilidade, o que facilita a escrita de código limpo e compreensível. Essas características tornam Python uma escolha popular para desenvolvedores, especialmente na área de IA, onde a experimentação rápida e a iteração são cruciais. Entre os motivos dessa popularidade podemos citar:

- ❑ Bibliotecas ricas: Python possui um vasto ecossistema de bibliotecas e frameworks que simplificam tarefas complexas de IA e ML.
- ❑ Comunidade forte: Uma grande comunidade de desenvolvedores contribui com ferramentas e documentação, facilitando o aprendizado e a resolução de problemas.
- ❑ Flexibilidade: Python permite prototipagem rápida e pode ser integrado com outras linguagens para otimizar partes do código.
- ❑ Suporta múltiplos paradigmas de programação: Ela suporta programação orientada a objetos, programação estruturada, programação funcional e até mesmo programação orientada a aspectos. Essa versatilidade permite que a linguagem seja usada por uma infinidade de programadores.

De acordo com Sarkar, Bali e Sharma (2018), para o uso de Python em aplicações de ML, existem diversas bibliotecas já implementadas e que auxiliam na criação de código mais rapidamente. São em geral mantidas pelas comunidades de desenvolvimento de

software, e seu uso é grande no meio acadêmico. Dentre as bibliotecas de ML em Python, podemos citar:

- ❑ Keras: Uma biblioteca de alto nível para redes neurais, que é conhecida por sua facilidade de uso e permite a construção de protótipos de modelos de ML rapidamente. Keras é um framework de DL de alto nível para Python, que é capaz de rodar em cima de Theano e Tensorflow. Desenvolvido por Francois Chollet, a vantagem mais importante do uso do Keras é o tempo economizado por suas APIs de alto nível fáceis de usar, mas poderosas, que permitem prototipagem rápida para uma ideia. O Keras nos permite usar as construções oferecidas pelo Tensorflow e Theano de uma maneira muito mais intuitiva e fácil de usar, sem escrever código clichê em excesso para construir modelos baseados em redes neurais. Esta facilidade de flexibilidade e simplicidade é a principal razão para a popularidade do keras.
  
- ❑ NumPy: Oferece suporte a *arrays* e matrizes multidimensionais, além de uma coleção de funções matemáticas para operações com esses *arrays*. Numpy é a espinha dorsal do ML em Python. É uma das bibliotecas mais importantes em Python para cálculos numéricos, sendo a principal biblioteca de programação de *arrays* para a linguagem Python (HARRIS et al., 2020). Ele adiciona suporte para matrizes multidimensionais e operações vetorizadas rápidas. A atual biblioteca NumPy é uma sucessora de uma biblioteca antiga, a Numeric, que foi criada por Jim Hugunin e alguns outros desenvolvedores. Travis Oliphant, presidente e co-fundador da Anaconda, tomou a biblioteca Numeric como base e adicionou muitas modificações, para lançar a atual biblioteca NumPy em 2005. É um grande projeto de código aberto e é uma das bibliotecas Python mais populares. Ele é usado em quase todas as bibliotecas de ML e computação científica. A extensão da popularidade do NumPy é verificada pelo fato de que as principais distribuições de sistemas operacionais, como Linux e MacOS, agrupam o NumPy como um pacote padrão em vez de considerá-lo como um pacote complementar.
  
- ❑ Pandas: Utilizada para manipulação e análise de dados. Oferece estruturas de dados como *DataFrames*, que são essenciais para a organização de dados em tabelas e para a realização de operações de pré-processamento. Pandas funciona como um conjunto intuitivo de ferramentas para realizar operações em qualquer tipo de dados. O trabalho inicial para pandas foi feito por Wes McKinney em 2008, enquanto ele era desenvolvedor na AQR Capital Management. Desde então, o escopo do projeto Pandas aumentou muito e se tornou uma biblioteca popular para cientistas de dados em todo o mundo. O Pandas permite que se trabalhe com dados transversais e dados baseados em séries temporais.

- Scikit-learn (Sklearn): Uma das bibliotecas mais populares para ML. Fornece ferramentas simples e eficientes para análise de dados e modelagem preditiva. Inclui algoritmos de classificação, regressão, clustering e redução de dimensionalidade. Scikit-learn é um dos frameworks Python mais importantes e indispensáveis para Ciência de Dados e ML em Python. Ele implementa uma ampla gama de algoritmos de ML cobrindo as principais áreas do ML, como classificação, *clustering*, regressão e assim por diante. Todos os algoritmos de ML convencionais, como Support Vector Machines (SVM), regressão logística, *random forest*, *K-means*, clustering hierárquico e muitos outros, são implementados de forma eficiente nesta biblioteca. Além disso, sua API fácil de usar e padrões de design de código também foram amplamente adotados em outros frameworks. O projeto scikit-learn foi iniciado como um projeto de verão de código do Google por David Cournapeau. O primeiro lançamento público da biblioteca foi no final de 2010. É um dos projetos Python mais ativos e ainda está em desenvolvimento ativo com novos recursos e aprimoramentos existentes sendo adicionados constantemente. Scikit-learn é escrito principalmente em Python, mas para fornecer um melhor desempenho, parte do código principal é escrito em Cython. Ele também usa *wrappers* em torno de implementações populares de algoritmos de aprendizagem, como regressão logística e SVM.

Essas bibliotecas são amplamente utilizadas devido à sua robustez, eficiência e facilidade de integração em projetos de IA e ML.

## 2.4 Self-Organizing Networks

De acordo com Zhang, Zhu e Hossain (2022), com o desenvolvimento das redes celulares em direção ao 5G e além, as redes estão evoluindo para estruturas mais complexas, caracterizadas pela heterogeneidade e implantação densa. Nessas redes, os métodos tradicionais (não automatizados) para implantação, configuração, otimização e manutenção de redes serão ineficientes e incorrerão em enormes despesas operacionais e de manutenção. Isso levou ao conceito de rede auto-organizável, ou Self-Organizing Network (SON), defendido pelo 3rd Generation Partnership Project (3GPP) e pela Next Generation Mobile Networks (NGMN).

As SONs são sistemas de telecomunicações que utilizam algoritmos e IA para gerenciar e otimizar automaticamente o desempenho da rede. O propósito das SONs é reduzir a intervenção humana, melhorar a eficiência operacional e garantir uma experiência de usuário consistente e de alta qualidade (KLAINÉ et al., 2017).

De acordo com Osterbo e Grondalen (2012), SON é um conjunto de procedimentos (ou funções) para configuração automática, otimização, diagnóstico e autocura de redes. É considerado uma necessidade crucial nas redes de telecomunicações, principalmente

devido às possíveis economias em investimentos (Capital Expenditure (CAPEX)) e custos operacionais (Operational Expenditure (OPEX)).

A família SON consiste em uma variedade de funções que são diferentes e cuja maturidade está em diferentes níveis. À medida que o número de funções SON aumenta com cada nova versão de software dos fornecedores de equipamentos de telecom, uma das principais questões para os operadores será determinar quais funções introduzir e também determinar o momento apropriado para ativar essas funções, a fim de obter uma rede bem-comportada e econômica.

Segundo Osterbo e Grondalen (2012), as redes SONs podem atuar da seguinte maneira:

- ❑ Auto-configuração: Configura automaticamente os parâmetros da rede quando novos elementos são adicionados.
- ❑ Auto-otimização: Ajusta continuamente os parâmetros da rede para melhorar o desempenho.
- ❑ Autoproteção: A capacidade do sistema de se proteger de ataques externos e internos
- ❑ Auto-cura (SH): Detecta e resolve problemas na rede de forma autônoma.

Entre as aplicações das SONs podemos citar as Redes móveis (Third Generation of Cellular Mobile Network (3G), 4G, 5G e além), as Redes de sensores sem fio, as redes de IoT, as Redes de comunicação de dados, Redes de banda larga, entre outras (KLAINÉ et al., 2017).

### 2.4.1 Self-Healing

A ação de autocura (SH) é um mecanismo crítico dentro das SONs que permite à rede detectar falhas, como interrupções ou degradação do desempenho, e tomar medidas corretivas automaticamente sem intervenção humana. Isso é alcançado através de monitoramento contínuo, diagnóstico e reconfiguração da rede.

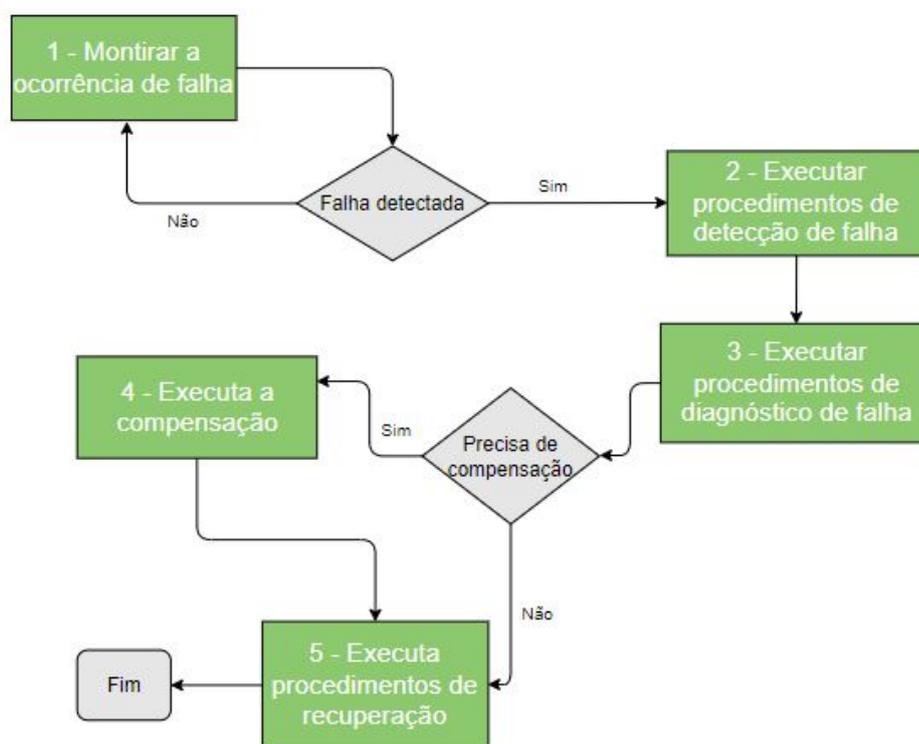
De acordo com Montani e Anglano (2008) e Omar et al. (2021), um sistema de auto-cura deve ser capaz de:

1. Monitorar seu próprio comportamento para detectar falhas na entrega de serviços;
2. Analisar essas falhas para diagnosticar o que as causam;
3. Planejar estratégias adequadas de remediação de falhas
4. Executar planos para restaurar o comportamento normal do sistema.

Esses quatro passos dão origem ao chamado ciclo de auto-cura, controlando um elemento gerenciado e fazendo com que ele retorne ao seu estado normal de funcionamento.

A Figura 12 ilustra as etapas de um sistema de SH. É importante citar que a etapa de Monitoramento de falha acontece normalmente em um *loop* infinito, podendo também acontecer em um período pré-determinado pela solução, de acordo com o tipo de equipamento e rede utilizados. Observamos também que uma vez detectada a falha, também é importante fazer uma análise do motivo da falha, identificando qual tipo de ação é necessária. Eventualmente, antes de corrigir a causa-raiz do problema, pode ser necessária a realização de uma compensação no sistema, para que o elemento de rede volte mais rapidamente ao seu estado normal.

Figura 12 – Fluxo genérico de Self-Healing



Fonte: Adaptado de Omar et al. (2021)

Entre os benefícios de SH conhecidos, espera-se:

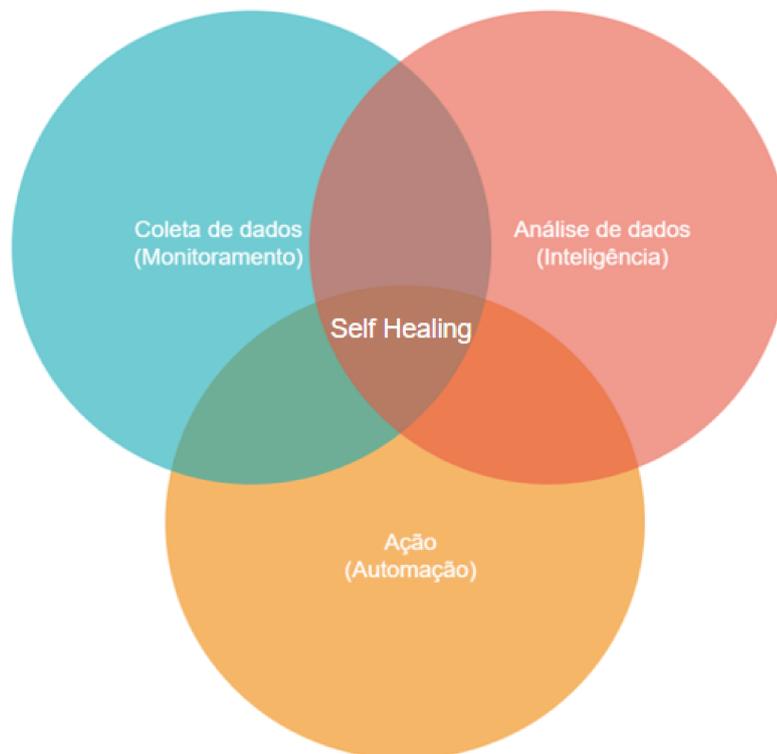
- ❑ **Confiabilidade aprimorada:** Minimiza o tempo de inatividade e interrupções, garantindo a operação contínua de sistemas críticos.
- ❑ **Eficiência operacional:** Reduz a necessidade de intervenção manual e solução de problemas.
- ❑ **Economia de custos:** Evita perdas associadas ao tempo de inatividade da rede e reduz os custos de manutenção com mecanismos de reparo automatizados.

As SONS e suas ações de auto-cura permitem que as redes sejam mais adaptáveis, resilientes e capazes de suportar a crescente demanda por serviços de comunicação de alta qualidade.

### 2.4.2 Aspectos de uma arquitetura de Self-Healing

A Figura 13 consegue cobrir de maneira resumida o que se deve observar para a implantação de uma arquitetura de SH em redes de telecomunicações, o que pode ser uma possível alternativa às atividades manuais de Operations and Maintenance (OAM) das redes.

Figura 13 – Aspectos de uma arquitetura de Self Healing

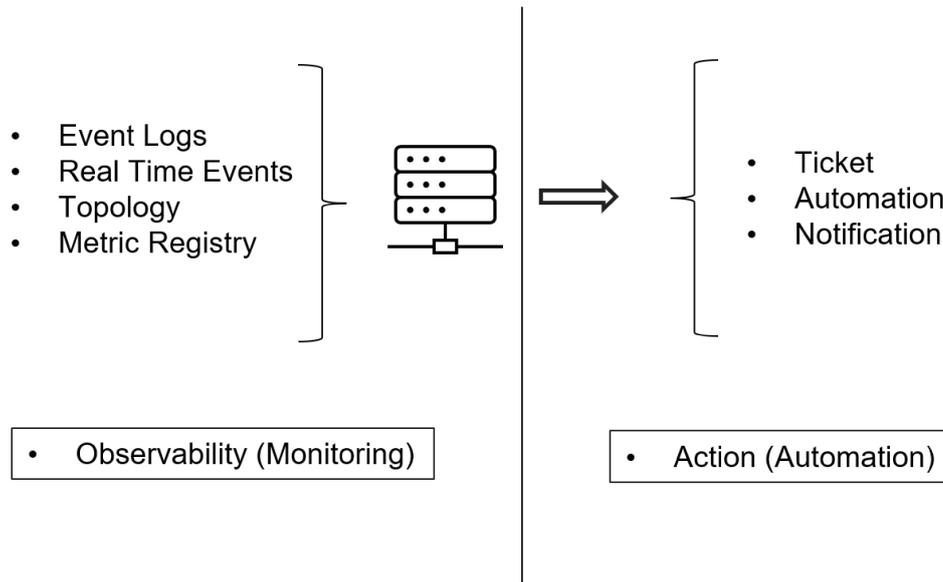


Fonte: O autor

Segundo, RedHat (2022), a coleta de dados da rede é compreendida pelo Domínio da Monitoração. Isso inclui coleta do histórico de eventos e alertas da rede assim como a coleta dos alertas em tempo real. As formas de coleta podem ser diversas, como coleta elemento a elemento ou unificação de informações em base de dados unificada, como um DTL ou uma base de Information Technology Service Management (ITSM). Em seguida passa-se para o Domínio de Análise e tomada de decisão, onde podemos ter regras de tratativas de incidentes pré-configuradas ou criadas em tempo real, se estivermos usando o auxílio de técnicas de IA como o ML. Uma vez definida uma ação a ser tomada, passa-se para o Domínio da Ação/Automação, que pode incluir a execução de um comando pré-determinado em um equipamento da rede ou então a abertura de um chamado (ticket) para registro e análise posterior. A figura 14 ilustra estes termos.

É importante destacar o papel da ML na otimização desses sistemas. A aplicação de ML permite que as redes não apenas detectem e corrijam falhas, mas também antecipem

Figura 14 – Domínios de ação do Self Healing



Fonte: Adaptado de RedHat (2022)

e previnam incidentes antes que eles ocorram.

A implementação de ações de SH em redes de telecomunicações deve considerar:

- ❑ Modelos Cognitivos: Utilização de modelos de ML para diagnóstico e atuação automática.
- ❑ Prevenção de Incidentes: Capacidade de antecipar falhas e tomar medidas proativas.
- ❑ Integração Tecnológica: Aplicação de novas tecnologias de automação e APIs para suportar a auto-cura.

Essas abordagens representam um avanço significativo na gestão de redes, proporcionando uma maior adaptabilidade e resiliência, essenciais para atender à demanda crescente por serviços de comunicação de alta qualidade e confiabilidade.

## 2.5 Trabalhos Relacionados

Nesta seção, discutiremos trabalhos relacionados e observaremos oportunidades de pesquisa para evolução. Organizamos os trabalhos de acordo com a data de sua publicação.

Em Kiciman e Fox (2005), os autores usaram algoritmos de DT para identificar falhas em redes corporativas. Eles utilizaram um conjunto de dados genérico e classificaram as situações como normais ou anômalas, associando isso com os componentes de hardware e

software correlacionados. Apesar disso, eles não aplicam conceitos de SH nem classificam as situações quase em tempo real.

O trabalho de Wang et al. (2007), utiliza ML com técnicas de aprendizagem supervisionada, como DT e SVM em uma rede de sensores sem fio, utilizando um conjunto de dados de 30 sensores. Eles geram estimativas de qualidade do link para melhorar falhas predição. Contudo, este trabalho não aplicou os conceitos de SH nem dados reais de operadores de telecomunicações.

No que diz respeito ao gerenciamento de redes, os autores de Xu et al. (2016), adotando o conceito de computação inteligente, propõem uma arquitetura inteligente de rede autônoma baseada na Virtualização de Funções de Rede (Network Functions Virtualization (NFV)), capaz de alcançar Qualidade de Serviço (QoS) e eficiência operacional adequadas. Esta arquitetura, chamada CogNet (XU et al., 2016), usa algoritmos de ML para adaptar recursos dinamicamente aos requisitos imediatos de NFVs, minimizando degradações de desempenho de aplicativos. Assim como no CogNet, nosso trabalho utilizará ML para análise de dados e posterior sugestão da melhor ação para ser trabalhado. No entanto, a CogNet só aplica a autogestão a Estruturas NFV genéricos, não em elementos reais de rede de banda larga.

O artigo de Hashmi et al. (2017) trabalha com aprendizagem não supervisionada em um serviço de internet banda larga provedor para melhorar o processo de detecção de falhas usando registros históricos de uma rede operador. Entretanto, eles não propõem uma solução quase em tempo real para este processo.

Em Rahmani et al. (2020), os autores testaram sete métodos SH para uma rede 4G. Eles perceberam que métodos de classificação não supervisionados como ADABOOST, e *Fuzzy Logic* baseada em *big data* se portaram bem, trazendo bons diagnósticos às causas raízes das falhas em uma rede móvel. Por outro lado, no presente trabalho são adotados métodos de aprendizagem supervisionada, atendendo às características dos dados disponíveis para a fase de treinamento.

O trabalho de Silva, Morais e Silva (2020) utiliza dados reais de uma operadora de telecomunicações para fazer análises de previsão de tráfego de um concentrador de banda larga - Broadband Network Gateway (BNG). Este trabalho classifica e detecta dados inconsistentes de um relatório de falha de campo de uma rede de banda larga. Nossa abordagem também usa dados reais de uma operadora de telecomunicações, mas aplica SH associado a ML aos dados da operadora rede.

Em Bothe et al. (2020), Redes Neurais Convolucionais (Convolutional Neural Networks (CNN)) são aplicadas ao diagnóstico de falhas em redes de telecomunicações. O estudo compara soluções usando o algoritmo de Naive Bayes, CNNs e o algoritmo *Random Forest*. Seus resultados mostram que as CNNs têm melhor desempenho do que os outros métodos estudados. Porém, apesar de trabalhar com incidentes de rede em um conjunto de dados adaptado entre dados reais e interpolados, este trabalho não aplicou o conceito de SH.

Segundo Omar et al. (2021), um ponto importante a ser considerado na realização de um processo de SH é trabalhar na parte de coleta de dados. O trabalho identifica que as telecomunicações redes têm quatro características marcantes que podem ser difíceis de lidar com automação: volume (grandes quantidades), rapidez (alta frequência de coleta), variedade, e valor (informações ocultas nos dados). Em Omar et al. (2021), um grupo de bancos de dados é usado em um único banco de dados para consulta, o que auxilia no aprendizado inteligente do agente. Apesar de estudar as características dos dados de redes de telecomunicações, Omar et al. (2021) não utiliza dados reais dos operadores para validar seu modelo.

A Tabela 3 compara os trabalhos relacionados discutidos acima, identificando cada abordagem, características e parâmetros utilizados.

Tabela 3 – Trabalhos Relacionados

Work	SH	ML	SH em GPON	Dados Reais	RT	Rede	Tipo de SH	Técnica de ML	Parâmetros
(KICIMAN; FOX, 2005)	Não	Sim	Não	Não	Não	Data Center	Localiz. de Falhas	Superv.: DT	HTTP Error
(WANG et al., 2007)	Não	Sim	Não	Não	Não	Sem Fio	Predição de Falhas	Superv.: DT,SVM	Signal strength, Channel load
(XU et al., 2016)	Sim	Sim	Não	Não	Sim	Celular	Gestão de Recursos	Sup.: ANN	QoS, Energy cons., VNF Load
(HASHMI et al., 2017)	Sim	Sim	Sim	Sim	Não	Banda Larga	Detecção de Falhas	Não-Sup.: k-Means, FCM · SOM	Fault time and cause, Region, MTTR
(RAHMANI et al., 2020)	Sim	Sim	Não	Não	Não	Celular	Diagnóst. de Falhas	Não-Sup.: SVM, Fuzzy, Adaboost	Cell Power, Avg speed, Signal Quality
(SILVA; MORAIS; SILVA, 2020)	Não	Sim	Não	Sim	Não	Banda Larga	Pred. tráfego, Detecção Anomalias	Não-Sup.: SOM Sup: RNN	OLT RX Power, ONT RX Power, ONU dist.
(BOTHE et al., 2020)	Sim	Sim	Não	Sim	Sim	Celular	Diagnóst. de Falhas	Superv.: RF, CNN, ANN.	Cell Outage, TxPower, Antenna Tilt
(OMAR et al., 2021)	Sim	Sim	Não	Não	Não	Celular	Detecção de Falhas e SH Automático	Superv.: DT	Max. data rate, #users per BS and SNR

SH - Self-healing, ML - Machine Learning, GPON - Gigabit Passive Optical Network, RT - Real Time

Fonte: O Autor

A coluna SH classifica se o trabalho aborda o conceito de SH. A coluna ML classifica se o trabalho aplica o conceito de ML. A próxima coluna mostra se o trabalho considera

SH aplicado a redes de banda larga GPON. A coluna a seguir mostra se o trabalho utiliza dados reais de operadoras de telecomunicações em seu experimento. A próxima coluna indica se o artigo usa um processo em tempo real para coletar os dados. Em seguida temos as colunas que identificam o tipo de rede estudado e o domínio de SH trabalhado. Na sequência, a coluna Técnica de ML nos diz se o trabalho usou aprendizado de máquina do tipo supervisionado ou não-supervisionado, especificando o tipo de algoritmo utilizado. Finalmente, a coluna Parâmetros indica que tipos de variáveis de rede foram trabalhadas em cada artigo.

Em resumo, podemos citar que:

1. Os *datasets* utilizados são em geral fictícios para banda larga, não se encontrando arquivos com dados reais de maneira pública
2. Poucos são os estudos e aplicações de SH em tempo real
3. Raros são os estudos de SH em redes de fibra GPON
4. Existem poucos trabalhos no domínio de detecção de falhas e SH automático
5. Existe uma forte lacuna em pesquisas nos parâmetros de qualidade do MODEM do usuário de banda larga, que determinam uma boa experiência do cliente (Customer Experience (CX)).

---

## Framework

O presente trabalho especifica um Framework para implementação de SH cognitivo em redes GPON. Esta abordagem pretende ser suficientemente genérica para que qualquer operador de telecomunicações possa implementá-lo, trazendo os benefícios já mencionados na introdução deste trabalho, como a melhoria do serviço e a satisfação dos usuários.

### 3.1 Visão Geral

Uma das principais reclamações dos usuários de internet banda larga fixa é a baixa velocidade da internet. O usuário geralmente não sabe se o problema está na rede da operadora, se está no roteador Wi-Fi ou se é algum problema específico do MODEM. Segundo a operadora Algar Telecom, o MODEM e o Wi-Fi representam em conjunto 42% do total de reclamações de performance dos usuários em 2023.

Quando se fala de reclamações neste trabalho, entenda-se como o tipo de reclamação por baixa velocidade da internet. Por exemplo, um usuário que reclama que está tendo uma experiência ruim de uso da internet por ter uma velocidade menor que a contratada junto a operadora.

Após análise das informações do sistema de relacionamento com o cliente (CRM) da operadora, percebeu-se, através do cruzamento das informações de reclamações dos clientes com os parâmetros do seu MODEM (ver Seção 2.1.5), que havia um padrão indicando que uma reclamação ocorria quando uma combinação de parâmetros do MODEM estava fora dos limites aceitáveis para uma boa experiência do usuário.

Para o Framework, os dados do CRM devem ser coletados e filtrados para determinar se um cliente reclamou, e combinados com os parâmetros reais da rede, vindos da gerência da rede de banda larga (NMS). Assim, pode-se perceber que, quando o usuário reclamou, os parâmetros do seu MODEM estavam fora dos limites aceitáveis para um bom indicador de *Quality of Experience (QoE)*.

Os resultados são armazenados em um novo banco de dados (DTL) que guiará o agente inteligente nos processos de aprendizagem e tomada de decisão automática sem

intervenção humana. Para esta estrutura, assumimos que o NMS pode exportar dados para o DTL em um modelo Near Real Time (NRT).

A Figura 15 identifica o framework proposto.

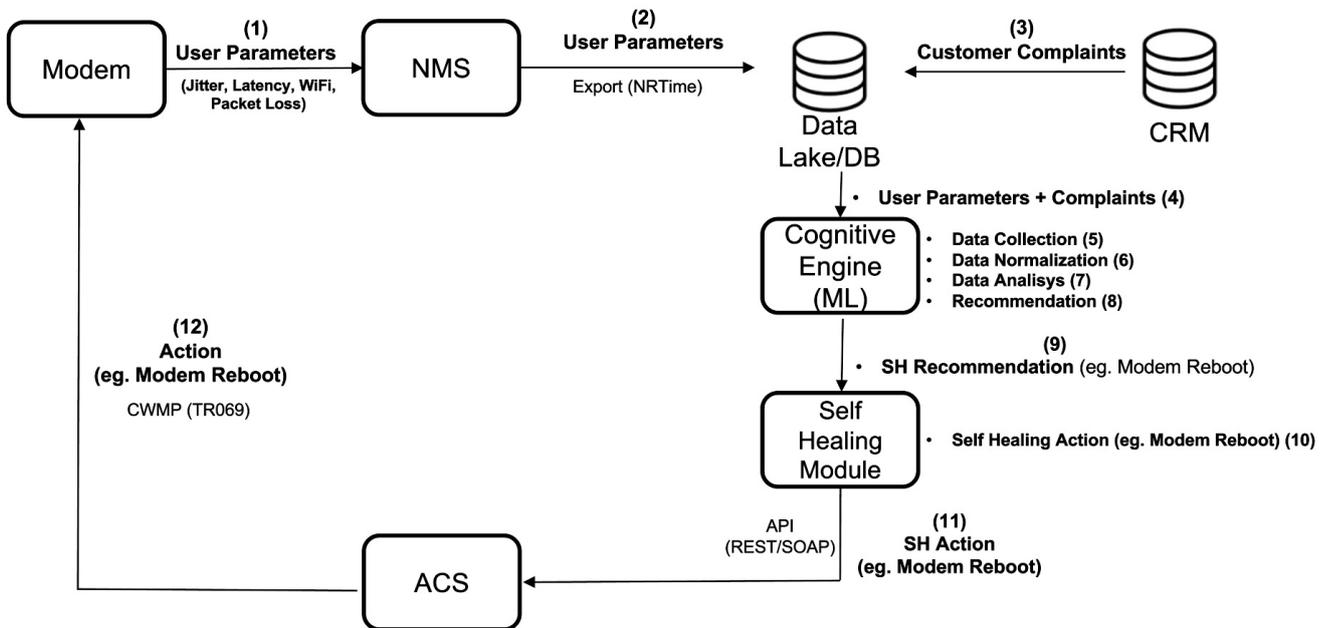


Figura 15 – Framework de SH Cognitivo em redes de Banda Larga GPON

O Framework inclui a implementação de um modelo ML treinado com dados normalizados do DTL usando aprendizagem supervisionada para análise de dados. No próximo capítulo mostraremos os três métodos de ML testados e iremos avaliar qual será mais preciso para identificar se o cliente iria reclamar. Depois que o modelo for treinado, ele poderá ser usado para analisar os parâmetros do usuário e prever se ele reclamará.

Uma vez identificado que o cliente iria reclamar, ações de SH deverão ser tomadas. Na prática, é proposto um módulo SH, capaz de chamar a API de uma plataforma ACS que suporte o CWMP em sua especificação TR069, definida em BroadBandForum (2020), e essa plataforma dá o comando necessário para corrigir a falha no MODEM.

A plataforma ACS é muito comum nas operadoras. Ela é usada hoje pelas equipes de atendimento ao cliente das operadoras para lidar de forma reativa com reclamações de clientes com comandos como reinicialização do modem e mudança de canal Wi-Fi.

A seguir detalhamos cada módulo de nosso framework. Os módulos são: Módulo de Rede, Módulo de Sistema, Módulo de Coleta de Dados, Módulo Cognitive Engine, Módulo SH e Módulo de Correção. Na Figura 16 identificamos os módulos do framework proposto.

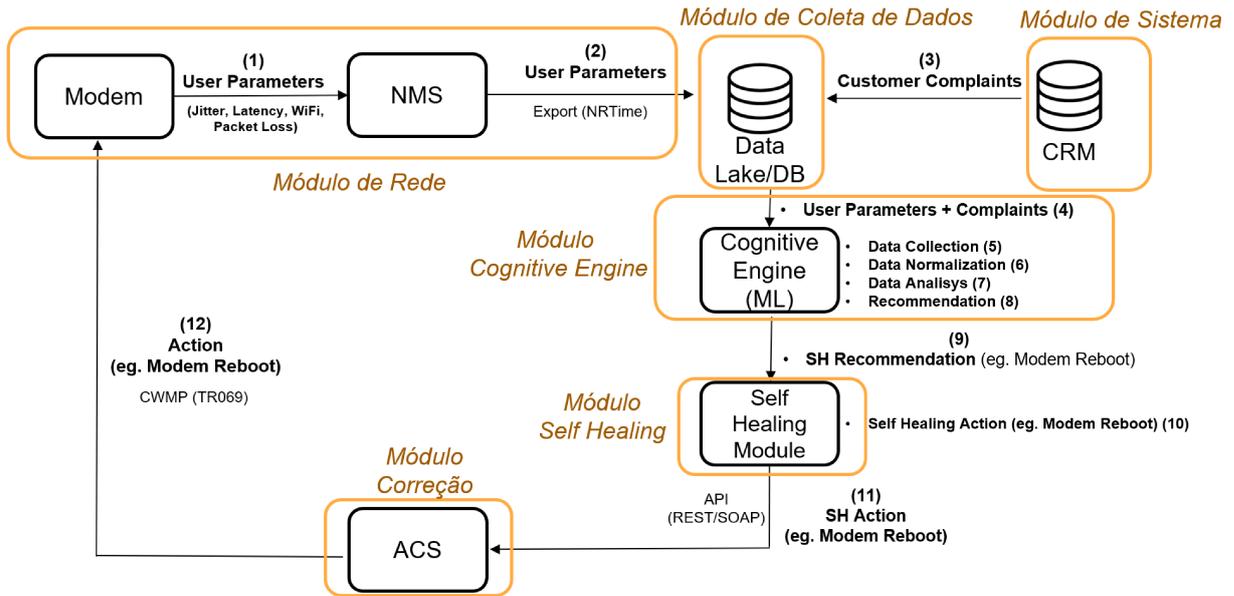


Figura 16 – Módulos do Framework

## 3.2 Módulo de Rede

O Módulo de Rede do Framework é responsável pela captura dos parâmetros de rede do usuário do serviço de banda larga (ver Seção 2.1.5). Além disso, esse módulo formata e disponibiliza os dados do usuário para o Módulo de Coleta de Dados. Este módulo é composto pelos sub-módulos MODEM e NMS, conforme ilustrado na Figura 17. Podemos considerar que a *entrada* do módulo de rede é a experiência do usuário, que é refletida nos parâmetros de rede (jitter, latência, perda de pacotes, etc).

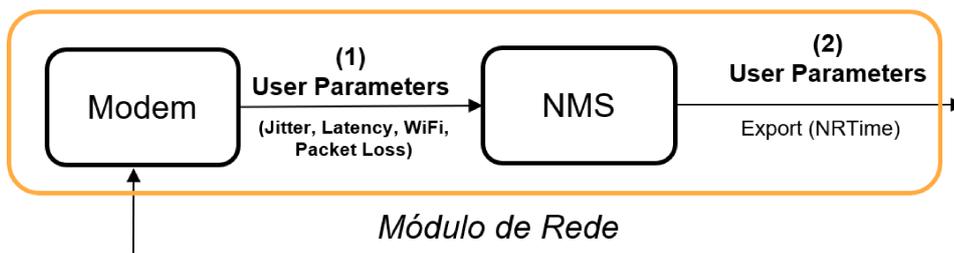


Figura 17 – Módulo de Rede e seus componentes

### 3.2.1 Sub-Módulo MODEM

O sub-módulo MODEM compreende o equipamento de acesso do usuário do serviço de banda larga. O MODEM fica no ambiente do usuário, seja numa casa ou empresa. Neste equipamento, temos uma interface via fibra óptica, por onde chegam os dados de internet, assim como interfaces Wi-Fi e Local Area Network (LAN), por onde o usuário acessa os

serviços de internet a partir de um celular, computador, Smart TV, ou outro equipamento que tenha estes tipos de conectividade. É importante ressaltar que escolhemos o MODEM para fazer parte do módulo de rede pois, além de este ser a principal causa de reclamações de usuários nas operadoras, conforme citado na seção 3.1, a grande maioria dos usuários possui MODEM com Wi-Fi embutido, englobando portanto estas fontes de reclamações neste elemento. Para os casos onde o usuário possui um roteador Wi-Fi separado do MODEM, assumimos a premissa de que também é possível acessar o roteador de maneira remota, através de uma plataforma ACS.

No diagrama da Figura 17, o MODEM envia os parâmetros do usuário (*Mensagem 1*) para o módulo NMS, que coleta estes dados.

É importante ressaltar que o fornecedor que implementa o MODEM deve seguir a norma (G.984 - ITU - International Telecommunication Union, 2008). Na seção 2.1.3, foram exemplificados estes fornecedores.

Além disso, a interface entre o MODEM e o NMS normalmente é implementada em um protocolo IP proprietário. Entretanto, isso não é um problema para Framework porque o gerenciamento dos MODEMs via NMS é obrigatório para se operar uma rede de banda larga fixa.

Os parâmetros do usuário saem do MODEM, passam pela OLT, e chegam ao NMS.

### 3.2.2 Sub-Módulo NMS

Este sub-módulo do framework recebe as informações do MODEM e as disponibiliza em banco de dados para visualização pela equipe de gerência de rede da operadora. No NMS, os analistas de operação visualizam os parâmetros do MODEM do cliente em interface gráfica (*Dashboard*), e tomam decisões técnicas sobre qual ação deve ser feita para reestabelecer ou melhorar a experiência do serviço para cada usuário de maneira distinta.

O NMS é implementado pelo fornecedor da rede banda larga, e fica dentro do ambiente da operadora, em uma estrutura *on premises* ou em *cloud*. Na estrutura em *cloud* é recomendado um *link* de internet entre a operadora e o provedor de *cloud* que garanta alta disponibilidade e o mínimo possível de latência, para uma boa performance da solução.

No contexto deste framework, o NMS coleta as informações do MODEM, representadas pela (*Mensagem 1*) e as armazena em seu banco de dados próprio.

Além disso, para a implantação do Framework, assumimos que o NMS pode exportar os parâmetros do usuário (*Mensagem 2*) para o módulo de Coleta de Dados quase em tempo real (NRT). O formato de exportação dos dados pelo NMS mais comuns são o padrão (.CSV), DBLINK e API, além de outros formatos dependendo do seu fabricante.

### 3.3 Módulo de Sistema

Este módulo consiste nos sistemas necessários para se obter informações de negócio do usuário do serviço. O principal sistema que compõe este módulo é o CRM, porém outros sistemas acessórios podem vir a compor este módulo se necessário durante a implementação nas operadoras.

Conforme citado na Seção 2.2.1, o CRM possui informações do usuário do serviço de banda larga como últimas reclamações, pedidos de informações, solicitações de serviço (mudança de endereço, troca de plano) entre outros. Em nosso framework, o módulo de sistema exporta a *Mensagem 3*, contendo as reclamações dos usuários, para o módulo de Coleta de dados. Lembrando que o conceito de reclamação do usuário foi explorado na seção 3.1.

O formato da *Mensagem 3* inclui o ID do usuário e se este reclamou no mesmo período de coleta dos dados de rede.

Como cada CRM pode ter implementado uma chave de usuário de uma forma diferente (pois são vários os fornecedores desta aplicação), o módulo seguinte, de Coleta de Dados, será responsável por adaptar essa mensagem para: ID do usuário + 1/0 (1 = usuário reclamou e 0 = usuário não reclamou).

O módulo de sistema, representado no Framework pelo CRM, é implementado pelo fornecedor de escolha da operadora (ver exemplos na seção 2.2.1), e fica dentro do ambiente da operadora, em uma estrutura *on premises* ou em *cloud*. Na estrutura em *cloud* é recomendado um *link* de internet entre a operadora e o provedor de *cloud* que garanta alta disponibilidade e o mínimo possível de latência, para uma boa performance da solução.

O Módulo de Sistema é ilustrado na Figura 18.

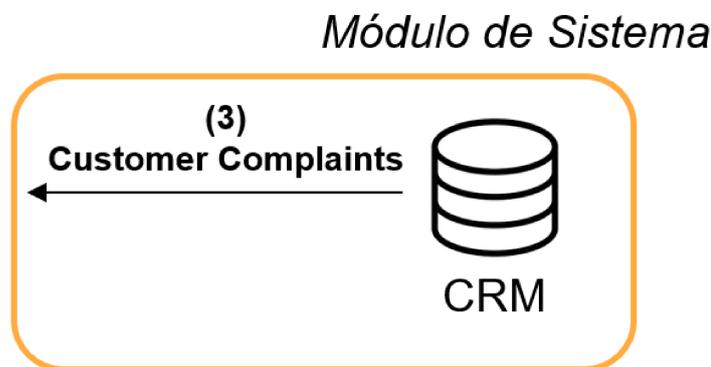


Figura 18 – Módulo de Sistema

Para o caso deste framework, usamos as informações de reclamação do usuário do serviço de banda larga do CRM para compor o módulo de Coleta de Dados, que cruza essas informações com as informações de rede do mesmo usuário.

### 3.4 Módulo de Coleta de Dados - DTL

O módulo de coleta de dados recebe de maneira ativa os dados dos módulos de rede e de sistema. É o módulo que consolida todos os dados necessários para o Framework. O módulo de sistema exporta as reclamações dos usuários (*Mensagem 3*) para este módulo. Além disso, o Módulo de coleta de dados recebe os parâmetros de rede do usuário do serviço (*Mensagem 2*).

Para a implementação do módulo de coleta de dados, podem ser feitos *scripts* em linguagens como Python, Java, ou outra em comum acordo com a arquitetura da TI da operadora, para que estes colem os dados ativamente e armazenem no seu DTL.

O módulo de coleta de dados pode ficar tanto em uma estrutura *on premises*, na operadora, ou em *cloud*. Na estrutura em *cloud* é recomendado um *link* de internet entre a operadora e o provedor de *cloud* que garanta alta disponibilidade e o mínimo possível de latência, para uma boa performance da solução.

Este módulo cruza as informações do módulo de rede com as informações do módulo de sistema, a partir a chave primária do "ID do Usuário". Esses dados (*Mensagem 4*) são importantes para treinar o módulo principal de nosso framework, o módulo Cognitive Engine. O "ID do Usuário" normalmente é identificado nas operadoras como "ID do circuito", pois ele identifica de maneira única o MODEM do usuário.

O formato dos dados coletados do NMS podem ser o padrão (.CSV), DBLINK e API, além de outros formatos dependendo do seu fabricante. O mesmo vale para o módulo de sistema (CRM). Dessa forma, o módulo de coleta de dados atua como um adaptador de dados, que unifica os dados recebidos, padroniza-os e envia para o próximo módulo.

O Módulo de Coleta de Dados é ilustrado na Figura 19

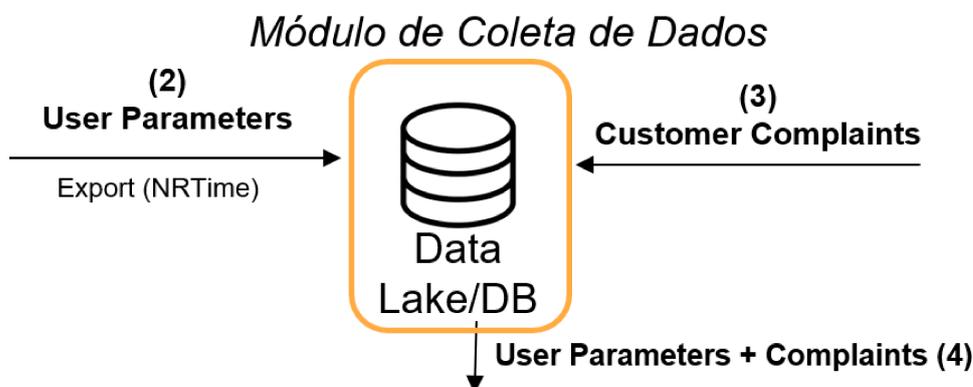


Figura 19 – Módulo de Coleta de Dados

## 3.5 Módulo Cognitive Engine

O módulo Cognitive Engine é um componente inteligente que analisa os dados recebidos do módulo de Coleta de Dados (*Mensagem 4*), treina e testa um algoritmo de ML para identificar onde uma combinação específica de parâmetros de rede do usuário pode se transformar em uma reclamação de uso do serviço de banda larga.

A Figura 20 ilustra os componentes do módulo, destacando as mensagens e ações.

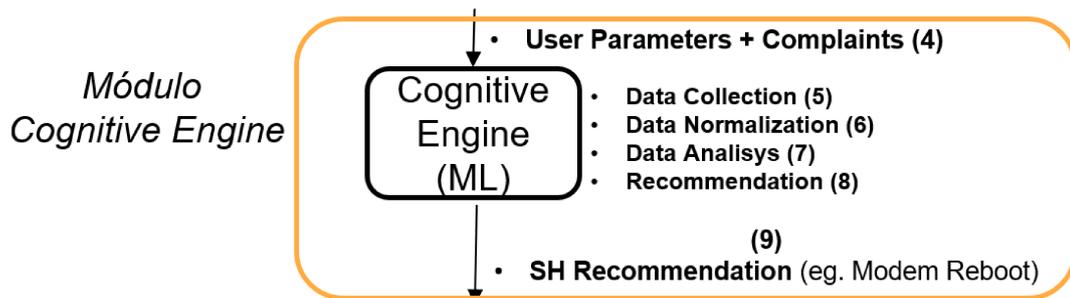


Figura 20 – Módulo Cognitive Engine

Este módulo, uma vez treinado, é responsável por analisar os parâmetros do MODEM de banda larga do usuário e prever se o usuário reclamará.

O módulo Cognitive Engine deve realizar algumas ações importantes. Primeiramente o módulo deve receber os dados do DTL (*Mensagem 5*) e uma vez recebidos todos os dados, passamos para a etapa de normalização dos dados (*Mensagem 6*). Em seguida o módulo analisa os dados formatados (*Mensagem 7*), e faz a recomendação de SH (*Mensagem 8*) se necessário.

A recomendação da ação de SH, por exemplo um *reboot* de MODEM, só deve acontecer caso o algoritmo entenda que, baseado nos parâmetros da experiência do uso do usuário (Jitter, Latência, Perda de Pacotes, Qualidade do canal Wi-Fi e número de dispositivos distantes do MODEM), o mesmo irá reclamar.

Logo, com o uso de técnicas de aprendizado supervisionado (visto que tem-se a saída esperada para treinar o modelo pelo histórico de reclamações do usuário), pode-se recomendar com boa precisão ações de SH automáticas para o módulo SH deste framework (*Mensagem 9*).

É importante citar que pode ser usado qualquer algoritmo de ML para esse framework, desde que seja com aprendizado supervisionado. Na seção 4.3, são apresentados 3 algoritmos que podem implementar o módulo Cognitive Engine.

Uma vez implementado, este módulo pode ficar tanto em uma estrutura *on premises*, na operadora, ou em *cloud*. Na estrutura em *cloud* é recomendado um *link* de internet entre a operadora e o provedor de *cloud* que garanta alta disponibilidade e o mínimo possível de latência, para uma boa performance da solução.

### 3.6 Módulo *Self-Healing*

O módulo de SH, planeja a restauração de serviços, e converte a(s) solução(s) proposta pelo módulo anterior em um plano de reparação/correção.

O módulo de SH é responsável por receber a ação recomendada do Cognitive Engine (em padrão de API ou sincronização de banco de dados), analisar a mensagem, identificar qual tipo de ação precisa ser tomada (*Mensagem 10*) e chamar a plataforma específica que aplicará a ação SH. Esta análise funciona da seguinte maneira: o módulo SH recebe que uma ação do módulo anterior, recomendando, por exemplo, um *reboot* do MODEM de um determinado ID que identifica o usuário do serviço. Este módulo pode avaliar em qual plataforma ACS está este MODEM, a partir de uma consulta via API ao NMS. Isto é necessário quando existe mais de um ACS na rede da operadora. Após esta avaliação, o módulo SH invoca uma API da plataforma que administra o MODEM especificado, com o comando relativo ao sugerido pelo módulo anterior. Estes comandos podem estar implementados numa base de dados específica deste módulo.

Este módulo pode ficar tanto em uma estrutura *on premises*, na operadora, ou em *cloud*. Na estrutura em *cloud* é recomendado um *link* de internet entre a operadora e o provedor de *cloud* que garanta alta disponibilidade e o mínimo possível de latência, para uma boa performance da solução. É importante ressaltar que este módulo deve ter acesso liberado (firewall) para conexão com o módulo seguinte (Módulo de Correção - Plataforma ACS).

A Figura 21 ilustra o módulo de SH.

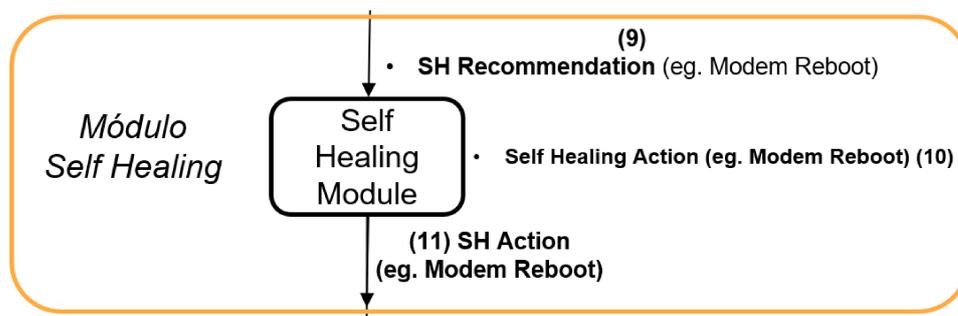


Figura 21 – Módulo de SH

A saída deste módulo é uma chamada de API para o módulo de correção. Esta interface é padronizada em HTTP e implementada pelos fabricantes de plataforma ACS, via Northbound Interface, no caso de redes de banda larga, como mostrado na subseção 2.2.2, na Figura 6.

Como exemplos de possíveis ações deste módulo podemos citar: Reboot de MODEM para melhoria de parâmetros como *Jitter* ou *Latency*, troca de canal Wi-Fi, ou a abertura de uma Ordem de Serviço (OS) para um sistema de Service Order Management (SOM)

para que um técnico de campo possa verificar localmente a integridade da fibra óptica do usuário.

A seguir, na Figura 22 ilustramos um exemplo de chamada de API que o módulo SH pode realizar. Neste comando, solicitamos ao servidor ACS (Módulo de Correção) que reinicie o dispositivo especificado (MODEM).

```
POST /ACS HTTP/1.1
Host: acs.example.com
Content-Type: text/xml; charset=utf-8
SOAPAction: ""

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:cwmp="urn:dslforum-
org:cwmp-1-0">
  <soapenv:Header>
    <cwmp:ID soapenv:mustUnderstand="1">ID:intrnl.unset.id.Reboot1234</cwmp:ID>
  </soapenv:Header>
  <soapenv:Body>
    <cwmp:Reboot>
      <CommandKey>RebootCommand</CommandKey>
    </cwmp:Reboot>
  </soapenv:Body>
</soapenv:Envelope>
```

Figura 22 – Exemplo de chamada de API do módulo SH

A seguir, identificamos cada campo desta chamada de API, e realizamos uma explicação de sua função.

#### **Cabeçalho HTTP:**

- ❑ POST /ACS HTTP/1.1: Solicitação POST para o ACS.
- ❑ Host: acs.example.com: Host do servidor.
- ❑ Content-Type: text/xml; charset=utf-8: Tipo de conteúdo eXtensible Markup Language (XML).
- ❑ SOAPAction: " ": Ação SOAP (vazia neste caso).

#### **Envelope SOAP:**

- ❑ <soapenv:Envelope>: Elemento raiz do envelope SOAP.
- ❑ <soapenv:Header>: Contém o ID da solicitação.

- ❑ <soapenv:Body>: Contém o corpo da mensagem SOAP.

#### Corpo da Mensagem:

- ❑ <cwmp:Reboot>: Ação de reinicialização.
- ❑ <CommandKey>RebootCommand</CommandKey>: Chave de comando para a ação de reinicialização.

Portanto, o módulo SH consegue, a partir do gatilho do módulo cognitivo, acionar o módulo de correção de maneira padronizada, visto que o módulo de correção usa plataformas que usam o protocolo TR069, normatizado pelo BroadBandForum (2020).

Da maneira que foi especificado, o módulo de SH torna-se genérico o bastante para que possa, caso seja necessário, definir ações e comandos para outras redes de telecomunicações além da rede de banda larga de fibra (GPON).

### 3.7 Módulo Correção

O módulo de Correção executa o plano de reparo correspondente à solução selecionada pelo módulo de SH, usando mecanismos adequados específicos da aplicação selecionada.

Em nosso framework, o módulo de correção compreende o ACS. Este módulo é referente à plataforma ACS existente nas operadoras. Normalmente, são usados pelas operadoras fornecedores como AVSYSTEM (2024), INCOGNITO (2024) e AXIROS (2024). Porém, caso a operadora deseje, pode implantar seu próprio ACS usando soluções abertas, como o geniacs (2024).

Ao receber uma chamada API (HTTP / SOAP) do módulo SH (*Mensagem 11*), o módulo ACS dispara uma ação para o sub-módulo MODEM (*Mensagem 12*). Esta ação final usa o protocolo TR069, e realiza a comunicação com o MODEM de maneira automática, com o comando solicitado pelo módulo anterior, fechando o *loop* de SH na rede de banda larga. Como exemplos de ações, podemos citar: Reboot do MODEM, troca de canal Wi-Fi, entre outras.

Para este módulo, conforme recomendado pelo BroadBandForum (2020) assumimos como premissa que:

- ❑ Todos os MODEMs independente do tipo de configuração (*bridge* ou *router*) obtêm um endereço IP para se comunicarem com um ACS.
- ❑ Um MODEM pode interagir com um único ACS por vez.

A figura Fig 23 ilustra o módulo de Correção.

Outro ponto importante a ser ressaltado neste módulo é que o protocolo TR069 é agnóstico ao tipo de rede de acesso. Ou seja, ele funciona para redes GPON, HFC ou ADSL e VDSL.

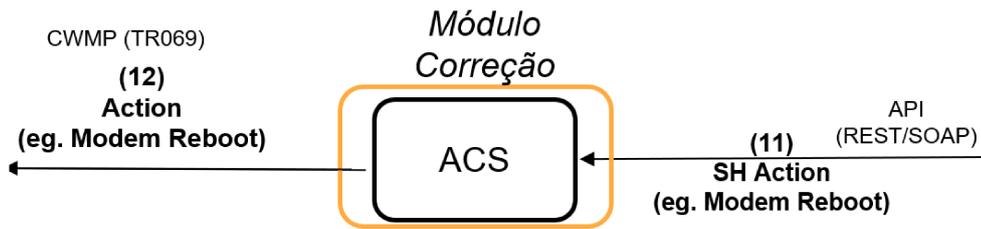


Figura 23 – Módulo de Correção

O módulo de correção, representado no Framework pelo ACS, é implementado pelo fornecedor de escolha da operadora (ver exemplos nesta seção), e fica dentro do ambiente da operadora, em uma estrutura *on premises* ou em *cloud*. Na estrutura em *cloud* é recomendado um *link* de internet entre a operadora e o provedor de *cloud* que garanta alta disponibilidade e o mínimo possível de latência, para uma boa performance da solução.

### 3.8 Visão Comportamental da Arquitetura

Neste ponto mostramos a interação dos módulos e como estes integram para produzir o comportamento esperado do framework. A Figura 24 mostra o diagrama de sequência do framework.

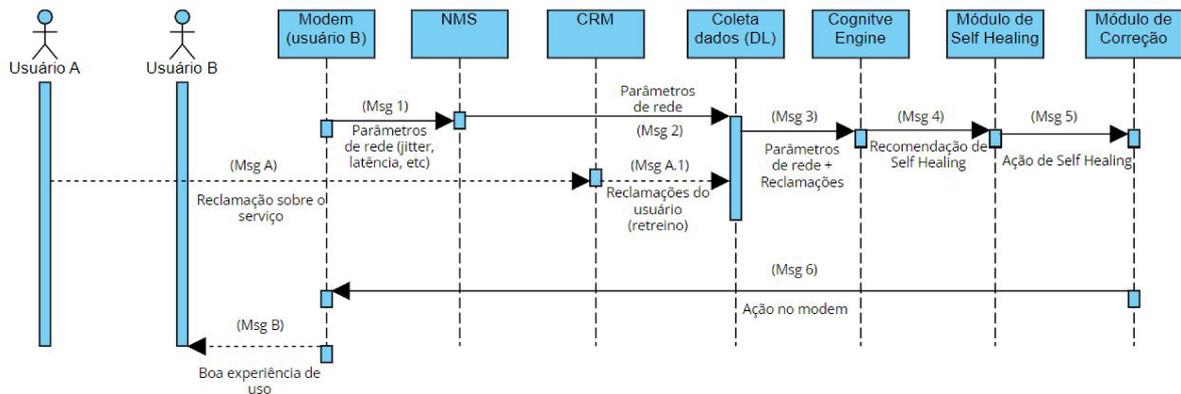


Figura 24 – Diagrama de Sequência

Podemos observar neste diagrama a sequência das mensagens do caso de uso onde o framework já está implantado em uma operadora. Assim, considera-se o algoritmo do módulo Cognitive Engine já pré-treinado com dados anteriores. Portanto, o diagrama mostra o caso de uso onde quando o usuário tiver uma experiência ruim de uso do serviço, isto será automaticamente detectado pelo módulo de rede (MODEM + NMS), e transmitido para os demais módulos de maneira a executar uma ação de SH. Além disso, para facilitar a descrição, abordamos somente a situação onde não há falhas nas respostas de cada componente do framework.

Ainda na Figura 24, inserimos os agentes "Usuário A" (que faz uma reclamação de uso do serviço e contribui para o retreino do modelo) e "Usuário B", que no início do processo tem uma experiência ruim do serviço de banda larga. Para o usuário B, a experiência ruim se reflete nos parâmetros de qualidade (ver seção 2.1.5) do seu MODEM. Neste diagrama de sequência as mensagens do usuário e para o usuário são identificadas de maneira diferente ("tracejadas") pois elas não afetam diretamente este caso de uso de execução automática do framework de SH.

A partir do monitoramento dos parâmetros que indicam uma experiência ruim do usuário B pelo NMS (*Msg 1*), este transmite esses dados em maneira *Near Real Time* para o módulo de coleta de dados (DTL) compondo a *Msg 2*.

É importante destacar que as reclamações do usuário A (*Msg A* e *Msg A.1*) são utilizadas para se realizar um treinamento contínuo do modelo de ML. Isto ocorre pois sempre que o módulo Cognitive Engine for utilizado sobre todos os parâmetros de rede recebidos do NMS (framework já treinado e implantado) ele vai identificar um problema e vai atuar previamente à reclamação do usuário, sendo portanto pró-ativo.

As informações da *Msg 2* são portanto enviadas pelo módulo de coleta de dados para o Cognitive Engine, que analisa esse input (*Msg 3*), e recomenda uma ação de SH (*Msg 4*) para o próximo módulo.

O Módulo de SH, baseado na recomendação recebida, define qual ação e qual plataforma irá acionar para executar a ação recomendada. Uma vez definida, esta ação é encaminhada para o módulo de Correção (plataforma ACS) pela *Msg 5*.

Uma vez chamado, o módulo de Correção que aciona o MODEM do usuário B com o comando necessário para melhorar a experiência do mesmo (*Msg 6*), fechando o loop de SH.

A *Msg B* mostrada no diagrama somente ilustra que, após a execução completa do caso de uso apresentado, o usuário B terá uma experiência de uso do serviço melhor, não sendo portanto uma mensagem que compõe o fluxo do sistema automatizado.

---

## Discussão e Análise dos Resultados

O objetivo deste capítulo é discutir e fazer a avaliação experimental em termos de estrutura e algoritmos de ML para o framework proposto para a autocura cognitiva de redes de banda larga. Nossa experimentação se concentrou no módulo Cognitive Engine, que é a função central deste framework, e ajuda a comprovar a eficácia dos algoritmos de ML em redes de banda larga. Descrevemos como foi o método de avaliação, fazemos uma discussão sobre os algoritmos implementados, detalhamos o *dataset* real utilizado para os experimentos e comparamos os resultados dos algoritmos implementados.

### 4.1 Método para a Avaliação

O primeiro passo foi estudar os problemas e necessidades de uma operadora de telecomunicações para identificar os parâmetros que devem ser monitorados para garantir a melhor experiência possível aos usuários da mesma na utilização do serviço de banda larga. Para tanto, foram realizadas diversas entrevistas com funcionários do Network Operations Center (NOC) e da área de marketing da operadora.

Inicialmente, verificou-se com a área de marketing da operadora que não é conhecida a experiência de uso de serviço banda larga fixa por cliente em sua plenitude. O centro de operações de rede (NOC) atua com foco em reestabelecer incidentes de rede. O ideal para a operadora seria atuar preventiva e proativamente na jornada do seu usuário.

Baseado nessa premissa, foi feita uma investigação em conjunto com os técnicos da operadora para identificar os principais motivos de reclamação e cancelamento dos usuários de banda larga GPON, com o intuito de se avaliar se seria possível uma solução automática que gerenciasse a experiência do usuário. Além disso, conhecendo-se a causa-raiz de uma má experiência, estima-se ser possível criar uma solução automática que atue preventivamente para garantir a melhor experiência possível de uso, além de reduzir custos com ligações no atendimento e envio de técnicos para as instalações do cliente.

Foram 6 meses de investigação, coletando informações da gerência de rede (NMS), do sistema de registro de reclamações de clientes (CRM), além de análise contínua da

correlação entre as informações de rede e reclamações de usuários do serviço. Estas informações foram consolidadas no *dataset* real que é detalhado na seção 4.2.

A partir da análise detalhada dos dados coletados durante o período citado, e em discussões com os especialistas de rede da operadora, notou-se que existia uma padronização nos parâmetros de uso do usuário quando este reclamava de baixa experiência do serviço na operadora.

Então, foi concluído que uma solução que poderia identificar esse padrão e, proativamente sugerisse que o usuário iria reclamar, seria um algoritmo de ML integrado à uma plataforma que conseguisse se comunicar com o MODEM.

Dessa forma, decidimos implementar os algoritmos do módulo Cognitive Engine, desse framework, para identificar se conseguíamos aplicar com sucesso (boa precisão) técnicas de ML em redes de banda larga.

Além de mostrar o resultado do desempenho de cada algoritmo de ML implementado, esta seção também valida o comportamento esperado do framework a partir de um estudo dos dados reais da operadora através de um estudo interpretativo do *dataset* utilizado.

## 4.2 *Dataset*

Nos testes realizados foram coletados dados reais de usuários de duas cidades atendidas pela operadora durante 6 meses (Janeiro a Junho de 2023): um conjunto de dados voltado para o segmento de mercado varejo (banda larga residencial) e outro para o segmento de mercado empresarial (micro e pequenas empresas).

Os dados do MODEM foram coletados para um conjunto específico de usuários no NMS. Ao final do período, os dados desse mesmo conjunto de usuários foram também coletados do CRM. Se o cliente reclamasse durante esse mesmo período, filtrávamos os parâmetros do MODEM no dia da última reclamação. Para os usuários que não reclamaram dentro do período, utilizamos os parâmetros da última coleta do seu MODEM.

Esses conjuntos de dados foram unificados, anonimizados e forneceram informações sobre os parâmetros do MODEM (Jitter, Latência, Perda de Pacotes e Qualidade do Canal Wi-Fi utilizado).

O resultado esperado em relação às reclamações de clientes no período (reclamação de clientes = 1 ou não houve reclamações = 0) foi adicionado a esses dados, completando o conjunto de dados final, que foi agrupado em um arquivo padrão .CSV, disponível em Vieira e Silva (2024).

Desta forma, esperava-se que os modelos previssem, através da análise dos dados do MODEM, se o utilizador iria reclamar, para posteriormente se tomar uma ação de SH.

Na Tabela 4, podemos ver a estrutura do *dataset* utilizado para o experimento.

Pode-se notar que temos 6 parâmetros de rede (colunas 2 a 7) e 1 parâmetro de sistema (coluna 8). O ID do usuário (Customer ID) é anonimizado na coluna 1. Observamos na

Tabela 4 – Parâmetros do dataset utilizado

Customer ID	Latency	Jitter	Packet Loss	Channel2 Quality	Channel5 Quality	N_distant devices	CRM Complaint (0/1)
1	9.26	3.56	0.00%	5	5	0	1
2	6.62	3.51	0.00%	5	5	1	1
3	6.26	2.36	0.00%	0	0	0	1
4	14.13	3.3	0.16%	4	4	2	1
5	13.81	3.19	0.04%	3	5	3	1
6	6.62	2.98	0.00%	4	5	0	1
7	6.53	2.75	0.00%	3	5	0	1
8	14.33	3.64	0.00%	3	5	1	1
9	14.37	3.81	0.03%	4	5	1	1
10	15.56	3.7	0.00%	0	0	0	1
11	6.63	2.66	0.00%	4	5	1	1
12	7.12	2.87	0.00%	5	5	7	1
13	6.2	2.82	0.00%	4	3	1	1
14	24.4	3.63	0.00%	4	5	0	1
15	18.31	3.93	0.00%	5	5	0	1
16	14.78	3.95	0.00%	5	5	0	1
17	21.66	3.09	0.00%	5	5	1	1
18	15.11	3.36	0.05%	3	5	0	1
19	15.57	3.29	0.00%	2	5	3	1
20	20.05	2.87	0.00%	4	5	7	1
21	12.89	3.63	0.29%	3	5	0	1

Fonte: (VIEIRA; SILVA, 2024)

tabela que algumas combinações de parâmetros de rede indicavam que o cliente tinha uma tendência maior a reclamar, o que comprovamos com a coluna 8, onde "1" indica que o cliente reclamou no período da coleta dos dados.

Para normalizar os dados, adaptamos o nosso *dataset* substituindo valores nulos por zero, além de adaptarmos algumas colunas com menos linhas, complementando com zero e concatenando valores.

A seguir apresentamos um resumo estatístico do *dataset*:

Número de reclamações totais: 697 usuários, de um total de 2043 usuários únicos presentes no arquivo.

Parâmetro	Média	Máximo	Mínimo	Desvio Padrão	Variância
Latency (ms)	10.06	24.89	0.00	5.58	31.11
Jitter (ms)	3.37	81.19	0.00	1.81	3.29
Packet Loss (%)	0.37%	75.83%	0.00%	4.34%	0.19
Channel 2 Quality	3.89	5	0	0.96	0.93
Channel 5 Quality	4.52	5	0	1.07	1.15
N-Distant-Devices	1.79	42	0	3.08	9.49

Tabela 5 – Resumo Estatístico dos Parâmetros do Dataset

A partir da análise destes dados, pode-se identificar que:

- ❑ **Latência e Jitter:** A latência média de 10.06 ms e o jitter médio de 3.37 ms indicam uma qualidade de conexão razoável, mas há casos extremos de jitter muito alto (81.19 ms), o que pode afetar a experiência do usuário.
- ❑ **Perda de Pacotes:** A perda de pacotes média é baixa (0.37%), mas há outliers significativos com perdas de até 75.83%, o que pode indicar problemas graves de rede em alguns casos.
- ❑ **Qualidade dos Canais:** A qualidade média dos canais 2 e 5 é alta (3.89 e 4.52, respectivamente), sugerindo que a maioria dos usuários tem uma boa qualidade de sinal.
- ❑ **Dispositivos Distantes:** A média de dispositivos distantes é relativamente baixa (1.79), mas há casos com até 42 dispositivos, o que pode indicar ambientes com muitos dispositivos conectados, potencialmente sobrecarregando a rede.

Nos gráficos a seguir ilustramos as relações entre os parâmetros do *dataset*.

Na Figura 25, identificamos a média dos valores dos parâmetros medidos no MODEM dos usuários de banda larga que reclamaram (CRM Complaint = 1).

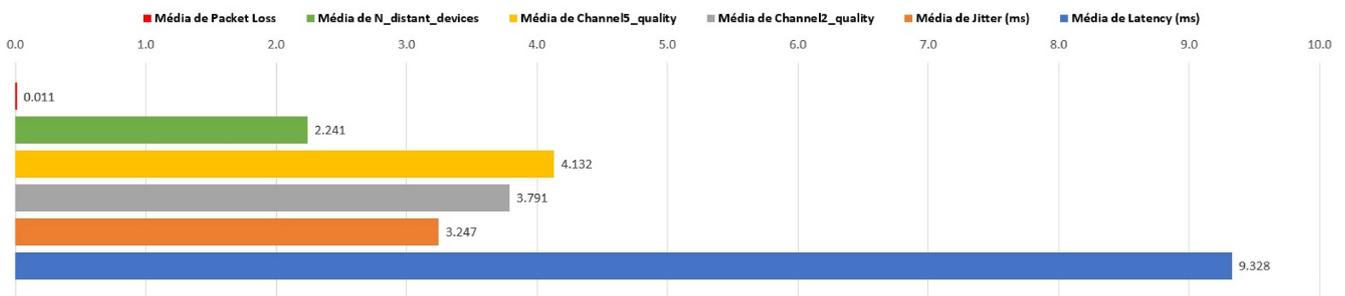


Figura 25 – Média dos parâmetros de rede dos usuários que reclamaram

Fonte: (VIEIRA; SILVA, 2024)

De acordo com entrevistas com equipe de operação e manutenção da operadora, muitos dos dados reais dos MODEMs mostrados neste gráfico ultrapassam os limites que garantem uma boa experiência de uso do usuário do serviço. De fato, na barra verde, observa-se uma média de 2,24 equipamentos conectados à rede Wi-Fi que estão a distâncias maiores que 10 metros do modem, o que sugere uma baixa experiência de uso de banda larga pela baixa potência do sinal capturado devido a distância ao MODEM. Outro parâmetro importante é a qualidade do canal Wi-Fi. É feita uma avaliação pelo próprio MODEM da qualidade do canal, considerando nível de potência do sinal da fibra assim como o nível de interferência nesta frequência. Neste ponto, observa-se uma avaliação média de 3,79 (de 0 a 5) para o canal de 2.4GHz dos clientes que reclamaram, ao

passo que observamos uma avaliação média de 4,13 (de 0 a 5) para o canal de 5GHz dos reclamantes. Faz sentido o canal de 2.4GHz ter uma nota inferior, se observarmos que a maioria dos usuários se conecta automaticamente ao canal de 2.4GHz (maior cobertura). Ainda neste gráfico vemos que a média da latência dos usuários que reclamaram é de 9,32 ms, e o jitter médio é de 3,24 ms dentre os reclamantes. Por fim, para estes usuários, a média da perda de pacotes medida foi de 1,07%.

Na Figura 26, identificamos que existe um alto volume de dispositivos longe do MODEM com valores de perda de pacotes entre 1% e 9%. Isso nos indica que, na média, quando temos muitos devices com distâncias maiores que 10 metros do modem, existe uma tendência de maiores perdas de pacote, e conseqüentemente uma experiência do usuário ruim.

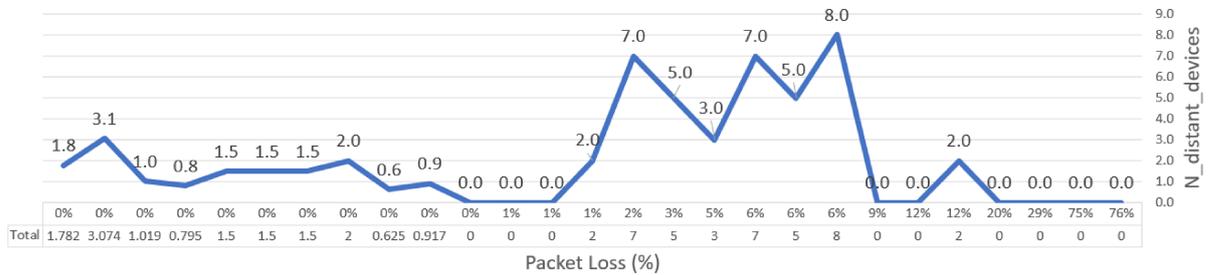


Figura 26 – Relação entre N-distant-devices e Perda de Pacotes

Fonte: (VIEIRA; SILVA, 2024)

Na Figura 27, identificamos que em média temos entre 0 e 5 devices conectados no MODEM da base avaliada. Isso reflete um comportamento geral dos usuários da operadora. Nesta amostra de dados, ainda encontramos 114 ocorrências onde existem entre 5 e 10 devices conectados ao serviço de banda larga naquela residência/empresa, o que reflete menos de 10% do maior valor (1902).

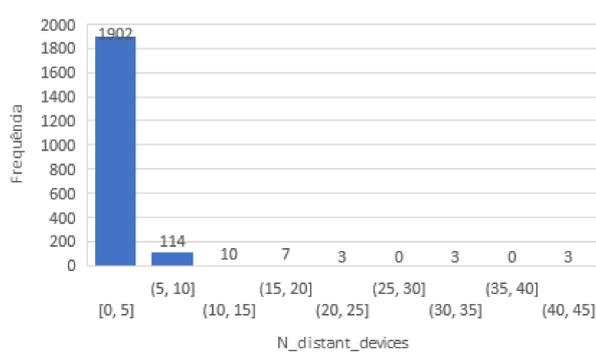


Figura 27 – Frequência de N-Distant-Devices por volume de devices

Fonte: (VIEIRA; SILVA, 2024)

Na Figura 28, identificamos um alto volume de dispositivos longe do modem (acima de 10m) justamente onde tem-se uma baixa nota de qualidade de Wi-Fi para o canal de 5GHz. Neste caso, para a nota "1"(0 a 5), temos uma quantidade média de "3,28"devices longe do MODEM. Lembrando que "Nota 1"significa experiência ruim. Considerando que o canal de 5GHz tem baixo alcance de cobertura, de fato quando temos muitos devices longe do MODEM nesta frequencia, a tendência é termos uma experiência de uso ruim do serviço de banda larga.

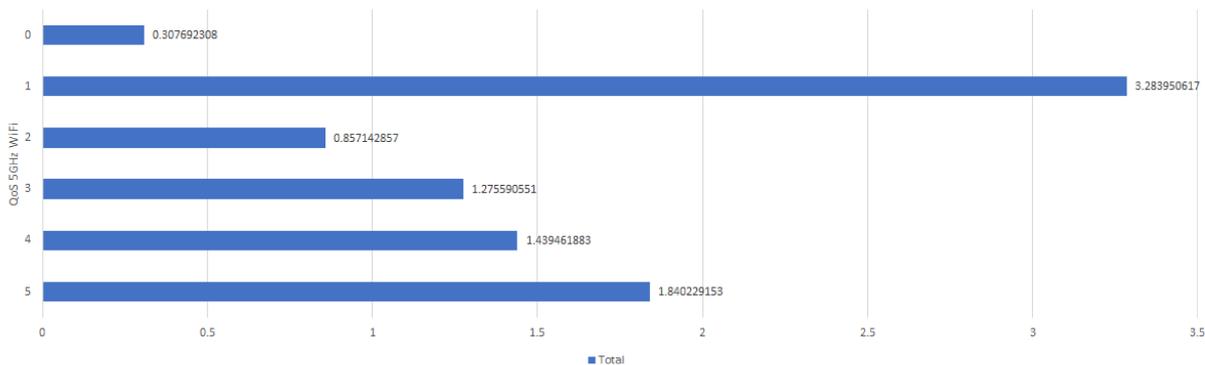


Figura 28 – Comparativo da média de dispositivos com mais de 10m do modem x QoS WiFi 5GHz

Fonte: (VIEIRA; SILVA, 2024)

Na Figura 29, identificamos maiores valores máximos de Jitter do que no cenário de não-reclamação. Por outro lado, tivemos altos valores de latência para ambos os casos. Entendemos que, por esses dados, o jitter alto também tem influência significativa na experiência do cliente.

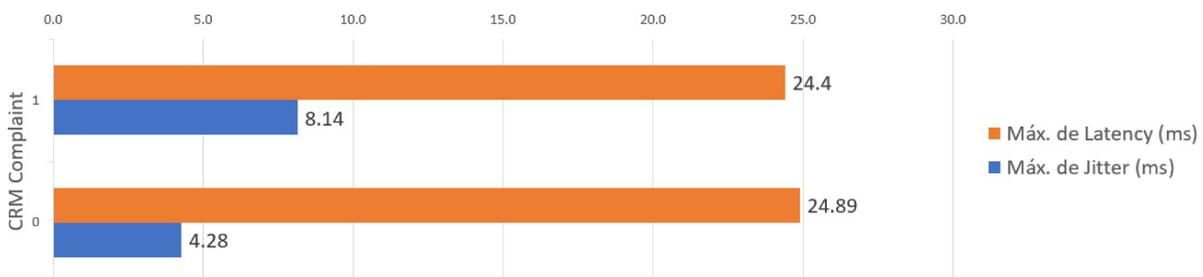


Figura 29 – Jitter e Latência máximos x Reclamações

Fonte: (VIEIRA; SILVA, 2024)

## 4.3 Discussões gerais sobre os algoritmos

O objetivo da implantação de algoritmos de ML em nosso experimento foi o de verificar se os mesmos conseguiam prever que o cliente iria reclamar (experiência de uso ruim) baseado nos seus parâmetros de rede. Para tal, usou-se como input para o modelo os parâmetros do MODEM coletados e os dados de reclamação de usuários (CRM), em um modelo de aprendizado supervisionado. O objetivo é que os algoritmos se aproximem dos dados de reclamação de usuários do CRM. Dessa forma, conseguimos criar um modelo autônomo de SH, que uma vez treinado, consegue identificar se um cliente está com uma baixa experiência de uso do serviço e pode tomar uma ação automática de correção antes que o mesmo reclame na operadora.

Foram implementados três tipos de modelos (em Python), avaliando a sua precisão, a saber: DT, ANN (MLP) e GB. Esses modelos foram selecionados por representarem modelos de aprendizagem supervisionada, pois já temos o resultado esperado (reclamações de usuários ou não). Escolhemos um algoritmo de maior complexidade (ANN), um de baixa complexidade (DT) e um de complexidade intermediária (GB), para verificar se todos eles poderiam ser aplicados ao nosso framework com boa precisão. Para esses modelos, os seguintes parâmetros do modem do usuário foram utilizados como entrada para o treinamento de cada modelo: Latência, Jitter, Perda de Pacotes, Qualidade do Canal de 2,4 GHz, Qualidade do Canal de 5 GHz e Número de Dispositivos Distantes do MODEM. Esses parâmetros foram descritos na Seção 2.1.5.

Os parâmetros do usuário definidos na Seção 2.1.5, foram cruzados com o banco de dados CRM a partir da chave primária "ID do usuário", com o objetivo de encontramos possíveis relacionamentos entre os parâmetros de rede e reclamações. Dependendo da configuração dos valores dos parâmetros de rede, o usuário reclama com a central de atendimento.

### 4.3.1 DT

Sabemos que uma DT usa aprendizado supervisionado e pode ser usada para problemas de classificação e regressão. O objetivo deste experimento é criar um modelo que preveja o valor de uma variável alvo, aprendendo regras de decisão simples inferidas a partir das características dos dados. Uma árvore pode ser vista como uma aproximação constante por partes. Quanto mais profunda a árvore, mais complexas são as regras de decisão e mais refinado é o modelo.

O algoritmo funciona dividindo os dados em subconjuntos menores e mais homogêneos, com base em um conjunto de regras de decisão derivadas dos atributos dos dados. A DT é uma estrutura hierárquica composta por nós, onde o nó Raiz (inicial) representa todo o conjunto de dados. Os nós Internos (intermediários) representam a divisão dos dados com base em um atributo específico. E os nós Folha (finais) representam a decisão ou a

previsão.

O nosso modelo de árvore de decisão analisa os parâmetros do DTL e identifica um problema potencial.

A figura 30 ilustra o algoritmo DT implementado dentro de nosso framework. O algoritmo foi criado com base na classe *DecisionTreeClassifier* da biblioteca *sklearn* (PEDREGOSA et al., 2011), usando um parâmetro de estado aleatório para iniciar este método. Nesse caso, usamos 70% dos dados para treinamento e 30% para teste.

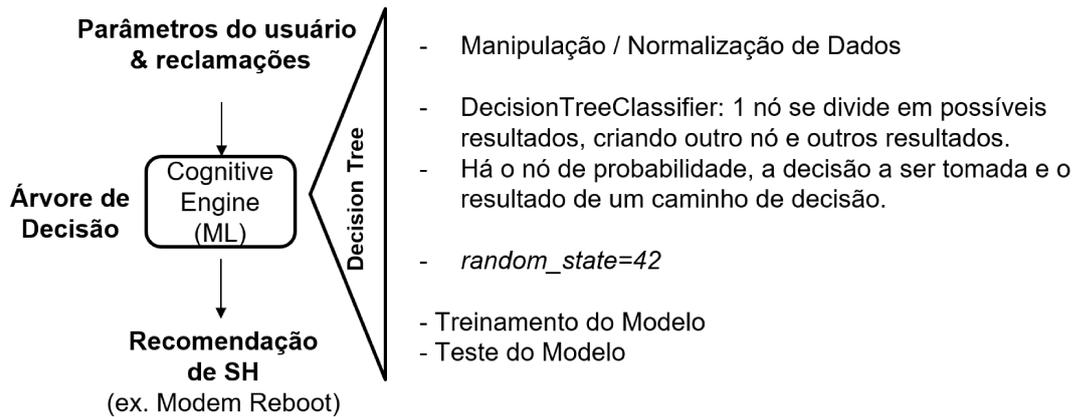


Figura 30 – Árvore de Decisão Implementada - DT

O parâmetro `random-state=42` é frequentemente usado em modelos de ML para garantir a reprodutibilidade dos resultados. Quando se define um valor específico para `random-state`, como 42, estamos fixando a semente do gerador de números aleatórios. Isso significa que, sempre que se executar o código com esse valor, os resultados serão os mesmos, permitindo que se possa replicar os experimentos e validar os resultados de forma consistente.

Como saída do método DT, um *Data Frame* é gerado como um arquivo Excel. Este *Data Frame* contém os parâmetros do usuário, a saída esperada (o usuário reclama ou não) e a saída real. Depois disso, o resultado esperado e o real podem ser comparados e a precisão do método pode ser medida.

### 4.3.2 RNA (MLP)

Também implementamos um MLP, que é uma ANN do tipo Perceptron (classificador), mas com mais de uma camada de neurônios em alimentação direta. Este tipo de rede compreende camadas de neurônios conectados por sinapses ponderadas.

Em nosso trabalho, um algoritmo ANN foi criado utilizando as classes *Sequential* e *Dense* da API Keras (CHOLLET et al., 2015). Após testes com diferentes topologias (números de camadas, neurônios e funções de ativação), chegamos em uma arquitetura mais estável com os seguintes parâmetros:

- ❑ 1 camada de entrada (100 neurônios, ativação função = relu)
- ❑ 2 camadas densas (100 neurônios, função de ativação = relu)
- ❑ 1 camada de saída (1 neurônio, função de ativação = sigmóide)
- ❑ Função de perda='entropia cruzada binária'; Otimizador='adam'; Métricas='precisão'

A figura 31 ilustra a arquitetura da nossa ANN implementada.

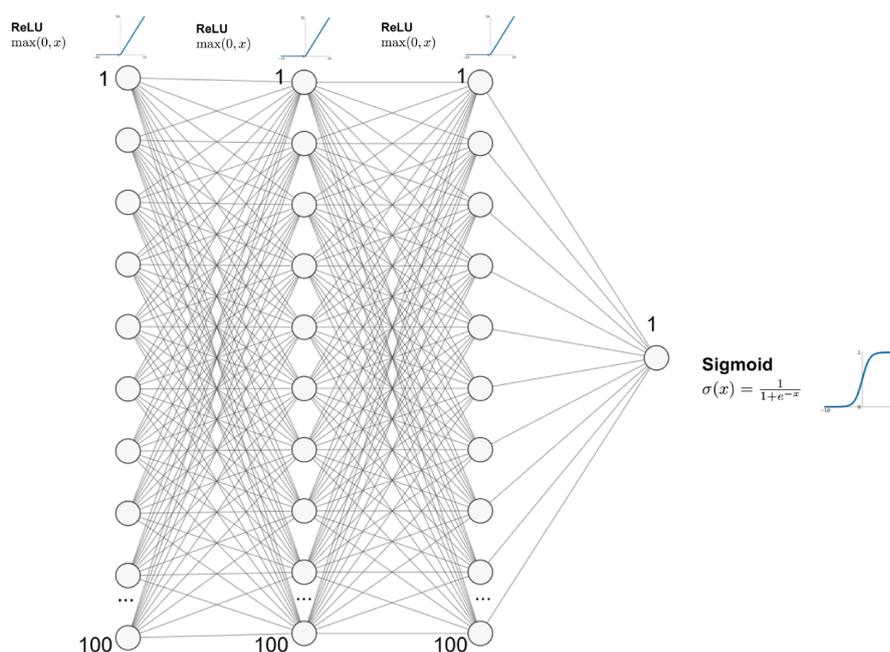


Figura 31 – Topologia da RNA Implementada

A figura 32 explica os parâmetros do algoritmo de ANN implementado dentro do framework. Neste caso usamos 70% dos dados para treinamento e 30% para teste.

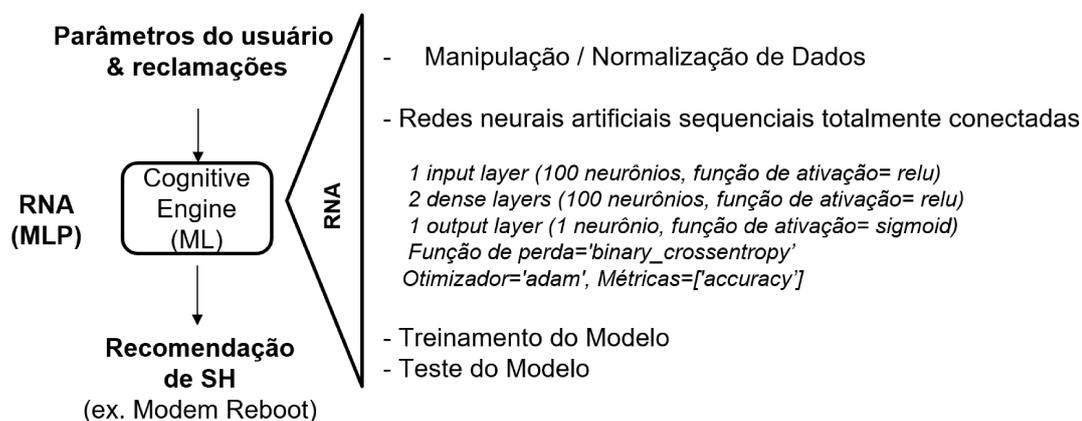


Figura 32 – Rede Neural Artificial Implementada

Um *Data Frame* é gerado num arquivo Excel como saída do método ANN. Este quadro de dados contém os parâmetros do usuário, a saída esperada (o usuário reclama ou não) e a saída real. Depois disso, os resultados esperados e reais são comparados e a precisão do método é medida.

### 4.3.3 GB

Já na implementação do algoritmos GB, que é uma técnica de ML para problemas de regressão e classificação, produzimos um modelo de predição na forma de um conjunto de modelos de predição fracos, geralmente árvores de decisão (DT). Construimos o modelo em etapas e as generalizamos, permitindo a otimização de uma função de perda diferenciável arbitrária. Em suma, os erros anteriores de análise de dados de aprendizagem são gradualmente reduzidos.

Em nosso trabalho, um algoritmo de GB foi criado usando a classe *GradientBoostingClassifier* da biblioteca *sklearn* (PEDREGOSA et al., 2011), usando um parâmetro de estado aleatório para iniciar este método. A figura 33 ilustra o algoritmo GB implementado dentro do framework. Da mesma forma como nos algoritmos anteriores, utilizamos 70% dos dados para treinamento e 30% para teste.

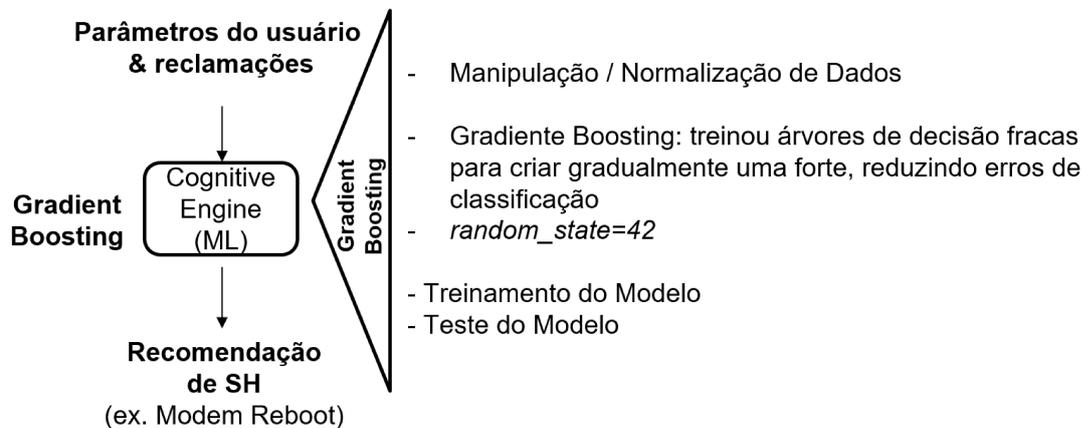


Figura 33 – Gradient Boosting Implementado

Da mesma forma como no algoritmo DT, o parâmetro `random-state=42` é usado para garantir a reprodutibilidade dos resultados, pois estamos fixando a semente do gerador de números aleatórios. Isso significa que, sempre que se executar o código com esse valor, os resultados serão os mesmos, permitindo que se possa replicar os experimentos e validar os resultados de forma consistente.

Como anteriormente, em nosso experimento, o método GB gera um *Data Frame* como saída no padrão de um arquivo Excel, para simplificação. Este *Data Frame* contém os parâmetros do usuário, a saída esperada (o usuário reclama ou não) e a saída real. Por

fim, os resultados esperados e os reais são comparados, e a precisão do método utilizado é medida.

## 4.4 Avaliação dos Resultados

### 4.4.1 Resultados dos Algoritmos implementados

Os resultados dos algoritmos são apresentados na Tabela 6, onde “Precisão” é o parâmetro mais importante para comparação dos modelos, visto que ele nos guia se um modelo de ML pode ou não ser aplicado em uma solução de SH em redes de banda larga, pois podemos entender que se ele acertou que o usuário vai reclamar, logo a ação de SH sugerida terá efeito.

No início da implementação, identificamos problemas de *overfitting*, ou ajuste excessivo, que é um comportamento indesejável de aprendizado de máquina que ocorre quando o modelo de aprendizado de máquina fornece previsões precisas para dados de treinamento, mas não para novos dados. Por exemplo, nos primeiros testes do modelo de redes neurais (MLP), chegamos a uma precisão de 100%. Logo, fizemos mais testes para o MLP, reduzindo o número de épocas de treinamento, o que fez com que o modelo tivesse uma precisão mais realista. Além disso, a normalização do *dataset* também ajudou nesse propósito, pois reduziu *outliers*, ou valores atípicos, em nosso conjunto de dados.

Tabela 6 – Resultados dos Algoritmos

Algoritmo	Nº Camadas	Nº Épocas	Nº Neurônios	Nº Linhas de código	TT (s)	Precisão (%)
DT	NA	NA	NA	162	120	97,87
ANN	4 camadas	1000	100	183	600	97,55
GB	NA	NA	NA	161	300	96,90

TT - Tempo para Treinamento

Conforme mostrado na tabela, todos os algoritmos levaram a uma precisão superior a 96%, o que é totalmente aceitável neste contexto. Esta precisão valida a nossa suposição de que um modelo cognitivo poderia ser importante no processo de atendimento ao cliente de uma operadora.

Na Figura 34 mostramos a matriz de confusão gerada para os algoritmos implementados.

As matrizes de confusão apresentadas para cada algoritmo ilustram a precisão apresentada na Tabela 6. No caso do algoritmo DT, obtivemos uma precisão segundo a matriz de 97,87%, enquanto no modelo ANN a precisão foi de 97,55%. Por fim, obtivemos a precisão de 96,90% com o algoritmo GB.

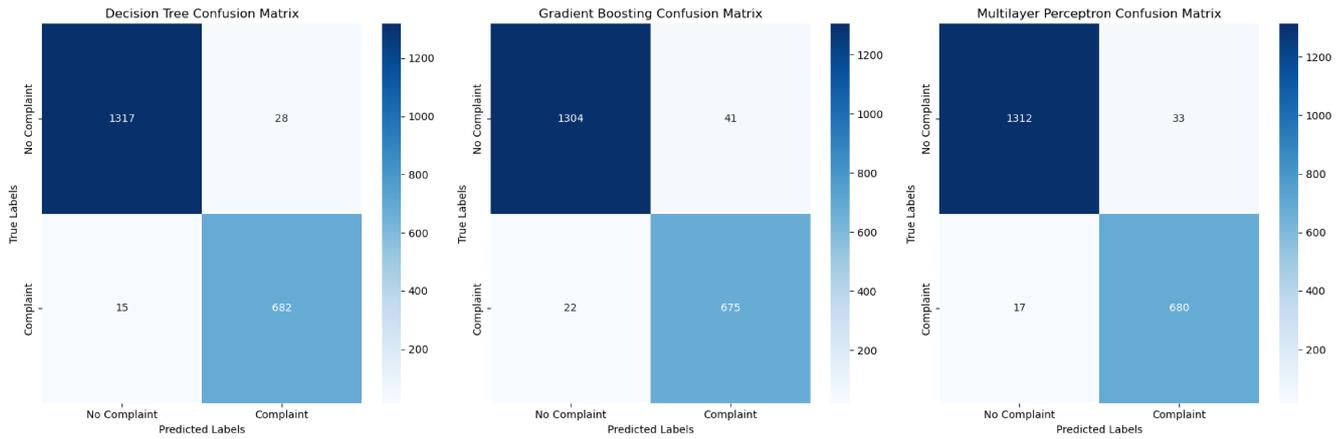


Figura 34 – Matriz de confusão dos algoritmos implementados

Interpretação das Matrizes de Confusão:

- ❑ True Positives (TP): Clientes que reclamaram e foram corretamente classificados como reclamantes. Na figura, é representado como azul escuro.
- ❑ True Negatives (TN): Clientes que não reclamaram e foram corretamente classificados como não reclamantes. Na figura, é representado como azul claro.
- ❑ False Positives (FP): Clientes que não reclamaram, mas foram incorretamente classificados como reclamantes. Na figura, é representado no canto superior direito.
- ❑ False Negatives (FN): Clientes que reclamaram, mas foram incorretamente classificados como não reclamantes. Na figura, é representado no canto inferior esquerdo.

No caso do algoritmo DT, tivemos 1317 *True Positives* e 682 *True Negatives*, o que representa que o modelo desenvolvido teve boa performance, considerando que esses valores representam 97,8% do *dataset*.

No caso do algoritmo GB, tivemos 1304 *True Positives* e 675 *True Negatives*, o que representa que o modelo desenvolvido teve boa performance, considerando que esses valores representam 96,8% do *dataset*.

Por fim, para o algoritmo ANN, tivemos 1312 *True Positives* e 680 *True Negatives*, o que representa que o modelo desenvolvido teve boa performance, considerando que esses valores representam 97,5% do *dataset*.

Observando que foram poucos os itens da matriz de confusão classificados como Falso Positivos e Falso Negativos, podemos concluir que a divisão do *dataset* em 70% para treinamento e 30% para testes foi adequada.

Os experimentos também mostraram que a ANN é uma ferramenta flexível e de alto potencial para resolver problemas de classificação. Neste modelo, o tempo de treinamento (neste experimento, o mais lento) é um ponto a ser observado e dependerá do problema a

ser resolvido. Como usamos 2.043 linhas em nosso *dataset*, isso não foi uma preocupação. No entanto, pode ser um desafio quando o problema necessitar de maiores volumes de dados para treinar o modelo.

Como acontece com qualquer modelo de ML, algumas precauções pós-treinamento são importantes para manter um bom desempenho quando o algoritmo está em produção, uma vez que as fontes de dados nos operadores podem mudar (novos dados, ambiente dinâmico, etc). A reavaliação do modelo é muito importante para garantir o desempenho do processo. O retreinamento periódico do modelo com novos dados também é importante para garantir que a precisão continuará em valores aceitáveis.

A estrutura proposta pode efetivamente mudar a forma como os operadores trabalham, permitindo-lhes tomar ações preventivas com base em ML. Desta forma, o sistema pode decidir automaticamente com base nos parâmetros do usuário, sem intervenção humana. Uma vez implementada esta estrutura nas operadoras de telecomunicações, a satisfação do usuário pode ser melhorada porque as ações SH podem ser realizadas em tempo hábil (por exemplo, reinicialização do modem), evitando reclamações dos usuários.

É importante notar que este framework pode ser implementado em operadoras de telecomunicações com soluções proprietárias, abertas ou em cenários híbridos. Esta flexibilidade reforça a proposta inicial deste Framework.

## 4.5 Visão deste trabalho e comparação com o estado da arte

A Tabela 7 compara os trabalhos relacionados discutidos anteriormente, identificando cada abordagem, suas características e comparando com o posicionamento do nosso trabalho.

Podemos notar que o Framework proposto por este trabalho se diferencia dos demais trabalhos do estado da arte, pois inclui uma abordagem de SH com o uso de ML em redes de banda larga GPON. Além disso, Nosso trabalho usa dados reais de uma operadora e trabalha com o conceito de *realtime*. Nossa aplicação atua no domínio de detecção automática de falhas e SH automático (autocura), usando técnicas de aprendizado supervisionado a partir dos dados informados pela operadora. Por fim, os parâmetros de rede utilizados em nosso Framework diferenciam este trabalho pois atuam diretamente na resolução do problema da baixa experiência do cliente.

Tabela 7 – Trabalhos Relacionados x Framework proposto

Work	SH	ML	SH em GPON	Dados Reais	RT	Rede	Tipo de SH	Técnica de ML	Parâmetros
(KICIMAN; FOX, 2005)	Não	Sim	Não	Não	Não	Data Center	Localiz. de Falhas	Superv.: DT	HTTP Error
(WANG et al., 2007)	Não	Sim	Não	Não	Não	Sem Fio	Predição de Falhas	Superv.: DT,SVM	Signal strength, Channel load
(XU et al., 2016)	Sim	Sim	Não	Não	Sim	Celular	Gestão de Recursos	Sup.: ANN	QoS, Energy cons., VNF Load
(HASHMI et al., 2017)	Sim	Sim	Sim	Sim	Não	Banda Larga	Detecção de Falhas	Não-Sup.: k-Means, FCM · SOM	Fault time and cause, Region, MTTR
(RAHMANI et al., 2020)	Sim	Sim	Não	Não	Não	Celular	Diagnóst. de Falhas	Não-Sup.: SVM, Fuzzy, Adaboost	Cell Power, Avg speed, Signal Quality
(SILVA; MORAIS; SILVA, 2020)	Não	Sim	Não	Sim	Não	Banda Larga	Pred. tráfego, Detecção Anomalias	Não-Sup.: SOM Sup: RNN	OLT RX Power, ONT RX Power, ONU dist.
(BOTHE et al., 2020)	Sim	Sim	Não	Sim	Sim	Celular	Diagnóst. de Falhas	Superv.: RF, CNN, ANN.	Cell Outage, TxPower, Antenna Tilt
(OMAR et al., 2021)	Sim	Sim	Não	Não	Não	Celular	Detecção de Falhas e SH Automático	Superv.: DT	Max. data rate, #users per BS and SNR
<i>Este trabalho</i>	Sim	Sim	Sim	Sim	Sim	Banda Larga	Detecção de Falhas e SH Automático	Superv.: ANN, DT, GradBoost	Latency, Jitter, WiFi Quality, Packet Loss

SH - Self-healing, ML - Machine Learning, GPON - Gigabit Passive Optical Network, RT - Real Time

Fonte: O Autor

---

## Conclusão

Este trabalho apresentou um novo framework para autocura cognitiva em redes reais de banda larga para reduzir reclamações de usuários, antecipando medidas corretivas usando ML. O estudo se baseia em parâmetros reais de usuários de banda larga coletados de uma operadora de rede no Brasil, discutidos com suas equipes de engenharia, marketing e operação, de forma que pudemos aprender como extrair valor dos dados de CRM e MODEM. Assim que os dados estavam disponíveis em um DTL, um Cognitive Engine (ML) foi implementado para prever, com precisão aceitável, se um usuário reclamaria do seu serviço. Os resultados mostraram que os três métodos de ML testados produziram uma precisão acima de 96%. Com isso comprovado, pode-se aplicar scripts de SH, utilizando a plataforma ACS para interagir com o dispositivo do usuário, fazendo uma intervenção pró-ativa para se evitar uma provável reclamação.

### 5.1 Principais Contribuições

Ao fim deste trabalho pudemos concluir que é viável a criação de um framework genérico e padronizado para autocura de redes de banda larga, que pode ser implementado pelas operadoras de telecomunicações. Outra contribuição identificada foi a criação de um módulo de ML que pode ser reutilizado em outras arquiteturas de rede. Pudemos ainda implementar e comparar três algoritmos de ML no contexto de SH para redes banda larga. Uma análise detalhada do comportamento de parâmetros do usuário de redes de banda larga foi apresentada identificando as causas de problemas de performance e má experiência de uso do serviço pelos usuários. Por fim, foi disponibilizado um *dataset* anonimizado com dados reais de usuários de uma operadora de telecomunicações, conforme referência citada durante o texto.

## 5.2 Trabalhos Futuros

Para trabalhos futuros, novos algoritmos de ML, como CNN e Generative Artificial Intelligence (GenAI), poderão ser considerados uma alternativa aos modelos implementados neste trabalho, com foco no aumento da precisão alcançada e na facilidade de implementação. Outra sugestão é implementar o módulo de SH para avaliar o desempenho do *loop* completo (tempo total desde o diagnóstico até a ação corretiva). Ainda pode ser explorada uma abordagem com o protocolo BroadBand Forum - User Service Platform (TR369), evolução do TR069 para IoT. Um ponto adicional a explorar é a aplicação deste framework a outros tipos de redes, como switches de redes metropolitanas, roteadores de backbone (*edge e core*), agregadores de acesso (OLTs) e centros de comutação de dados e voz 5G. Por fim, um trabalho futuro sugerido é a implantação do framework completo em uma operadora, testando-o fim a fim, e validando o conceito proposto. Para tal, pode ser usado o *dataset* avaliado neste trabalho um um específico da operadora onde será implementado.

## 5.3 Contribuições em Produção Bibliográfica

Artigo aceito e apresentado no congresso: *38th International Conference of Advanced Information Networking and Applications (AINA) 2024*, com o título: *Framework for Cognitive Self-Healing of Real Broadband Networks* (VIEIRA; CARVALHO; SILVA, 2024).

---

## Referências

AVSYSTEM. **Crash course in TR-069 (CWMP)**. 2024. Disponível em: <<https://www.avsystem.com/crashcourse/tr069/>>.

AXIROS. **Axiros AXESS ACS (Servidor de Configuração Automática)**. 2024. Disponível em: <[https://pt.axiros.com/products/axess-ac3?gad\\_source=1&gclid=CjwKCAjwy8i0BhAkEiwAdFaeGLizBj0l4vx\\_uTMGhRUZ74LOcpXxBZiTVJiGewrXe9l-yvEs0\\_-TgRoCcnUQAvD\\_BwE](https://pt.axiros.com/products/axess-ac3?gad_source=1&gclid=CjwKCAjwy8i0BhAkEiwAdFaeGLizBj0l4vx_uTMGhRUZ74LOcpXxBZiTVJiGewrXe9l-yvEs0_-TgRoCcnUQAvD_BwE)>.

BAZAZEH, D.; SHUBAIR, R. Comparative study of machine learning algorithms for breast cancer detection and diagnosis. In: **2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)**. [s.n.], 2016. p. 1–4. Disponível em: <<https://doi.org/10.1109/ICEDSA.2016.7818560>>.

BOTHE, S. et al. Neuromorphic AI empowered root cause analysis of faults in emerging networks. **CoRR**, abs/2005.01472, 2020. Disponível em: <<https://doi.org/10.1109/BlackSeaCom48709.2020.9235002>>.

BRISCOE, B. et al. Reducing internet latency: A survey of techniques and their merits. **IEEE Communications Surveys Tutorials**, v. 18, n. 3, p. 2149–2196, 2016. Disponível em: <<https://doi.org/10.1109/COMST.2014.2375213>>.

BROADBANDFORUM. **Technical Report - TR-069 CPE WAN Management Protocol**. 2020. Disponível em: <<https://www.broadband-forum.org/pdfs/tr-069-1-6-1.pdf>>.

CHENG, C. C.; SHIU, E. C. How to enhance smes customer involvement using social media: The role of social crm. **International Small Business Journal**, SAGE Publications Sage UK: London, England, v. 37, n. 1, p. 22–42, 2019. Disponível em: <<https://doi.org/10.1177/0266242618774831>>.

CHOLLET, F. et al. **Keras**. GitHub, 2015. Disponível em: <<https://github.com/fchollet/keras>>.

CLEMM, A. **Network Management Fundamentals**. Cisco Press, 2006. ISBN 1587201372. Disponível em: <<https://www.ciscopress.com/store/network-management-fundamentals-9781587052804>>.

G.984 - ITU - International Telecommunication Union. **G.984.1: Gigabit-capable passive optical networks (GPON): General characteristics**. [S.l.], 2008.

Disponível em: <<https://www.itu.int/rec/T-REC-G.984.1>>.

GALBREATH, J.; ROGERS, T. Customer relationship leadership: a leadership and motivation model for the twenty-first century business. **The TQM magazine**, MCB UP Ltd, v. 11, n. 3, p. 161–171, 1999. Disponível em: <<https://doi.org/10.1108/09544789910262734>>.

GENIACS. **Fast, lightweight TR-069 ACS**. 2024. Disponível em: <<https://genieacs.com/>>.

HARRIS, C. R. et al. Array programming with numpy. **Nature**, v. 585, n. 7825, p. 357–362, 2020. ISSN 1476-4687. Disponível em: <<https://doi.org/10.1038/s41586-020-2649-2>>.

HASHMI et al. Enabling proactive self-healing by data mining network failure logs. In: **2017 International Conference on Computing, Networking and Communications (ICNC)**. [s.n.], 2017. p. 511–517. Disponível em: <<https://doi.org/10.1108/09544789910262734>>.

INCOGNITO. **TR-069 Auto Configuration Server para gestão de dispositivos**. 2024. Disponível em: <<https://www.incognito.com/pt/products/tr-069-auto-configuration-server/>>.

KARAM, L. J.; MOUNSEF, N. Modem. In: \_\_\_\_\_. **Introduction to Engineering: A Starter's Guide With Hands-On Digital Multimedia and Robotics Explorations**. Cham: Springer International Publishing, 2008. p. 133–137. ISBN 978-3-031-79315-8. Disponível em: <[https://doi.org/10.1007/978-3-031-79315-8\\_10](https://doi.org/10.1007/978-3-031-79315-8_10)>.

KAZOVSKY, L. G. et al. **Broadband Optical Access Networks**. 1. ed. John Wiley Sons, Inc., 2011. Disponível em: <<https://doi.org/10.1002/9780470910931>>.

KICIMAN, E.; FOX, A. Detecting application-level failures in component-based internet services. **IEEE Transactions on Neural Networks**, v. 16, n. 5, p. 1027–1041, 2005. Disponível em: <<https://doi.org/10.1109/TNN.2005.853411>>.

KLAINÉ, P. V. et al. A survey of machine learning techniques applied to self-organizing cellular networks. **IEEE Communications Surveys Tutorials**, v. 19, n. 4, p. 2392–2431, 2017. Disponível em: <<https://doi.org/10.1109/COMST.2017.2727878>>.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015. Disponível em: <<https://doi.org/10.1038/nature14539>>.

MONTANI, S.; ANGLANO, C. Achieving self-healing in service delivery software systems by means of case-based reasoning. **Appl. Intell.**, v. 28, p. 139–152, 04 2008. Disponível em: <<https://doi.org/10.1007/s10489-007-0047-1>>.

NAMBIAR, A.; MUNDRA, D. An overview of data warehouse and data lake in modern enterprise data management. **Big Data and Cognitive Computing**, v. 6, n. 4, 2022. ISSN 2504-2289. Disponível em: <<https://doi.org/10.3390/bdcc6040132>>.

NETO, N. V. d. S. **Autocura para redes definidas por software**. Tese (Doutorado) — Universidade Federal de Uberlândia, 2021. Disponível em: <<http://doi.org/10.14393/ufu.te.2021.246>>.

NTIA. **Broadband Network Deployment Engineering**. [S.l.], 2022. Disponível em: <<https://broadbandusa.ntia.doc.gov/sites/default/files/2022-03/Broadband%20Network%20Deployment%20Engineering%20PDF.pdf>>.

OMAR et al. A novel self-healing model using precoding & big-data based approach for 5g networks. **Pervasive and Mobile Computing**, Elsevier, v. 73, p. 101365, 2021. Disponível em: <<https://doi.org/10.1016/j.pmcj.2021.101365>>.

OSTERBO, O.; GRONDALEN, O. Benefits of self-organizing networks (son) for mobile operators. **Journal of Computer Networks and Communications**, Hindawi, v. 2012, 2012. Disponível em: <<https://doi.org/10.1155/2012/862527>>.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Disponível em: <<https://dl.acm.org/doi/10.5555/1953048.2078195>>.

PEREZ-VEGA, R. et al. From crm to social crm: A bibliometric review and research agenda for consumer research. **Journal of Business Research**, v. 151, p. 1–16, 2022. ISSN 0148-2963. Disponível em: <<https://doi.org/10.1016/j.jbusres.2022.06.028>>.

RAHMANI et al. Mpra - self-healing in lte networks with unsupervised learning techniques. 2020. Disponível em: <[https://www.researchgate.net/publication/340817803\\_Self-Healing\\_in\\_LTE\\_Networks\\_with\\_Unsupervised\\_Learning\\_Techniques](https://www.researchgate.net/publication/340817803_Self-Healing_in_LTE_Networks_with_Unsupervised_Learning_Techniques)>.

RAMAN, L. **Fundamentals of telecommunications network management**. Wiley-IEEE Press, 1999. Disponível em: <<https://doi.org/10.1109/MCOM.2000.819893>>.

REDHAT. **RedHat: Whitepaper: Accelerate your path to self-healing IT infrastructure (redhat.com)**. 2022. Disponível em: <<https://www.redhat.com/en/resources/accelerate-self-healing-whitepaper>>.

SARKAR, D.; BALI, R.; SHARMA, T. The python machine learning ecosystem. In: \_\_\_\_\_. **Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems**. Berkeley, CA: Apress, 2018. p. 67–118. ISBN 978-1-4842-3207-1. Disponível em: <[https://doi.org/10.1007/978-1-4842-3207-1\\_2](https://doi.org/10.1007/978-1-4842-3207-1_2)>.

SATHYAN, J. **Fundamentals of Ems, Nms and Oss/bss**. CRC Press, 2016. Disponível em: <<https://doi.org/10.1201/b15748>>.

SHINDE, P. P.; SHAH, S. A review of machine learning and deep learning applications. In: **2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)**. [s.n.], 2018. p. 1–6. Disponível em: <<https://doi.org/10.1109/ICCUBEA.2018.8697857>>.

SILVA, W. S.; MORAIS, A. Silva de; SILVA, W. d. O. Proposal for the use of neural networks for data clustering in the context of qualitative analysis of complaints information in telecommunications services. **New Trends in**

- Qualitative Research**, v. 4, p. 499–506, Jul. 2020. Disponível em: <<https://doi.org/10.36367/ntqr.4.2020.499-506>>.
- TANG, L. Z. . X. D. X. Sar image despeckling with a multilayer perceptron neural network. **International Journal of Digital Earth**, p. 354–374, 2019. Disponível em: <<https://doi.org/10.1080/17538947.2018.1447032>>.
- UZOKA, A.; CADET, E.; OJUKWU, P. U. The role of telecommunications in enabling internet of things (iot) connectivity and applications. **Comprehensive Research and Reviews in Science and Technology**, CRR Journals, v. 2, n. 2, p. 55–73, 2024. Disponível em: <<https://doi.org/10.57219/crrst.2024.2.2.0037>>.
- VALIM, E. F.; ISHII, R. P. Revisão de técnicas de aprendizado de máquina no contexto de correspondência de produtos. 2023. Disponível em: <<https://repositorio.ufms.br/handle/123456789/7476>>.
- VIEIRA, E.; SILVA, F. D. O. **Real Dataset From Broadband Customers of a Brazilian Telecom Operator**. 2024. Disponível em: <<https://zenodo.org/records/10482897>>.
- VIEIRA, E. C. A.; CARVALHO, P.; SILVA, F. de O. Framework for cognitive self-healing of real broadband networks. In: BAROLLI, L. (Ed.). **Advanced Information Networking and Applications**. Cham: Springer Nature Switzerland, 2024. p. 163–174. ISBN 978-3-031-57870-0. Disponível em: <[https://link.springer.com/chapter/10.1007/978-3-031-57870-0\\_15](https://link.springer.com/chapter/10.1007/978-3-031-57870-0_15)>.
- WANG et al. Predicting link quality using supervised learning in wireless sensor networks. **Mobile Computing and Communications Review**, v. 11, p. 71–83, 07 2007. Disponível em: <<https://doi.org/10.1145/1317425.1317434>>.
- XIE, F.; WEI, D.; WANG, Z. Traffic analysis for 5g network slice based on machine learning. **EURASIP Journal on Wireless Communications and Networking**, v. 2021, n. 1, p. 108, 2021. Disponível em: <<https://doi.org/10.1186/s13638-021-01991-7>>.
- XU, L. et al. Cognet: A network management architecture featuring cognitive capabilities. In: **2016 European Conference on Networks and Communications (EuCNC)**. [s.n.], 2016. p. 325–329. Disponível em: <<https://doi.org/10.1109/EuCNC.2016.7561056>>.
- ZHANG, T.; ZHU, K.; HOSSAIN, E. Data-driven machine learning techniques for self-healing in cellular wireless networks: Challenges and solutions. **Intelligent Computing**, v. 2022, 2022. Disponível em: <<https://spj.science.org/doi/abs/10.34133/2022/9758169>>.