PGCOMP – Programa de Pós-Graduação em Ciência da Computação
Universidade Federal da Bahia (UFBA)
Av. Milton Santos, s/n – Ondina
Salvador, BA, Brasil, 40170-110

https://pgcomp.ufba.br
pgcomp@ufba.br

This work presents the Névoa das Coisas Confiável (do Inglês *Realible Fog of Things*) (RFoT), a framework that integrates an Internet das Coisas (do inglês *Internet of Things*) (IoT) network with Blockchain and Contratos Inteligentes (do Inglês *Smart Contracts*) (CI) technologies, to offer IoT as a reliable data source for training Aprendizado de Máquina (do Inglês *Machine Learning*) (AM) models. The reliability concept applied in this research is grounded in the four pillars of information security: confidentiality, integrity, availability, and authenticity. Confidentiality was subdivided into process automation and privacy, associated with CI and Fernet synchronous cryptography, while integrity and authenticity were linked to the immutability and traceability provided by Blockchain. This reliability concept was evaluated through experiments that subjected the RFoT and a conventional IoT to stress situations. In this research, a conventional IoT is understood as one that does not implement measures to ensure the reliability of collected samples. In these experiments, a AM system was used, based on the Aprendizado Federado (do inglês, *Federated Learning*) (AF) algorithm to train a Rede Neural (do inglês *Neural Network*) (RN) capable of predicting the thermal comfort of an environment, guided by the calculation of the Indice de Desconforto Térmico (IDT). The AM system acted as a consumer in the experiments, to validate whether corrupted data is being propagated by the source and its impact on trained models. The results revealed that a conventional IoT propagates corrupted data, which affects the classification capacity of the models.

Palavras-chave: Internet of Things, Blockchain, Smart Contract, Digital Twin

MSC | 188 | 2024

RFoT: Um Framework para Garantia da Confiabilidade de Dados na Névoa das Coisas

Eliabe Nascimento Silva

UFBA

# RFoT: Um Framework para Garantia da Confiabilidade de Dados na Névoa das Coisas

Eliabe Nascimento Silva

Dissertação de Mestrado

Universidade Federal da Bahia

Programa de Pós-Graduação em Ciência da Computação

Novembro | 2024


Universidade Federal da Bahia
Instituto de Computação

Programa de Pós-Graduação em Ciência da Computação

# RFOT: UM FRAMEWORK PARA GARANTIA DA CONFIABILIDADE DE DADOS NA NÉVOA DAS COISAS

Eliabe Nascimento Silva

DISSERTAÇÃO DE MESTRADO

Salvador
25 de Novembro de 2024

ELIABE NASCIMENTO SILVA

# RFOT: UM FRAMEWORK PARA GARANTIA DA CONFIABILIDADE DE DADOS NA NÉVOA DAS COISAS

Esta Dissertação de Mestrado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Cássio Vinícius Serafim Prazeres

Salvador
25 de Novembro de 2024

Ficha catalográfica elaborada pela Biblioteca Universitária de Ciências e Tecnologias
Prof. Omar Catunda, SIBI – UFBA.

S586 Silva, Eliabe Nascimento.
RFoT: Um Framework para Garantia da Confiabilidade de Dados na Névoa das Coisas / Eliabe Nascimento Silva – Salvador, 2024.
87p.: il.

Orientador: Prof. Dr. Cássio Vinícius Serafim Prazeres.
Dissertação (Mestrado) – Universidade Federal da Bahia, Instituto de Computação, 2024.

1. Internet das Coisas. 2. Blockchain. 3. Contratos Inteligentes. 4. Gêmeo Digital. I. Prazeres, Cássio Vinícius Serafim. II. Universidade Federal da Bahia. Instituto de Computação. III Título.

CDU – 004.41

# TERMO DE APROVAÇÃO

## ELIABE NASCIMENTO SILVA

## RFOT: UM FRAMEWORK PARA GARANTIA DA CONFIABILIDADE DE DADOS NA NÉVOA DAS COISAS

Esta Dissertação de Mestrado foi julgada adequada à obtenção do título de Mestre em Ciência da Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia.

Salvador, 25 de novembro de 2024

gov.br Documento assinado digitalmente
CASSIO VINICIUS SERAFIM PRAZERES
Data: 25/11/2024 12:47:45-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Cássio Vinicius Serafim Prazeres (Orientador - PGCOMP)

gov.br Documento assinado digitalmente
VINICIUS FERNANDES SOARES MOTA
Data: 25/11/2024 14:45:28-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Vinicius Fernandes Soares Mota (UFES)

gov.br Documento assinado digitalmente
BRUNO PEREIRA DOS SANTOS
Data: 25/11/2024 13:01:58-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Bruno Pereira dos Santos (UFBA)

*Ao Deus todo-poderoso que amou o mundo de tal maneira que entregou seu filho unignito para que todo aquele que nele cr no perea, mas tenha vida eterna!*

# AGRADECIMENTOS

Primeiramente a Deus, que por sua graça salvífica e seu amor infinito tem me proporcionado oportunidades de realizar grandes conquistas. A minha família que está sempre ao meu lado, motivando e acalentando nos momentos difíceis da jornada, em especial minha esposa Valquíria Melo e meu pai Edvan Avelar, que com amor e carinho me mostram todos os dias que posso sempre ir além. Aos meus queridos e eternos professores João Neto e Camila Bezerra, que acreditaram no meu potencial, vendo além do que eu poderia imaginar. Ao meu orientador, que com paciência e sabedoria tem me conduzido nessa caminhada e me apresentado um mundo novo. A todos os professores da UFBA que compartilharam um pouco da sua experiência e conhecimento, sempre prontos para dar um impulso na direção correta.

# RESUMO

Este trabalho apresenta a Névoa das Coisas Confiável (do Inglês *Realible Fog of Things*) (RFoT), um framework que integra uma rede de Internet das Coisas (do inglês *Internet of Things*) (IoT) com as tecnologias de Blockchain e Contratos Inteligentes (do Inglês *Smart Contracts*) (CI), para ofertar a IoT como uma fonte de dados confiável para sistemas de treinamento de modelos de Aprendizado de Máquina (do Inglês *Machine Learning*) (AM). O conceito de confiabilidade aplicado nesta pesquisa fundamenta-se nos quatro pilares da segurança da informação: confidencialidade, integridade, disponibilidade e autenticidade. A confidencialidade foi subdividida em automação de processos e privacidade, associadas aos CI e criptografia síncrona de Fernet, enquanto a integridade e a autenticidade foram associadas à imutabilidade e à rastreabilidade proporcionadas pela Blockchain. Esse conceito de confiabilidade foi avaliado por experimentos que submeteram o RFoT e uma IoT convencional a situações de estresse, a fim de comparar suas capacidades protetivas. Nesta pesquisa, se entende como convencional uma IoT que não implementa medidas para garantir a confiabilidade das amostras coletadas. Nesses experimentos, foi utilizado um sistema AM, baseado no algoritmo de Aprendizado Federado (do inglês, *Federated Learning*) (AF) para treinar uma Rede Neural (do inglês *Neural Network*) (RN) capaz de prever o conforto térmico de um ambiente, guiada pelo cálculo do Indice de Desconforto Térmico (IDT). O sistema AM atuou como consumidor nos experimentos, a fim de validar se dados corrompidos estão sendo propagados pela fonte e o impacto gerado nos modelos treinados. Os resultados revelaram que uma IoT convencional propaga dados corrompidos, os quais afetam a capacidade de classificação dos modelos. Em contrapartida, o framework proposto consegue garantir que apenas os dados originais sejam enviados a um consumidor, mesmo que o atacante seja capaz de alterá-los, sem comprometer os princípios que regem a confiabilidade da fonte de dados e o processo de treinamento dos modelos AM. Além disso, para determinar a viabilidade de implementação da arquitetura, foram realizadas análises de custo computacional, recursos de mercado e impacto temporal da integração das tecnologias.

**Palavras-chave:** Internet das Coisas, Blockchain, Contratos Inteligentes, Gêmeo Digital

# ABSTRACT

This work presents the Névoa das Coisas Confiável (do Inglês *Realible Fog of Things*) (RFoT), a framework that integrates an Internet das Coisas (do inglês *Internet of Things*) (IoT) network with Blockchain and Contratos Inteligentes (do Inglês *Smart Contracts*) (CI) technologies, to offer IoT as a reliable data source for training Aprendizado de Máquina (do Inglês *Machine Learning*) (AM) models. The reliability concept applied in this research is grounded in the four pillars of information security: confidentiality, integrity, availability, and authenticity. Confidentiality was subdivided into process automation and privacy, associated with CI and Fernet synchronous cryptography, while integrity and authenticity were linked to the immutability and traceability provided by Blockchain. This reliability concept was evaluated through experiments that subjected the RFoT and a conventional IoT to stress situations. In this research, a conventional IoT is understood as one that does not implement measures to ensure the reliability of collected samples. In these experiments, a AM system was used, based on the Aprendizado Federado (do inglês, *Federated Learning*) (AF) algorithm to train a Rede Neural (do inglês *Neural Network*) (RN) capable of predicting the thermal comfort of an environment, guided by the calculation of the Indice de Desconforto Térmico (IDT). The AM system acted as a consumer in the experiments, to validate whether corrupted data is being propagated by the source and its impact on trained models. The results revealed that a conventional IoT propagates corrupted data, which affects the classification capacity of the models.

**Keywords:** Internet of Things, Blockchain, Smart Contract, Digital Twin

# SUMÁRIO

**Capítulo 1—Introdução** 1

1.1 Motivação e Problema . . . . . 3
1.2 Objetivos . . . . . 5
1.3 Metodologia . . . . . 5
1.4 Contribuições . . . . . 7
1.5 Estrutura do Texto . . . . . 7

**Capítulo 2—Revisão Bibliográfica** 9

2.1 Internet das Coisas . . . . . 9
2.1.1 Problemas de segurança em uma rede Internet das Coisas (do inglês *Internet of Things*) (IoT) ..... 12
2.2 Blockchain ..... 13
2.2.1 Problema dos Generais Bizantinos ..... 15
2.3 Contratos Inteligentes ..... 18
2.4 *Digital Twin* ..... 18
2.4.1 MININET ..... 20
2.4.2 *Intel Lab Data* ..... 20
2.4.3 Protocolo O Universo Acessível das Coisas (do inglês *The Accessible Thing Universe*) (TATU) ..... 21
2.5 Aprendizado de Máquina Federado ..... 22
2.6 Considerações Finais ..... 24

**Capítulo 3—Trabalhos Relacionados** 25

3.1 Visão Geral dos Trabalhos Relacionados ..... 25
3.2 Análise dos Trabalhos Relacionados ..... 25
3.3 Considerações Finais ..... 35

**Capítulo 4—RFOT: Um Framework para Garantia da Confiabilidade de Dados na Névoa das Coisas** 37

4.1 Proposta ..... 37
4.2 Metodologia ..... 37
4.3 Arquitetura do Névoa das Coisas Confiável (do Inglês *Realible Fog of Things*) (RFoT) ..... 39
4.3.1 Parametrização de Confiabilidade (PC) ..... 43

**Capítulo 5—Avaliação** 47

5.1 Planejamento de Experimentos....47
5.1.1 Configuração do Gêmeo Digital....48
5.1.2 Roteiro dos experimentos....49
5.1.3 Coleta e Armazenamento das Amostras....50
5.1.4 Acionamento do Sistema Consumidor....51
5.1.5 Pré-processamento dos Dados....51
5.1.6 Execução do Treinamento dos Modelos....54
5.1.7 Análise dos Resultados de Treinamento....58
5.2 Execução de Experimentos....60
5.2.1 Cenário de experimentos 1....61
5.2.2 Cenário de Experimentos 2....62
5.2.3 Cenário de experimentos 3....65
5.2.4 Cenário de experimentos 4....66
5.3 Resultados e Discussões....68
5.4 Análise de Custo Computacional....69
5.4.1 Análise Assintótica....70
5.4.2 Análise de Custo de Rede....71
5.5 Análise de Custo Arquitetural....71
5.5.1 Camada 1: Sensores....71
5.5.2 Camada 2: Nós de Borda....72
5.5.3 Camada 3: Nuvem....73
5.5.4 Análise de Impacto Temporal da Integração das Tecnologias....73
5.6 Considerações Finais....74

**Capítulo 6—Conclusão** 77

6.1 Resultados Obtidos....77
6.2 Limitações do Trabalho....78
6.3 Trabalhos Futuros....78

**Referências Bibliográficas** 81

# SIGLAS

***loss*** Taxa de perda. 6, 64, 77

**6LowPan** IOV6 em Rede de Área Pessoal sem Fio de Baixo Consumo de Energia (do inglês *IPv6 over Low power Wireless Personal Area Networks*). 12

**AF** Aprendizado Federado (do inglês, *Federated Learning*). ix, xi, xix, 6, 22–24, 29, 49, 51, 54, 55

**AI** Inteligência Artificial (do Inglês *Artificial Intelligence*). 4, 5, 22, 23, 51, 68, 79

**AM** Aprendizado de Máquina (do Inglês *Machine Learning*). ix, xi, xix, 6, 7, 11, 16, 22, 26, 28, 29, 32, 47, 48, 51, 55, 68, 77

**BCD** Blockchain de Dados. 40–42, 65

**BCR** Blockchain de Resultados. xx, 41–43, 51, 65, 67, 70

**BIDTC** Árvore de classificação ID3 Baseado em Blockchain (do inglês *Blockchain-based ID3 Decision Tree Classification*). 32

**CI** Contratos Inteligentes (do Inglês *Smart Contracts*). ix, xi, 1, 5–7, 13, 18, 23–26, 32–35, 37, 38, 40, 43, 50, 60, 69, 75, 77, 78

**CI1** Contrato Inteligente 1. 40

**CI2** Contrato Inteligente 2. 40, 41, 65

**CI3** Contrato Inteligente 3. 40, 70

**CIN** Coletor Inteligente (do inglês *Collective Intelligence Node*). 29

**CNC** Controle de Computação Numérica ( do inglês *computer numerical control*). 30

**CRT** Teorema do Resto Chinês (do inglês *Chinese Remainder Theorem*. 32

**DDoS** Detecção de Ataque de Negação de Serviço (do inglês *Distributed Denial of Service Attack Detection*). 33

**DISFLF** *Framework* de Aprendizado Interativo Seguramente Distribuído (do inglês *Distributed Interactive Secure Federated Learning Framework*). xix, 29

**DoS** negação de serviço (do inglês *Denial of Service*). 33

**Edge** borda de uma rede IoT. 2, 6, 13, 16, 18, 20, 23, 25, 26, 28, 30, 37–39, 48, 73, 78

**FoT** Névoa das Coisas (do inglês *Fog of Things*). 2, 7, 11, 20, 38, 49, 50

**GD** Gêmeo Digital (do inglês *Digital Twin*). xix, 2, 6, 7, 18–21, 24–26, 30, 31, 38, 39, 48, 50, 51, 73, 78

**GECA** Arquitetura Global de Computação em Névoa (do inglês *Global Edge Computing Architecture*. 34

**IDT** Indice de Desconforto Térmico. ix, xi, xxi, 48, 51, 53, 61, 63, 65

**IIOT** Internet das Coisas Industriais (do inglês *Industrial Internet of Things*). 5, 30

**IoT** Internet das Coisas (do inglês *Internet of Things*). ix, xi, xiii, xix–xxi, 1, 3–7, 9–13, 16, 17, 20, 21, 23–35, 37–40, 44, 47–51, 55, 60, 61, 63, 68, 69, 73–75, 77, 78

**IPV6** Protocolo de Internet Versão 6 (do inglês *Internet Protocol Version 6* ). 12

**MAE** Média do erro absoluto. 6, 59, 64

**MBH** Mínimo *hash* de bloco (do inglês *minimum block hash*). 15

**MedAE** Mediana do erro absoluto. 6, 59, 64, 77

**MoT** Prova de Confiança (do inglês *ameasure of trust*). 15

**MQTT** Fila de Transporte de Menssagem de Telemetria (do inglês *Message Queuing Telemetry Transport*). 6, 12, 19, 21, 24, 39, 55

**NASA** Administração Nacional Aeronatica e Espacial (do inglês *National aeronautics and space administration*). 19

**PBFT** Tolerância Bizantina à Falha Prática (do inglês *practical byzantine fault tolerance*). 15

**PC** Parametrização de Confiabilidade. xiii, 43

**PoI** Prova de Importância (do inglês*proof of importance*). 15

**PoS** Prova de Parada (do inglês *proof of stake*). 15

**PoSpace** Prova de Espaço (do inglês *proof of space*). 15

**PoW** Prova de Trabalho (do inglês *proof of work*). 15, 27

**PUF** modelo computacional secreto da função fisicamente não clonável (do inglês *the secret computational model of physically unclonable function*). 32, 33, 45, 78

**QoS** Qualidade de serviço (do inglês *Quality of Service, QoS*. 55

**RFID** Identificação por Rádio Frequência (do inglês *Radio-Frequency Identification)*. 1

**RFoT** Névoa das Coisas Confiável (do Inglês *Realible Fog of Things*). ix, xi, xiii, xix, xx, xxiii, 6, 7, 20, 23, 37–41, 43–45, 47, 49–51, 60, 65, 66, 69–71, 73–75, 77–79

**RFoT-R1** Certificado de Rastreabilidade de Dados. 44

**RFoT-R2** Certificado de Rastreabilidade de Modelos. 44

**RFoT-S1** Certificado de Confiabilidade da Camada de Física. 44, 78

**RFoT-S2** Certificado de Automação. 44

**RFoT-S3** Certificado de Confiabilidade da Camada de Rede. 44

**RFoT-S4** Certificado de Imutabilidade. 44

**RFoT-S5** Certificado de Disponibilidade. 44

**RMSE** Raiz quadrada do erro médio. 6, 59, 64

**RN** Rede Neural (do inglês *Neural Network*). ix, xi

**SHS** Sistemas Inteligentes de Assistência Médica (do inglês *Smart Healthcare Systems*). 27, 28

**SHS2** Sistema Seguro e Inteligênte para Assistência Médica (do inglês *Secured and Smart Healthcare System*). 28

**TATU** O Universo Acessível das Coisas (do inglês *The Accessible Thing Universe*). xiii, 6, 12, 19, 21, 24, 37, 39

**UFP** Processamento por Comentário de Usuário (do inglês *User Feedback Process*). 29

**URLLC** Comunicação Ultra Confiável e de Baixa Latência. 10

# LISTA DE FIGURAS


1.1 Crescimento anual de dispositivos conectados a uma rede IoT . . . . . 3
1.2 Estimativa de dados gerados por dispositivos conectados a uma rede IoT 4

2.1 Cenários fundamentais de uma IoT ..... 11
2.2 Estrutura de um bloco da Blockchain. ..... 14
2.3 Exemplo de Blockchain quando sofre ataque de falsificação. Os blocos seguintes precisam ser reminerados, por isso estão em vermelho. ..... 17
2.4 Envolvimento de um Digital Twin no ciclo de vida de um produto ..... 19
2.5 Mapa de distribuição dos nós de sensores no ambiente do laboratório. ..... 20
2.6 Modelo de comunicação FLOW..... 22
2.7 Classes de Aprendizado de Máquina (do Inglês *Machine Learning*) (AM) 22
2.8 Fluxo do processo de treinamento do modelo Global no algoritmo Aprendizado Federado (do inglês, *Federated Learning*) (AF), para sistema consumidor 24

3.1 Estrutura de camadas proposta por Zhou (2021) ..... 27
3.2 Estrutura esquemática do processo de coleta dos dados biométricos dos esportistas usando IoT ..... 28
3.3 Estrutura esquemática do *Framework* de Aprendizado Interativo Seguramente Distribuído (do inglês *Distributed Interactive Secure Federated Learning Framework*) (DISFLF) ..... 29
3.4 Arquitetura 3D de um Gêmeo Digital (do inglês *Digital Twin*) (GD) para ambientes interiores, combinado com IoT ..... 31
3.5 Arquitetura do sistema de *Smart Home* ..... 33

4.1 Arquitetura da solução proposta intitulada RFoT..... 38
4.2 Fluxograma do algoritmo para o cálculo do *Proof Of Work* ..... 41
4.3 Mapa de rastreio dos elementos que interferem no percurso do dado, desde a coleta, até a entrega a um consumidor ..... 42
4.4 Cobertura de Confiabilidade do RFoT..... 45

5.1 Gráfico de distribuição da temperatura e umidade originais..... 52
5.2 Mapa de rastreio dos elementos que interferem no percurso do dado, desde a coleta até a entrega a um consumidor..... 53
5.3 Gráfico de distribuição da temperatura e umidade após ajustes do posicionamento do separador decimal..... 54
5.4 Histograma das amostras de temperatura e umidade pós-tratamento de *outliers* ..... 55

5.5 Histograma da distribuição dos dados utilizados para o treinamento dos modelos ..... 56
5.6 Gráfico da arquitetura da Rede Neural implementada para geração dos modelos de Aprendizado de Máquina ..... 56
5.7 Comparação entre a arquitetura de uma IoT convencional e do RFoT ..... 60
5.8 *Dashboard* desenvolvido para análise e apresentação de resultados ..... 61
5.9 Comparação dos dados originais e recebidos no cenário 1 ..... 62
5.10 Métricas de classificação do cenário 1 ..... 62
5.11 Métricas de erros do cenário 1 ..... 62
5.12 Matriz de confusão dos resultados de predição do Cenário 1 ..... 63
5.13 Comparação dos dados originais e recebidos no cenário 2 ..... 63
5.14 Comparação dos histogramas dos *datasets* ..... 64
5.15 Métricas de classificação do cenário 2 ..... 64
5.16 Métricas de erros do cenário 2 ..... 64
5.17 Matriz de confusão dos resultados de predição do Cenário 2 ..... 65
5.18 Comparação dos dados originais e recebidos no cenário 3 ..... 66
5.19 Métricas de classificação do cenário 3 ..... 66
5.20 Dados reais e preditos do cenário 3 ..... 66
5.21 Matriz de confusão dos resultados de predição do Cenário 3 ..... 67
5.22 Tela do Dashboard que apresenta as Blockchain de Resultados (BCR) geradas pelo consumidor ..... 67
5.23 Lista de Blockchains criadas para o experimento do cenário 4 ..... 68
5.24 Comparação entre Blockchain válida e inválida no cenário 4 ..... 69
5.25 Comparação das métricas de classificação dos cenários 1 e 2 ..... 69
5.26 Comparação das métricas de classificação dos cenários 3 e 4 ..... 70
5.27 Comparação entre Blockchain válida e inválida no cenário 4 ..... 72
5.28 Análise temporal dos eventos de coleta e registro dos dados na IoT convencional e RFoT ..... 74
5.29 Análise temporal dos eventos de coleta e registro dos dados no RFoT para os desafios de *hash* com cinco e seis zeros no início ..... 74
5.30 Análise temporal dos eventos de coleta e registro dos dados no RFoT para os desafios de *hash* com sete zeros no início ..... 75

# LISTA DE TABELAS

3.1 Matriz de trabalhos relacionados
Fonte: Elaborado pelo autor. ..... 26
3.2 Comparação trabalhos relacionados
Fonte: Elaborado pelo autor. ..... 26
3.3 Comparação entre as propriedades de segurança de uma IoT convencional e a solução proposta por Zhou (2021).
Fonte: Adaptado de Zhou (2021) ..... 27
3.4 Avaliação de segurança da proposta de Patil et al. (2020).
Fonte: Adaptado de Patil et al. (2020) ..... 33

5.1 Tabela do índice Indice de Desconforto Térmico (IDT) para inferência do conforto térmico ..... 48
5.2 Tabela IDT adaptada para apenas duas faixas de conforto térmico. ..... 48
5.3 Tabela de dados de temperatura e umidade ..... 52
5.4 Tabela de dados de temperatura e umidade após pré-processamento ..... 54
5.5 Tabela que apresenta o tamanho em bytes das mensagens trocadas pelos dispositivos com e sem dados ..... 71

# LIST OF QUADROS

5.1 Resultado da análise assintótica para cada camada do RFoT..... 70

Capítulo 1

# INTRODUÇÃO

A Internet das Coisas (IoT) emergiu como um paradigma capaz de conectar bilhões de dispositivos baseados em hardware e software diferentes. Esta vasta e interconectada rede gera um fluxo contínuo de dados, rompendo as barreiras entre o mundo real e o virtual (NGUYEN; NGUYEN; NUGUYEN GIA, 2024).

O termo "Internet of Things" foi apresentado durante o desenvolvimento do MIT Auto-ID, visando padronizar a infraestrutura de computadores que permitia identificar objetos via tecnologia de Identificação por Radiofrequência (Identificação por Rádio Frequência (do inglês *Radio-Frequency Identification)* (RFID)). Atualmente, a IoT é alvo de diversos estudos que buscam solucionar problemas e otimizar atividades em áreas como a indústria, comércio, agricultura e saúde (ATZORI; IERA; MORABITO, 2010). A IoT caracteriza-se pela integração de dispositivos inteligentes equipados com sensores, software e conectividade, permitindo a coleta, troca e análise de dados em tempo real (ALAM; BENAIDA; RAZZAQUE, 2022). Esta infraestrutura possibilita a captura de dados de uma ampla gama de fontes, desde dispositivos pessoais até sistemas industriais complexos.

A natureza distribuída e diversificada da IoT garante cobertura e representação precisa dos ambientes e processos monitorados (HAMDAN; AYYASH; ALMAJALI, 2021). Conforme destacado por (HUSSAIN et al., 2020), uma rede IoT é composta por objetos ou coisas habilitadas que possuem capacidades limitadas de computação, comunicação, armazenamento, software e conectividade, permitindo a interação com o mundo físico para coleta e processamento de dados. O conceito de IoT refere-se à rede coletiva de dispositivos conectados e à tecnologia que facilita a comunicação entre eles. Isso significa que dispositivos do dia a dia, como escovas de dentes, aspiradores, carros e máquinas, podem usar sensores para coletar dados e responder de forma inteligente aos usuários (Amazon Web Services, 2023).

Em diversos contextos, uma rede IoT é capaz de atuar sobre o meio sem a necessidade de intervenção humana. No entanto, existem alguns processos que dependem de um especialista humano. Esse tipo de processo pode ser automatizado a partir de CI, estabelecendo regras para que os dispositivos cumpram o papel do especialista humano sem

perdas e com mais confiabilidade, uma vez que o fator de erro humano não será incluso. Além disso, a IoT oferece oportunidades sem precedentes para a integração e correlação de dados de várias fontes. A interoperabilidade entre dispositivos e plataformas permite que os dados sejam combinados e analisados em conjunto, revelando percepções mais abrangentes e precisas (GARG et al., 2021). Essa capacidade de fusão de dados é particularmente valiosa em aplicações que requerem uma compreensão holística de sistemas complexos, como cidades inteligentes, onde dados de sensores de tráfego, qualidade do ar e consumo de energia podem ser integrados para otimizar os serviços urbanos (CHAMOLA et al., 2020).

A IoT tem sido estudada como uma fonte de dados versátil, adaptável, de baixo custo e alta qualidade, pois os dispositivos IoT modernos são equipados com sensores, capazes de medir variáveis físicas (GOUDARZI et al., 2021). Além disso, os protocolos de comunicação robustos e as técnicas avançadas de processamento de dados garantem a integridade das informações transmitidas. Essa combinação de hardware avançado e algoritmos sofisticados resulta em conjuntos de dados de qualidade, adequados para análises e tomadas de decisão (SYED et al., 2021). Sua capacidade de fornecer informações em tempo real permite a coleta contínua de dados e visão atualizada dos sistemas monitorados (NAUMAN et al., 2020).

Cerca de treze anos após o surgimento da IoT, o termo Névoa das Coisas (do inglês *Fog of Things*) (FoT) foi apresentado ao mundo pela CISCO (BONOMI et al., 2012). Percebeu-se que parte do processamento dos dados poderia ser realizado na borda da rede, mais comumente chamada de borda de uma rede IoT (Edge), visando reduzir custos, otimizar o tempo de resposta da arquitetura e aumentar a qualidade do serviço oferecido.

A Névoa das Coisas (do inglês *Fog of Things*) (FoT) é uma abordagem que combina os princípios da IoT com a computação em névoa, visando superar limitações como latência, segurança e eficiência energética. (MAHMUD; RAMAMOHANARAO; BUYYA, 2023) descrevem o FoT como um paradigma de computação distribuída que *"permite a criação de uma camada intermediária entre dispositivos IoT e a nuvem, facilitando o processamento em tempo real, redução de latência e melhor gerenciamento de recursos em ambientes IoT de larga escala"*.

Alguns dos fatores que potencializam essa perspectiva são a miniaturização dos dispositivos, com aumento da capacidade de processamento e armazenamento. Além disso, a heterogeneidade da IoT abre espaço para integração com diversos tipos de tecnologias, que, aliada à evolução dos *softwares*, hoje possibilita a simulação de diversos tipos de processos reais em ambiente virtualizado. Essa técnica, chamada Gêmeo Digital (do inglês *Digital Twin*) (GD), foi reconhecida pela National Aeronautics and Space Administration (NASA) na década de 1960, que destacou o alto nível de compatibilidade do contexto virtual proposto pelo GD com o mundo real, motivando pesquisas e projetos que visam reduzir custos e antecipar resultados de processos (SHUKLA et al., 2021).

A partir dela, é possível representar toda a arquitetura da borda da rede de uma IoT em um computador. Isso permite testar, modificar e reutilizar topologias de redes inteiras com algumas linhas de código, sem a preocupação com orçamentos e queima de dispositivos. Além disso, ela proporciona elasticidade tanto vertical quanto horizontal, estando limitada apenas aos recursos do dispositivo que roda a simulação.

Essas características são especialmente valiosas em aplicações que requerem respostas rápidas e tomadas de decisão ágeis, como monitoramento de saúde, gerenciamento de tráfego e detecção de falhas em equipamentos industriais. O acesso a dados em tempo real permite que as organizações identifiquem padrões, detectem anomalias e respondam prontamente a mudanças nas condições (DEY et al., 2021).

Com sua capacidade de coletar dados de alta qualidade em tempo real, integrar informações de várias fontes e fornecer uma visão abrangente de sistemas complexos, a IoT está pronta para revolucionar a maneira como tomamos decisões e resolvemos desafios.

## 1.1 MOTIVAÇÃO E PROBLEMA

Com a proposta de explorar e aproveitar todo o potencial da IoT, neste trabalho é proposto abordar questões de privacidade e segurança, bem como promover a padronização e a interoperabilidade, baseado na ideia de que, com isso, é possível desbloquear o potencial da IoT como uma fonte de dados confiável, impulsionando o progresso e transformando setores em todo o mundo.

É essencial reconhecer os desafios associados ao uso da IoT como fonte de dados e implementar medidas robustas de segurança, como criptografia e controle de acesso, para proteger os dados contra acessos não autorizados e violações. Questões de privacidade e segurança são preocupações significativas, uma vez que os dispositivos IoT coletam e transmitem continuamente dados sensíveis (NESHENKO et al., 2019).

Como a Internet representa um dos principais veículos de transmissão de dados e a conexão é feita a partir de dispositivos, o cenário de redes IoT vem apresentando crescimento a cada ano. Conforme a empresa de análise de dados (HOLST, 2024), até o final de 2023, o número de dispositivos conectados a uma rede IoT alcançou cerca de 15,14 bilhões, e esse número deve atingir aproximadamente 17,08 bilhões em 2024, conforme ilustrado na Figura 1.1.

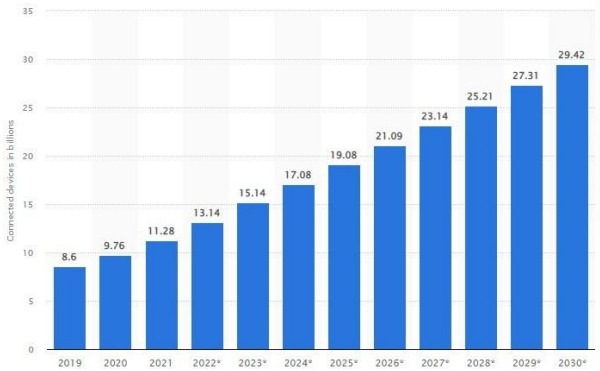



**Figura 1.1** Crescimento anual de dispositivos conectados a uma rede IoT
Fonte: Adaptado de Holst (2024)

Com bilhões de dispositivos, espera-se que a quantidade de dados gerados seja igualmente massiva. Isso sem considerar as características de heterogeneidade das redes IoT, a entidade física alvo das coletas de dados e o tempo de coleta desses dispositivos.

Para exemplificar, considere o volume de dados que pode ser gerado por uma rede IoT no período de um ano, assumindo que todos os dispositivos apresentados no estudo são câmeras IP transmitindo a uma taxa de 256 kbps. Pode-se calcular a quantidade de dados, em terabytes, gerada por um dispositivo, multiplicando o tempo total em segundos em um ano por 256 e dividindo o resultado por $10^9$. Assim, ao multiplicar esse resultado pela quantidade de dispositivos estimada em cada ano, obtém-se a estimativa de dados apresentada na Figura 1.2.

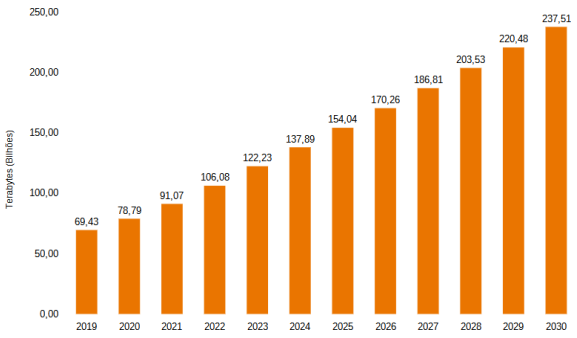



**Figura 1.2** Estimativa de dados gerados por dispositivos conectados a uma rede IoT
Fonte: Adaptado de Holst (2024)

segundo Tiinside (2024) foram registradas diversas operações contra hackers no ano de 2023. As operações resultaram na prisão de centenas de criminosos. Uma operação relevante foi realizada em janeiro de 2023, quando a Europol, a polícia europeia, anunciou a apreensão da infraestrutura ligada ao grupo de ransomware Hive. Ao longo de 2022, o FBI, a polícia federal americana, já havia infiltrado as redes do Hive, capturado mais de 300 chaves de descriptografia, e assim permitindo que empresas comprometidas pelo grupo economizassem cerca de US$ 130 milhões em pagamentos de resgate. Em dezembro de 2023, a polícia francesa também conseguiu deter um cidadão russo suspeito de estar ligado ao grupo Hive. Instituições que desejam ou já utilizam a IoT como fonte de dados e não implementam tecnologias para garantir a confiabilidade dos dados, estão vulneráveis a grupos como o Hive.

Dentro dessa perspectiva, imagine um tipo de ataque para um ambiente inteligente, onde sensores e atuadores conectados a uma central de gerenciamento da rede elétrica, gerenciados por uma Inteligência Artificial (do Inglês *Artificial Intelligence*) (AI) para aprimorar a capacidade de eficiência energética. Grupos como o Hive poderiam realizar ataques de manipulação dessa AI, gerando apagões para facilitar algum ato terrorista. Isso mostra a importância da confiabilidade dos dados utilizados no desenvolvimento de uma solução e como podem impactar uma sociedade.

Visando solucionar essa vulnerabilidade à manipulação das amostras coletadas por uma IoTs, Tripathi, Ahad e Paiva (2020) utilizaram a integração de uma rede IoT com Blockchain para garantir uma arquitetura segura para o armazenamento e compartilhamento dos dados de pacientes, proporcionando segurança aos dados biométricos dos pacientes que precisam de uma proposta de acompanhamento moderno e ágil. Isso porque a principal proposta da tecnologia Blockchain é dificultar a ação de alteração de um

registro, ao ponto de torná-lo imutável. Shan e Mai (2020) também entenderam que a IoT proporciona uma solução acessível e eficiente para a coleta de dados biométricos, possibilitando o treinamento de uma AI capaz de auxiliar o treinamento de atletas para aumento do desempenho.

Baseado nessa mesma perspectiva, Zhou (2021) desenvolveram uma integração de uma Internet das Coisas Industriais (do inglês *Industrial Internet of Things*) (IIOT) com a tecnologia Blockchain para garantir a confiabilidade do processo de treinamento de uma AI capaz de predizer a viabilidade de processos, considerando até o desgaste das peças, que fosse realizado com segurança. Além disso, implementaram uma automação da coleta e processamento dos dados utilizando Contratos Inteligentes (do Inglês *Smart Contracts*) (CI), para minimizar erros humanos e ataques de engenharia social. Assim, baseado em uma análise desses e outros trabalhos, foram desenvolvidas duas hipóteses sobre uma IoT ser capaz de prover confiabilidade das amostras que coleta:

- A confiabilidade de uma fonte é garantida se está proporcionar automação de processos, disponibilidade, privacidade, rastreabilidade e imutabilidade aos dados.
- A integração de uma rede IoT com a Blockchain e Contratos Inteligentes fornece ferramentas que garantem confiabilidade dos dados.

## 1.2 OBJETIVOS

Objetivo Principal: Garantir que os dados gerenciados pelos dispositivos da borda de uma rede IoT sejam confiáveis.

- **Objetivo específico 1** - Garantir a integridade do processo de pertinência dos dados;
- **Objetivo específico 2** - Prover privacidade dos dados na camada de monitoramento;
- **Objetivo específico 3** - Implementar a automação dos processos de coleta, armazenamento e distribuição dos dados;
- **Objetivo específico 4** - Incorporar a rastreabilidade dos dados, a partir da camada física.
- **Objetivo específico 5** - Possibilitar a disponibilidade dos dados, a partir da camada de suporte.

## 1.3 METODOLOGIA

A metodologia adotada neste estudo é estruturada em três etapas distintas. A primeira etapa envolve uma pesquisa abrangente nas principais bases de dados, utilizando variações de combinações das palavras-chave IoT, Blockchain, *Machine Learning*, *privacy* e *tracking*.

Por meio dessa abordagem, foram identificados aproximadamente dois mil artigos nas bases Scopus e Springer Link. Esses artigos foram submetidos a um processo de filtragem, com o critério de seleção centrado na integração da IoT com Blockchain e/ou AM para abordar questões de segurança e privacidade dos dados.

Esse processo resultou em um conjunto de 40 trabalhos considerados mais relevantes, dos quais 25 foram utilizados para extrair conceitos fundamentais, enquanto os 15 restantes apresentavam estratégias distintas para solucionar problemas de segurança de dados e resultados de integração de diferentes tecnologias.

A segunda etapa consistiu no desenvolvimento de um GD utilizando o MINIET como orquestrador de uma topologia de rede virtual, possibilitando a simulação de uma Edge, composta pelos principais dispositivos da rede como sensores, *gateways*, *brokers* e servidores. Além disso, a implementação desse GD viabilizou a integração da Edge com a Blockchain, CI, o protocolo TATU e o protocolo Fila de Transporte de Menssagem de Telemetria (do inglês *Message Queuing Telemetry Transport*) (MQTT). A integração com esses protocolos norteou a comunicação entre os dispositivos e o método de coleta de dados utilizado pelo RFoT.

A terceira etapa é caracterizada pela validação do problema e da solução, por meio de um estudo de caso, onde o RFoT atua como fonte de dados para um sistema de AM, baseado no algoritmo AF, destinado ao treinamento de um modelo capaz de classificar o desconforto térmico de um ambiente, baseado nos dados de temperatura e umidade, que são dados sensíveis em ambientes onde existem atuadores que controlam essas medidas e podem afetar o bem-estar dos indivíduos que ocupam esse espaço. A escolha do algoritmo AF foi baseada na sua implementação para ambientes descentralizados, onde diversos nós da rede atuam paralelamente sobre o contexto. Essa característica foi importante para a validação da capacidade de garantia de confiabilidade do RFoT, pois com esse algoritmo é possível simular vários dispositivos consumindo a IoT como fonte de dados simultaneamente.

Assim, a avaliação do resultado do treinamento dos modelos foi realizada utilizando métricas de classificação como (2) Acurácia e (3) Precisão. Adicionalmente, a classificação foi avaliada por meio de métricas de erro, incluindo (1) Taxa de perda (*loss*), (2) Mediana do erro absoluto (MedAE), (3) Raiz quadrada do erro médio (RMSE) e (4) Média do erro absoluto (MAE). Essas métricas permitiram quantificar o impacto do treinamento com dados corrompidos nos modelos. A confiabilidade é um conceito central neste trabalho, definida como a garantia de automação de processos, disponibilidade, privacidade, rastreabilidade e imutabilidade dos dados. Com base nas hipóteses desenvolvidas nesta pesquisa, ao proporcionar essas características, a arquitetura se torna confiável.

Dado que a solução propõe a integração da IoT com outras abordagens, foi conduzida uma análise do impacto temporal do fluxo da rede, visando mensurar a variação na latência da rede. Além disso, foi realizada uma análise de custo computacional e arquitetural para compreender a aplicabilidade do RFoT. Para mapear a confiabilidade aplicada em cada camada da arquitetura, foi desenvolvida uma abordagem de parametrização de confiabilidade, utilizando certificados para medir automação de processos, disponibilidade, privacidade, rastreabilidade e imutabilidade dos dados. Ao mapear a confiabilidade de cada camada do RFoT, foi possível identificar os limites do framework e analisar os pon-

tos de vulnerabilidade da arquitetura. A combinação desses certificados revela o nível de proteção que a arquitetura pode proporcionar.

## 1.4 CONTRIBUIÇÕES

As contribuições resultantes desse trabalho são:

- Aprimoramento do GD, utilizando como base a arquitetura implementada por Batista, Prazeres e Figueiredo (2017) para o paradigma de Névoa das Coisas, tornando-o ainda mais escalável e com boas práticas de programação para estudos que se baseiem nos princípios da FoT;
- Desenvolvimento de um framework para otimização da confiabilidade de uma IoT, baseado Integração da IoT com a tecnologia Blockchain, Criptografa Síncrona e CI;
- Integração escalável e de baixo custo da IoT com AM, que pode ser utilizada em estudos para desenvolvimento de testes;
- Implementação de um modelo de *Dashboard* para exibição dos dados e resultados providos pelo RFoT;
- Criação de modelo de avaliação da segurança dos dados fornecidos pelas camadas de uma IoT;

## 1.5 ESTRUTURA DO TEXTO

O Capítulo 2 apresenta os conceitos e as tecnologias utilizadas neste trabalho, com foco nos detalhes necessários ao entendimento deste trabalho de mestrado. Em seguida, o Capítulo 3 mostra os principais trabalhos que apresentam soluções similares a este trabalho, utilizando uma ou mais tecnologias abordadas nesta pesquisa. Após os capítulos de contextualização, a proposta de solução desenvolvida nesta pesquisa é detalhada no Capítulo 4. Assim, no capítulo 5 foram desenvolvidos experimentos, submetendo as arquiteturas em processos de estresse, a fim de avaliar a capacidade de garantia de confiabilidade, assim como análises da aplicabilidade do framework. Por fim, o capítulo 6, apresenta a conclusão do trabalho, incluindo uma análise crítica dos resultados obtidos e sugestões para trabalhos futuros. Este capítulo sintetiza as principais contribuições da pesquisa, avalia o impacto dos resultados e propõe direções para estudos subsequentes que possam expandir ou aprofundar os achados deste trabalho.

Capítulo 2

# REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta os principais conceitos das tecnologias e paradigmas que fundamentam este trabalho, guiando o leitor para um melhor entendimento do problema abordado e demonstrando como cada um dos elementos apresentados pode contribuir para a solução proposta.

## 2.1 INTERNET DAS COISAS

A Internet das Coisas (IoT) é um paradigma que possibilita a interoperabilidade entre dispositivos heterogêneos em um ambiente descentralizado, por meio de uma infraestrutura de Internet Novo et al. (2015).

De acordo com Hussain et al. (2020), uma rede IoT é composta por objetos ou "coisas" que possuem capacidades limitadas de computação, comunicação, armazenamento, software e conectividade. Isso permite que esses objetos interajam com o mundo físico, coletando e processando dados. Além disso, oferece a possibilidade de integração com diversos tipos de tecnologias, visando resolver problemas em escalas variadas. O termo "objeto" ou "coisa" refere-se à característica de heterogeneidade, que está relacionada à diversidade de dispositivos que compõem a rede. Quando se discute heterogeneidade em uma IoT, deve-se considerar tanto a estrutura de *hardwares* e *softwares*, quanto os padrões e paradigmas de comunicação.

Essa característica torna a estrutura da rede IoT adaptável aos recursos disponíveis para sua construção. A seguir, são apresentadas algumas das principais características da IoT, conforme descrito por Hussain et al. (2020), que a tornam atraente como fonte de dados.

- **Heterogeneidade**: esta é uma das principais características de uma rede IoT, pois envolve a diversidade de dispositivos e tecnologias que compõem a rede, permitindo a interoperabilidade entre diferentes sistemas.

- **Ampla escala de implementação**: conforme discutido anteriormente, estudos estimam um aumento significativo no número de dispositivos conectados a uma

IoT. Essa característica de ampla escala de implementação apresenta desafios como o projeto de rede e arquitetura de armazenamento para dispositivos inteligentes, protocolos de comunicação de dados eficientes, identificação proativa e proteção da IoT contra ataques maliciosos, além da padronização de tecnologias, dispositivos e interfaces de aplicação.

- **Segurança**: dado que a IoT frequentemente lida com dados sensíveis em contextos sociais ou cotidianos, a segurança da rede é essencial para garantir um funcionamento harmonioso, tanto para os consumidores quanto para os dispositivos.
- **Inteligência**: uma das funcionalidades da IoT é a capacidade de atuar sobre entidades físicas, tomando decisões baseadas nos dados coletados pelos sensores, que ganham significado real ao serem processados.
- **Resiliência**: uma rede IoT deve ser capaz de se recuperar de emergências e desastres sem intervenção externa, possuindo habilidades de auto-organização e autocura.
- **Comunicação de baixo poder e baixo custo**: para proporcionar ampla conectividade de dispositivos, uma rede IoT requer soluções de ultra baixo consumo de energia e custo para operações eficientes.
- **Comunicação Ultra Confiável e de Baixa Latência (URLLC)**: esta propriedade é crucial em contextos de aplicações em tempo real, onde a perda de comunicação pode ser fatal ou causar grandes perdas financeiras, como na indústria 4.0 e em sistemas de saúde inteligentes, exemplificados por cirurgias remotas.
- **Interconectividade**: uma rede IoT deve oferecer conectividade a qualquer hora e em qualquer lugar do mundo para seus dispositivos. Cada rede IoT define sua conectividade com base nos serviços e aplicações que oferece, como em tecnologias conectadas a carros, onde a conectividade é local, e em casas inteligentes, onde a conectividade é global.
- **Comunicação nas proximidades**: esta capacidade da IoT permite que os dispositivos troquem informações sem a necessidade de uma central de intermediação, evitando assim "gargalos" e sobrecargas nos canais de comunicação.

Segundo Group (2017), uma rede IoT consiste basicamente em quatro camadas:

- **Dispositivos ou Física**: responsável por interagir com as entidades físicas de interesse (como temperatura e luminosidade), que serão observadas e/ou modificadas na tomada de decisão. Esta camada inclui sensores (responsáveis pela coleta e transmissão de dados) e atuadores (responsáveis pela modificação de uma ou mais entidades físicas).
- **Monitoramento e Controle**: responsável por executar a lógica de controle em loop, gerenciando o estado dos dispositivos com base nas medições dos sensores. Com base na telemetria dos sensores, alarmes ou eventos podem ser disparados,

iniciando o fluxo de trabalho de forma autônoma via comunicação máquina-máquina ou intervenção humana.

- **Suporte Operacional**: esta camada é responsável por analisar e armazenar o conjunto de dados coletados pelos sensores. A análise foca nos aspectos operacionais do ambiente físico, podendo utilizar interfaces para apresentar os resultados, como painéis de controle, aplicações web ou móveis.
- **Suporte ao Negócio**: para gerenciar a eficiência operacional das camadas anteriores, nesta camada todo o histórico das operações de IoT é analisado e armazenado, criando um registro para as operações da rede. Segundo Group (2017), a análise é realizada em escala de petabyte, auxiliando na geração de opiniões, planejamento de negócios e comparação da eficiência dos processos, utilizando algoritmos de AM para otimização operacional.

Existem três configurações fundamentais de uma rede IoT atualmente, que atendem a contextos de recursos e objetivos variados, como apresentado na Figura 2.1. Pode-se observar que o que diferencia cada configuração é a lógica de processamento, distribuição de recursos e fluxo de comunicação entre as camadas.

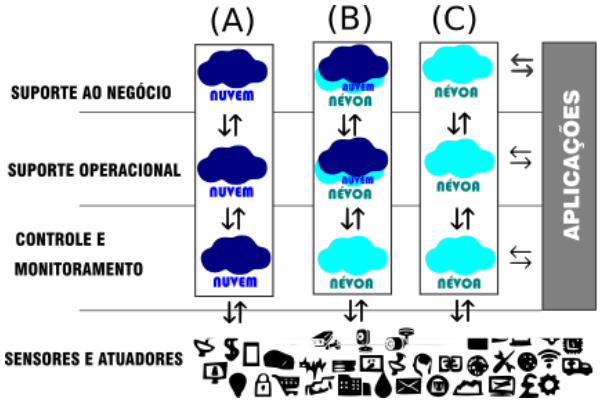



**Figura 2.1** Cenários fundamentais de uma IoT
Fonte: Adaptado de Andrade, Serrano e Prazeres (2018)

Na configuração (A) todo processamento se concentra na nuvem, o que pode tornar o custo do projeto elevado, visto a necessidade de poder de processamento. Esse é o contexto mais tradicional e ainda muito empregado.

Na configuração (B) emprega-se o paradigma FoT, que combina os conceitos de computação em névoa e nuvem, de modo que demandas que necessitam de menor poder de processamento são resolvidas localmente e o que demandar maior poder de processamento pode ser enviado para a nuvem, reduzindo assim a carga de processamento da nuvem e otimizando o fluxo de operações Prazeres e Serrano (2016).

Já a configuração (C) utiliza apenas a computação em névoa, onde as demandas são processadas localmente, sem a necessidade ou uso da computação em nuvem.

A solução proposta nesse trabalho é aplicável aos cenários **B** e **C**, onde existe espaço para processamento na borda da rede.

### 2.1.1 Problemas de segurança em uma rede IoT

Segundo Patil et al. (2020) os dispositivos em uma rede IoT são capazes de produzir uma abundância de dados, mas possuem limitações para garantia da privacidade de dados e usuários.

De acordo com Hussain et al. (2020), existem as soluções de mercado que já foram validadas para ambientes centralizados e garantem a segurança e privacidade. No entanto, são inviáveis para ambientes descentralizados, tal como redes IoT por, basicamente, dois fatores:

- **limitação dos dispositivos IoT**: a capacidade dos dispositivos IoT não são suficientes para suportar a robustez dessas soluções. Além disso, a heterogeneidade dos dispositivos, serviços e aplicações, que uma rede IoT oferece, aumentam o nível de dificuldade, ao tornar a superfície de ataque extensa e variável.

  Nessa pesquisa, essa limitação foi abordada utilizando abordagens compatíveis com ambientes descentralizados, que não resolve em si o problema, mas reduz a limitação e aumenta a aplicabilidade ao considerar a pluralidade de contextos onde a IoT é utilizada.

- **o ambiente de operação**: a internet tradicional não foi projetada para uma rede IoT. Na tentativa de superar essas questões de segurança, uma IoT usa diferentes tecnologias de comunicação, a exemplo do Protocolo de Internet Versão 6 (do inglês *Internet Protocol Version 6* ) (IPV6), Zigbee, IOV6 em Rede de Área Pessoal sem Fio de Baixo Consumo de Energia (do inglês *IPv6 over Low power Wireless Personal Area Networks*) (6LowPan), Bluetooth, z-Wave e WiFi Al-Sarawi et al. (2017).

  Nessa pesquisa, por exemplo, foi utilizado o WiFi, juntamente com os protocolos MQTT e TATU para nortear a comunicação dos dispositivos, utilizando a criptografia síncrona para encriptar os dados trafegados na rede.

O próprio protocolo TCP/IP apresenta desafios de escalabilidade, complexidade, endereçamento e configuração, que limitam o seu uso em ambientes descentralizados e heterogêneos. Existem diversos tipos de ataques para cada camada, mas, em geral, o objetivo do ataque é manipular os dados ou negar o acesso à informação resultante da análise e processamento dos dados. Segue abaixo alguns exemplos de ataques e uma breve explicação de como estão sendo tratados nesta pesquisa.

- **Física**: nessa camada um sensor pode ser corrompido gerando dados de leitura ou coleta falsos. Nessa pesquisa, não foi abordada proteções para a camada física, pois se entendeu que exigiria uma análise um pouco mais extensa por existiram diversos tipos de falso positivo para ataques a esses dispositivos;

- **Monitoramento e Controle**: alguns tipos de ataques à camada de rede são: *man-in-the-middle*, onde o atacante intercepta a comunicação e manipula o que cada uma das partes recebe, podendo substituir completamente o dado original ou fabricar dados (MOHAMMADI et al., 2018). Para impedir esse tipo de ataque, foi

utilizada uma criptografia síncrona, pois mesmo que os dados sejam acessados, não será possível obter a informação sem a chave, que por sua vez é gerenciada por CI;

- **Suporte operacional**: (a) falsificação, em que o dado é alterado; (b) *sybil*, em que uma identidade falsa é criada para inserir dados falsos ou criar ilusões na rede (MOHAMMADI et al., 2018). Esse é o principal tipo de ataque abordado neste trabalho, resolvido com a imutabilidade e rastreabilidade providas pela integração da IoT com a tecnologia Blockchain. É claro que as demais proteções já citadas contribuem também para a eficácia da segurança contra esse ataque;

- **Suporte ao negócio**: o ataque mais sério a essa camada é o DDoS, que sufoca a rede e resulta na negação de serviços para as aplicações (HUSSAIN et al., 2020). No contexto dessa pesquisa, esse tipo de ataque pode comprometer a disponibilidade dos dados, no entanto, aplicações descentralizadas oferecem uma proteção natural para ataques do tipo DDoS, uma vez que os diversos nós da rede podem atender às requisições. Outro fator que pode dificultar a aplicação desse ataque é que o custo computacional para atender cada requisição é da ordem $O(n^2)$, como será apresentado na análise assintótica no capítulo da proposta;

Nessa pesquisa, a tecnologia Blockchain é uma peça importante para garantia da confiabilidade dos dados de uma IoT, visto que se adequa às limitações dos dispositivos da Edge, proporciona imutabilidade, rastreabilidade e pode ser implementada para se adequar ao contexto de aplicabilidade. A seção seguinte apresenta mais detalhes sobre essa tecnologia.

## 2.2 BLOCKCHAIN

Ainda não se sabe ao certo quem desenvolveu essa tecnologia. Existem apenas algumas evidências de que um pseudônimo, Satoshi Nakamoto, publicou um artigo em um fórum de criptografia, elaborando os princípios da criptomoeda Bitcoin e da tecnologia Blockchain no ano de 2008 (ZHOU, 2021).

Alguns anos antes desse evento, Haber e Stornetta (1991) propuseram procedimentos práticos de carimbo de data e hora para documentos digitais, visando impedir que usuários pudessem retroagir ou encaminhar seus documentos, mesmo com o conluio de um serviço de carimbo de data/hora, mantendo a privacidade dos documentos sem a necessidade de manutenção dos carimbos.

Embora Haber e Stornetta (1991) não tenham introduzido o conceito de Blockchain, eles apresentaram uma abordagem para garantir a privacidade e a confiabilidade de documentos digitais. De acordo com Ali, Karimipour e Tariq (2021), o termo Blockchain se popularizou a partir de 2008, com a publicação de Satoshi Nakamoto, o que motivou diversos projetos com implementações de Blockchain.

> "Blockchain é um livro-razão compartilhado e imutável que facilita o processo de registro de transações e o rastreamento de ativos em uma rede empresarial" (IBM, 2021).

Segundo Li et al. (2018), Blockchain pode ser considerado um esquema de arquivamento compartilhado cujo objetivo é permitir a distribuição de registros imutáveis entre diversos atores.

Elmamy et al. (2020) afirma que uma Blockchain proporciona descentralização, imutabilidade, transparência, persistência, auditabilidade, segurança e privacidade aos dados e, consequentemente, ao sistema integrado a ela.

Para entender seu funcionamento, é preciso compreender que uma Blockchain é uma estrutura de blocos que se conectam a partir do número hash do bloco, similar a uma lista encadeada, formando uma corrente ou cadeia de blocos.

Um bloco é composto por quatro elementos básicos: (I) transação ou dado, que representa a informação armazenada no bloco; (II) o *hash* do bloco anterior, utilizado para ligar o bloco atual aos blocos anteriores da cadeia; (III) o *nonce* ou prova de trabalho, utilizado para gerar um *hash* válido para adicionar o bloco à cadeia; (IV) o *hash* do bloco, gerado a partir de todo o conteúdo do bloco. Essas características podem ser visualizadas na Figura 2.2.

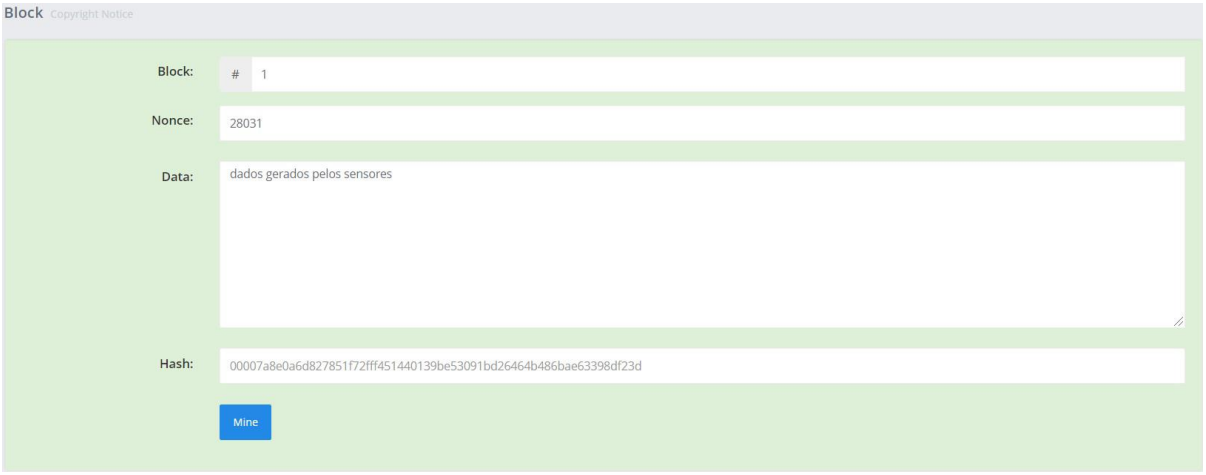



**Figura 2.2** Estrutura de um bloco da Blockchain.
Fonte: Adaptado de lekhasy (2021)

O fato de cada bloco armazenar o número *hash* do bloco anterior garante que cada bloco da cadeia conheça apenas o bloco imediatamente anterior. Como o *hash* do bloco anterior é utilizado para gerar o *hash* do bloco atual, qualquer alteração, mesmo que mínima, resultará em uma mudança no *hash*, causando uma ruptura na cadeia.

A cada novo bloco inserido, uma nova versão da Blockchain é gerada e replicada para todos os dispositivos (nós) que compõem a rede. Portanto, a dificuldade de corromper uma Blockchain é diretamente proporcional ao tamanho da cadeia, ao desafio da prova de trabalho e à quantidade de dispositivos conectados à rede.

Para manter a integridade e a autenticidade da Blockchain, o processo de mineração é gerenciado por um protocolo de consenso, que resolve os conflitos entre mineradores. Esses conflitos surgem devido ao paralelismo na mineração dos blocos pelos mineradores. Um minerador é um nó da rede que se propõe a resolver o desafio de mineração,

caracterizado por um algoritmo. Os algoritmos geralmente encontrados na literatura incluem Prova de Trabalho (do inglês *proof of work*) (PoW), Prova de Espaço (do inglês *proof of space*) (PoSpace), Prova de Confiança (do inglês *ameasure of trust*) (MoT), Prova de Parada (do inglês *proof of stake*) (PoS), Prova de Importância (do inglês*proof of importance*) (PoI), Tolerância Bizantina à Falha Prática (do inglês *practical byzantine fault tolerance*) (PBFT) e Mínimo *hash* de bloco (do inglês *minimum block hash*) (MBH) (ALI; KARIMIPOUR; TARIQ, 2021).

O PoW foi escolhido para implementação da Blockchain nessa pesquisa, pois é um dos protocolos mais comumente utilizados, especialmente em cenários públicos, embora exija maior poder de processamento (ALI; KARIMIPOUR; TARIQ, 2021). Ele propõe aos mineradores um problema complexo, de modo que o primeiro a resolvê-lo ganha o direito de incluir um bloco na cadeia e recebe uma recompensa. Além disso, quando diferentes mineradores resolvem o problema em tempos próximos, o algoritmo de consenso assegura que apenas um deles seja reconhecido como o minerador desse bloco, impedindo a duplicação do *hash* na cadeia, que é então propagada para todos os nós da rede.

### 2.2.1 Problema dos Generais Bizantinos

No contexto de um ambiente de rede descentralizado, vários dispositivos estão envolvidos nos mesmos processos. Portanto, um sistema deve ser capaz de gerenciar situações conflitantes, impedindo falhas e a monopolização de recursos. Isso também serve para proteger o sistema de dispositivos que apresentam comportamentos estranhos ou nocivos, disseminando informações conflitantes na rede. Esse tipo de problema é conhecido como o Problema dos Generais Bizantinos (JIANG; LIAN, 2019).

O problema bizantino pode ser ilustrado por um castelo que está sendo atacado por uma aliança de inimigos, comandada por quatro generais, onde existem apenas duas opções: atacar ou recuar, e cada general deve tomar sua decisão. Além disso, pode haver traidores entre os generais.

Assim, os generais precisam de um algoritmo que garanta duas coisas: (i) todos os generais devem decidir pela mesma ação; (ii) um pequeno contingente de traidores, menor que a quantidade de generais leais, não pode levar os generais leais a optar por um plano ruim.

A moral desse problema é que os traidores só vencem se os generais leais não conseguirem chegar a um consenso (JIANG; LIAN, 2019). Se o algoritmo garantir as duas condições apresentadas anteriormente, mesmo havendo um espião entre os generais, as ações desse espião serão neutralizadas, visto que prevalecerá a decisão da maioria, representada pelos generais leais.

No contexto de uma Blockchain pública, também conhecida como Blockchain sem permissão, por permitir que qualquer indivíduo se associe à rede, existe a possibilidade de um ou mais indivíduos tentarem fraudar a Blockchain, já que têm acesso a todas as informações e podem realizar o processo de mineração. No entanto, o protocolo de consenso garante que, mesmo que a Blockchain daquele nó seja alterada, ela será posteriormente substituída pela versão do restante da rede, neutralizando as ações do atacante ao propagar a versão válida por toda a rede.

Nessa pesquisa foi implementada uma Blockchain privada para integrar com a Edge de uma IoT que atuará como fonte de dados para um sistema de AM. A escolha de uma Blockchain privada foi baseada na flexibilidade de implementação e adequação ao contexto da pesquisa, visando também proporcionar latência menor do que 10 segundos, uma vez que Blockchains publicas podem gerar valores de latência altos, a depender da quantidade de transações e o tipo da operação, tal como apresentado no estudo de Pongnumkul, Siripanpornchana e Thajchayapong (2017) sobre o desempenho da Blockchain pública Ethereum, onde o tempo até a finalização da mineração pode ultrapassar 8 minutos.

Dentro dessa perspectiva, as características de uma Blockchain privada se alinham melhor com a proposta dessa pesquisa, pois ao atuar como fonte de dados, uma rede IoT pode precisar atender uma grande quantidade de clientes simultaneamente. Nesse cenário, uma latência alta pode comprometer a eficiência e aplicabilidade do fornecimento de dados. por isso, quanto mais dispositivos estiverem presentes na Edge, menor será a latência e mais difícil será violá-la.

É nesse ponto que o problema dos generais bizantinos se torna tão importante, pois é a maioria quem decide o que é uma Blockchain válida, ou seja, a imutabilidade proporcionada pela Blockchain se baseia em sempre manter uma quantidade de nós com uma versão sem corrupção superior à quantidade de nós que apresentam uma Blockchain que sofreu alteração. Além disso, corromper uma versão da Blockchain em um nó significa alterar e reminerar toda a versão da Blockchain, antes que o algoritmo de consenso seja acionado para reverter qualquer tentativa de ataque. Desse modo, o atacante precisa infectar, pelo menos, 51% da rede com sua versão fraudulenta para romper a imutabilidade de uma Blockchain. Isso porque a alteração de um bloco gera uma ruptura na cadeia dos blocos subsequentes, pois o hash do bloco também muda com a alteração de qualquer elemento do bloco. (DINH et al., 2018).

No contexto do Bitcoin, por exemplo, ele teria cerca de 10 minutos para realizar a corrupção da rede, superando todos os recursos de segurança que uma Blockchain pública dispõe, incluindo a quebra da criptografia e o desafio da mineração de um novo *hash* (WOO et al., 2021). Baseado no estudo de Pongnumkul, Siripanpornchana e Thajchayapong (2017), no melhor dos casos ele teria cerca de 8 minutos. Já para a Blockchain apresentada nesse estudo, baseado nos experimentos desenvolvidos para o nível de dificuldade 7, esse tempo seria de menos cerca de 2 minutos.

Abaixo seguem as principais características de uma Blockchain, apresentadas por Ali, Karimipour e Tariq (2021) e Nguyen et al. (2020), que permitirão entender como alguns dos problemas apresentados na seção 2.1.1 podem ser resolvidos com Blockchain.

- **Estrutura descentralizada**: refere-se à ausência de necessidade de uma entidade centralizada como intermediária no processo de comunicação entre os nós da rede. Em uma estrutura de rede Blockchain, os nós comunicam-se diretamente entre si. Mesmo que o nó que originou a rede, ou qualquer outro, seja desativado, a rede continua operando normalmente com os nós restantes;

- **Imutabilidade**: esta propriedade assegura que, uma vez inserido na cadeia, um bloco não pode ser alterado. Qualquer modificação em um bloco invalidaria todos os blocos subsequentes devido à geração de um novo *hash*, que seria incompatível

com o registrado pelos blocos subsequentes, resultando em um conflito de versões na rede. Este conflito é resolvido pelo protocolo de consenso, que substituirá a Blockchain minoritária pela versão majoritária nos nós da rede. Este problema é ilustrado na Figura 2.3;

- **Segurança**: todos os dados e informações sobre os usuários da IoT são criptografados, garantindo a proteção contra acessos não autorizados;

- **Transparência**: todos os nós da rede possuem uma versão completa e atualizada da Blockchain, permitindo que cada nó valide e verifique transações individualmente a qualquer momento;

- **Rastreabilidade**: cada informação armazenada na Blockchain recebe um *hash* e um *timestamp*, permitindo que a informação seja rastreável. Além disso, possibilita a restrição de acesso ao conteúdo, específico para cada usuário;

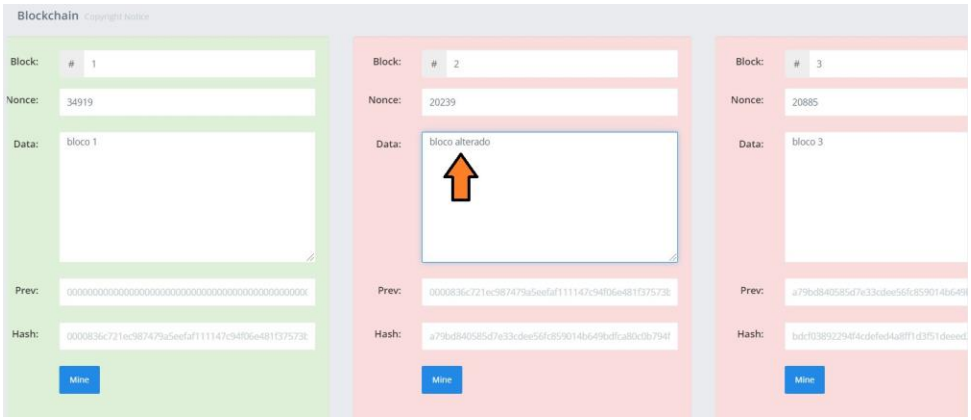



**Figura 2.3** Exemplo de Blockchain quando sofre ataque de falsificação. Os blocos seguintes precisam ser reminerados, por isso estão em vermelho.
Fonte: Adaptado de lekhasy (2021)

Segundo Nguyen et al. (2020), a tecnologia Blockchain pode resolver alguns dos principais problemas enfrentados por redes de IoT que utilizam um modelo de computação centralizada na nuvem. Ao integrar uma rede IoT com a Blockchain, torna-se seguro transferir parte do processamento e da inteligência para a borda da rede.

De acordo com Elmamy et al. (2020), a Blockchain consegue assegurar a confidencialidade, a rastreabilidade, a integridade e a preservação dos dados, sem comprometer a disponibilidade e a privacidade. Esses fatores são fundamentais para o conceito de confidencialidade proposto neste trabalho. Ao analisar todos esses aspectos e os resultados apresentados por Zhou (2021), Tripathi, Ahad e Paiva (2020) e Kebande et al. (2020), foi possível compreender que a Blockchain pode oferecer um conjunto de recursos que contribuem para a garantia da confiabilidade dos dados.

## 2.3 CONTRATOS INTELIGENTES

Em contextos descentralizados, onde a comunicação entre dispositivos ocorre sem a necessidade de intervenção humana, é essencial que os dispositivos possuam a capacidade de gerenciar transações de inclusão e recuperação de dados em uma Blockchain (HUSSAIN et al., 2020).

De acordo com Zyskind, Nathan e Pentland (2015), os contratos inteligentes formam uma parceria robusta com a tecnologia Blockchain, oferecendo automação de processos e segurança para operações do mundo real. No âmbito da Ciência da Computação, um contrato inteligente é definido como um protocolo para a execução de transações computadorizadas que gerencia o cumprimento de um contrato (ZHENG et al., 2018).

A possibilidade de utilizar contratos inteligentes em uma Blockchain é uma das características que a torna atraente para ambientes descentralizados, pois permite a autonomia das transações entre dispositivos da borda (Edge) e a Blockchain.

Segundo (ZHENG et al., 2018), os contratos inteligentes podem ser classificados em duas categorias principais:

- **desenvolvimento**: esta categoria abrange contratos inteligentes desenvolvidos com ou sem o uso de uma plataforma específica; a Ethereum é uma das plataformas que tem se destacado na pesquisa para o desenvolvimento de modelos de Blockchain, como exemplificado pelos modelos propostos por (WOOD et al., 2014) e Kosba et al. (2016);

- **avaliação**: este tipo de contrato é utilizado para avaliar os processos executados, para evitar bugs que possam causar prejuízos significativos; a implementação desses contratos deve ser realizada com cautela para não comprometer o desempenho da aplicação.

Nessa pesquisa, os CI foram utilizados para gerenciar o processamento dos dados desde a coleta até a disponibilização para um consumidor, visando minimizar a necessidade de atuação de um especialista humano em qualquer dessas etapas, pois a interferência do fator humano pode potencializar a falhas e possibilitar ataques de engenharia social para coleta de informações ou influência na atuação do especialista, como abordado por Shukla et al. (2021).

## 2.4 *DIGITAL TWIN*

O GD é uma abordagem que surgiu por volta de 2002, para reproduzir um conjunto de processos do mundo real em um ambiente digital, ou seja, criar uma réplica do ambiente real em um ambiente simulado (SHUKLA et al., 2021). Uma das primeiras tentativas foi a reprodução do ciclo de vida, um conjunto complexo de processos de produção de uma espaçonave.

Segundo Wu et al. (2020), um GD é uma réplica digital de uma entidade física, mantendo uma conexão muito próxima entre ambos. Kraft (2016) afirma que um GD é uma simulação multidisciplinar de um produto do mundo real, que utiliza dados e

informações de sensores como entrada para um modelo que espelha e prediz o estado e comportamento do ciclo de vida do sistema.

Rosen et al. (2015) desenvolveu um estudo para defender a importância do GD na inclusão de autonomia no processo fabril, onde um dos objetivos é remover o fator humano do centro do processo, substituindo-o por um conjunto de paradigmas capazes de estabelecer comunicação e tomar decisões baseadas em análises de dados.

Ele acredita que um GD é uma nova forma de modelar, simular e otimizar tecnologias, fornecendo um grande conjunto de artefatos digitais, devido ao envolvimento do GD em todo o ciclo de vida de um produto, como ilustrado na Figura 2.4.

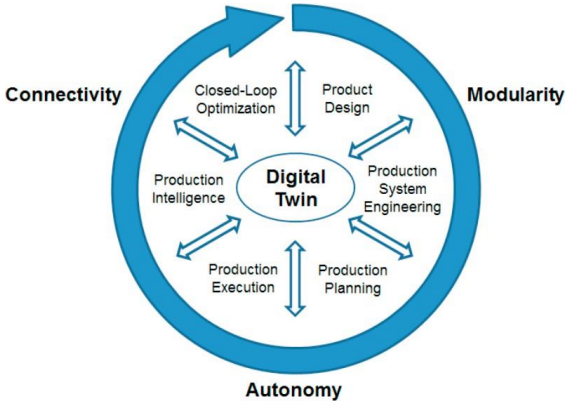



**Figura 2.4** Envolvimento de um Digital Twin no ciclo de vida de um produto
Fonte: Adaptado de Rosen et al. (2015)

Como resultado do estudo, Rosen et al. (2015) demonstram a viabilidade do GD para a representação de ambientes complexos, possibilitando a redução de custos, aumento da eficiência dos processos de desenvolvimento de produtos e redução de impactos por perdas de materiais causadas por falhas.

De acordo com Rosen et al. (2015), a abordagem Digital Twin consiste em uma nova maneira de modelagem, simulação e otimização de tecnologia, que fornece um grande conjunto de artefatos digitais. Para Grieves (2014), é simplesmente uma expressão digital que equivale aos produtos físicos.

Em 2012, essa abordagem foi revisada pela Administração Nacional da Aeronáutica e Espaço (Administração Nacional Aeronatica e Espacial (do inglês *National aeronautics and space administration*) (NASA)), que a declarou como uma simulação multiescala, multifísica, probabilística e de ultra fidelidade, refletindo prontamente o estado de um gêmeo correspondente com base em dados históricos, dados de sensores em tempo real e modelos físicos.

Baseado nos benefícios propostos por um GD e os resultados apresentados por Shukla et al. (2021), Cai et al. (2023) e Kerrison, Jusak e Huang (2023), o estudo apresentado nesse trabalho foi desenvolvido em ambiente virtualizado e por tanto foram utilizados o MININET para criação e gestão da topologia de rede e o *dataset* Intel Lab para representação dos dados coletados na camada física. Também foi possível incorporar os protocolos TATU e MQTT ao GD, utilizando bibliotecas apropriadas da linguagem Python.

### 2.4.1 MININET

O MININET é um projeto de código aberto desenvolvido em Python, disponível na plataforma *GitHub* desde 2014. Ele é projetado para emular uma infraestrutura de rede, incluindo *hosts*, *links* e *switches*, em um único computador Lantz et al. (2022).

O MININET oferece *branches* com diferentes versões do projeto, o que simplifica o processo de instalação. O usuário precisa apenas clonar o repositório e seguir as instruções passo a passo para a instalação. Graças ao MININET, foi possível desenvolver um GD para a Edge, permitindo a virtualização de dispositivos FoT, como sensores e *gateways*.

### 2.4.2 *Intel Lab Data*

O Intel Lab Data é um *dataset* que foi desenvolvido no período de 28 de fevereiro a 05 de março, do ano de 2004, a partir dos dados coletados de um laboratório com 54 diferentes nós de sensores distribuídos no ambiente. Os nós de sensores apresentam dados como temperatura, umidade, luminosidade e voltagem do ambiente. (MADDEN, 2004).

Utilizando esse *dataset*, é possível simular a interação de sensores com as entidades físicas, fornecendo dados para o processo de pertinência na Blockchain e a disponibilização desses dados os consumidores. A Figura 2.5, apresenta um mapa de distribuição dos nós de sensores, espalhados pelo laboratório.

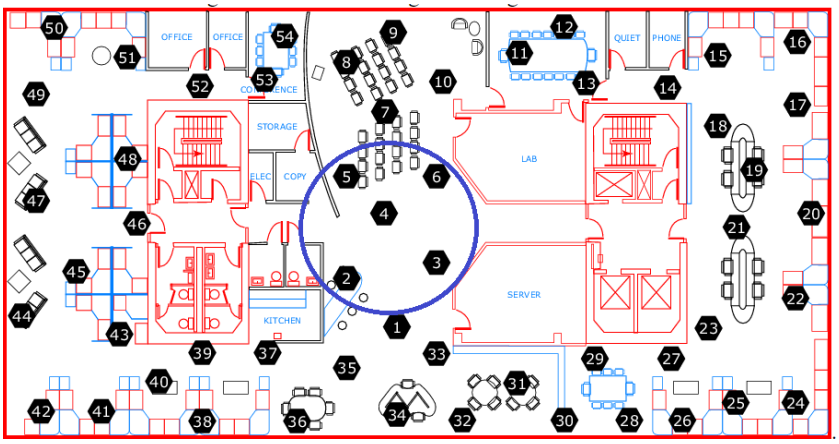



**Figura 2.5** Mapa de distribuição dos nós de sensores no ambiente do laboratório.
Fonte: Adaptado de Madden (2004)

O uso do Intel Lab foi essencial para a simulação da camada Física, provendo amostras de temperatura e umidade utilizados nos experimentos de avaliação da capacidade de provisão de confiabilidade do Framework RFoT. Isso porque a proposta de implementação da solução proposta neste trabalho é utilizar um ambiente virtualizado, simulando os dispositivos da borda de uma rede IoT.

### 2.4.3 Protocolo TATU

O protocolo TATU foi desenvolvido com o objetivo de propor uma nova metodologia de comunicação no contexto de um IoT, utilizando o protocolo MQTT como base de comunicação entre os dispositivos. MQTT foi criado pela IBM por volta de 1990, apresentando os conceitos de tópico, corretor e inscritos, em uma arquitetura de rede de computadores. Os *hosts* se inscrevem em um ou mais tópicos e recebem mensagens de forma assíncrona. Isso reduz o tráfego na rede, pois cada mensagem postada no tópico é automaticamente enviada para todos os assinantes, sem a necessidade de um pedido ou consulta, contribuindo para a redução do congestionamento na rede (LIN et al., 2022).

O protocolo TATU por sua vez estabelece padrões de comunicação entre os dispositivos, otimizando o modo operante do MQTT e abrindo possibilidades de adaptabilidade em diversos contextos. Desse modo, o protocolo TATU propõe três padrões de comunicação (1) baseado em requisições GET/SET, (2) baseado em comunicação em intervalo de tempo pré-determinado (FLOW), (3) comunicação baseada em eventos (BATISTA; PRAZERES; FIGUEIREDO, 2017).

Como o objetivo dessa pesquisa é tornar a IoT uma fonte confiável de dados, foi utilizado o método FLOW por possibilitar um controle da quantidade de informações publicadas na rede, uma vez que se pode configurar o tempo de coleta e publicação de cada sensor. Permitindo alternar o tempo operacional de cada sensor e ajustar a carga de acordo as limitações da rede.

- **Modelo de comunicação FLOW**

  A comunicação no modelo FLOW permite estabelecer intervalos de coleta, em que o destinatário da mensagem não precisa fazer solicitações, pois o remetente enviará a mensagem no intervalo predefinido. E por suas características de controle sobre a previsibilidade e controle do fluxo, assim como controle da massa de dados gerados, este modelo de comunicação foi escolhido para representar a interação no GD.

  Ao utilizar o protocolo TATU, é possível controlar a quantidade de dados que flui no sistema, uma vez que quanto menor o intervalo estabelecido, e quanto maior o número de inscritos em um tópico, mais dados serão gerenciados. A Figura 2.6 apresenta o fluxo de comunicação desse modelo.

  A versão atual do protocolo TATU estabelece que a mensagem publicada no *broker* deve conter os seguintes parâmetros:

  - ***code***: esse parâmetro determina o método HTTP utilizado na mensagem, podendo ser apenas POST ou GET;
  - **post**: recebe o nome do dispositivo de onde a mensagem se origina
  - ***method***: determina o método do protocolo TATU a ser utilizado, ou seja, GET/SET, EVENT ou FLOW.
  - **data**: onde se especifica o valor da informação coletada por um sensor ou um nó de sensores.
  - ***header***: esse parâmetro é composto pelos parâmetros abaixo:

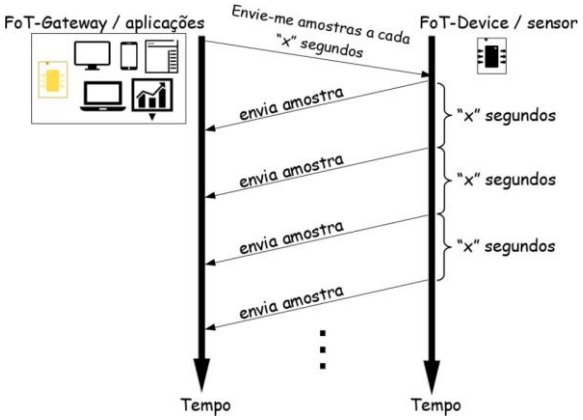



**Figura 2.6** Modelo de comunicação FLOW.
Fonte: Adaptado de Batista, Prazeres e Figueiredo (2017)

- **sensor**: tipo do sensor. Exemplo: temperatura, umidade, etc.
- **device**: nome ou identificador do sensor
- **time**: contém os sub parâmetros "*collect*" que determina o tempo de coleta e "*publish*" que determina o tempo de publicação

## 2.5 APRENDIZADO DE MÁQUINA FEDERADO

De acordo com Hussain et al. (2020), o AM é uma tecnologia utilizada para a criação de AI capazes de realizar tarefas de predição, classificação, agrupamento, regressão, etc. Hoje existem quatro classes de algoritmos, como mostra a Figura 2.7, que possibilitam adequar o treinamento de modelos AM ao contexto de volume, coleta de dados e particularidades do projeto. Nessa pesquisa foi utilizado Aprendizado Federado (do inglês, *Federated Learning*) (AF), baseado na classe de Aprendizado supervisionado, que realiza o treinamento dos modelos AM para classificação ou regressão, baseado em um *dataset* (uma base de dados) rotulado, que apresenta o resultado esperado para cada conjunto de valores. Desse modo, o treinamento consiste em fornecer esses dados para o algoritmo, que avalia as resposta e ajusta os pesos para melhor desempenho

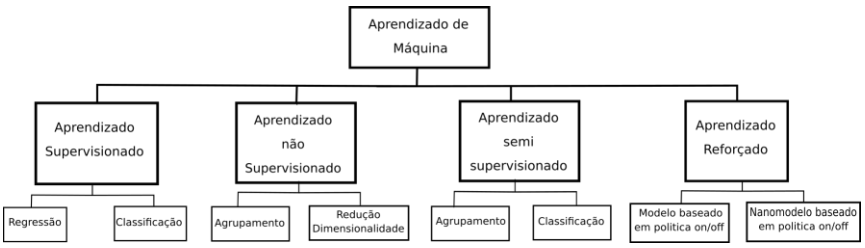



**Figura 2.7** Classes de AM
Fonte: Adaptado de Hussain et al. (2020)

O Aprendizado Federado (do inglês, *Federated Learning*) (AF) é um algoritmo de AM

projetado para ambientes descentralizados, utilizando os nós da rede como uma força-tarefa para o treinamento incremental de uma AI. Sob a perspectiva de paradigma, o AF se alinha com os recursos de uma IoT, já que os dispositivos da Edge oferecem os requisitos mínimos de processamento e memória necessários para a implementação do AF.

Ali, Karimipour e Tariq (2021) afirmam que o modelo de AF não requer um grande conjunto de dados, como no modelo tradicional. Cada nó utiliza um pequeno conjunto de dados para realizar o treinamento. Os autores também defendem que os modelos baseados em AF oferecem soluções que garantem segurança, autenticidade e confiabilidade, mesmo em um contexto de heterogeneidade dos dispositivos. Além disso, o modelo é atualizado iterativamente e sua implementação ocorre em três etapas:

- **Seleção do modelo:** O primeiro passo é a seleção de um modelo, chamado de "modelo primário", que será distribuído entre os dispositivos para iniciar o processo de treinamento do algoritmo AF;

- **Treinamento do modelo local:** Consiste no treinamento do modelo compartilhado, a partir de dados locais de cada dispositivo, individualmente;

- **Agregação do Modelo:** Uma vez que os modelos locais individuais são treinados, eles são enviados a um servidor central, onde ocorre o treinamento de um modelo global, que representa uma atualização do modelo anterior. Uma vez gerado o modelo global, este é compartilhado com os nós, reiniciando todo o processo, mas a partir do novo modelo.

Cada nó da rede realiza um treinamento independente e publica seus resultados na rede. Assim, um dos nós da rede é escolhido para assumir a função de integrar todos os resultados a partir de cálculos matemáticos, gerando versões do modelo global em ciclos. Um ciclo é caracterizado pela distribuição de uma versão do modelo global para os dispositivos da rede. Esses dispositivos realizam um novo treinamento usando o modelo global e geram uma nova versão chamada de modelo local. Ao finalizar o treinamento, o modelo local é publicado para o nó integrador, que, ao receber os modelos de uma quantidade de dispositivos, realiza uma nova integração, reiniciando o ciclo (ALI; KARIMIPOUR; TARIQ, 2021).

Como a proposta dessa pesquisa é apresentar o RFoT como uma fonte de dados confiável, o Aprendizado Federado (do inglês, *Federated Learning*) escolhido por sua compatibilidade com um ambiente distribuído, permitindo simular vários consumidores realizando requisições simultaneamente. Isso é importante para validar a confiabilidade da arquitetura relativo ao impacto gerado nos resultados dos treinamentos realizados pelo consumidor ao receber dados corrompidos. Além disso, permite entender o comportamento da solução mediante a um volume de transações, assim como a automação do processamento via Contratos Inteligentes (do Inglês *Smart Contracts*). Para melhor compreensão, a Figura 2.8 apresenta o fluxo de execução do ciclo de treinamento proposto pelo AF.

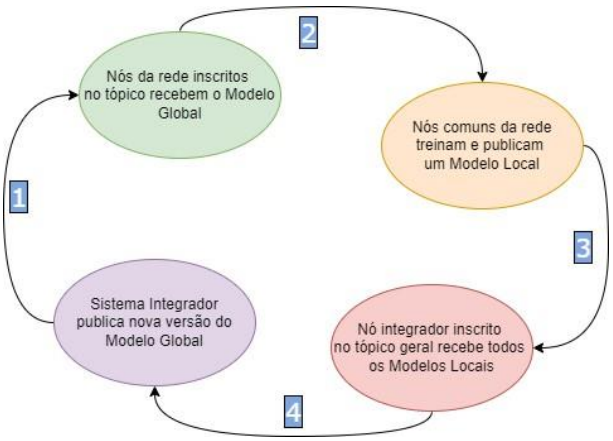



**Figura 2.8** Fluxo do processo de treinamento do modelo Global no algoritmo AF, para sistema consumidor

Fonte: Elaborado pelo autor.

## 2.6 CONSIDERAÇÕES FINAIS

A integração com outras tecnologias é um conceito intrínseco ao paradigma da IoT, favorecendo a aplicabilidade do uso das vantagens de diferentes abordagens, que se complementam como composição de uma solução para diversos problemas. O GD de uma IoT permite que essa integração ocorra com menor custo, com mais flexibilidade e escalabilidade.

O protocolo TATU, juntamente com os CI e o protocolo MQTT favorecem automatizar todo o processo, reduzindo as falhas e interferências do fator humano. Além disso, a Blockchain fornece os subsídios para garantir a automação de processos, disponibilidade, privacidade, rastreabilidade e imutabilidade dos dados gerados pelos sensores virtualizados.

Para corroborar com essa perspectiva, no Capítulo 3 são apresentados os resultados do uso e integração dessas tecnologias, abordagens e paradigmas e como eles conseguiram resolver problemas similares.

Capítulo 3

# TRABALHOS RELACIONADOS

Neste capítulo são apresentadas algumas das abordagens que buscam aprimorar a confiabilidade e a segurança em ambientes IoT. Destacam-se, em particular, as integrações com tecnologias como Blockchain, CI e GD, que têm demonstrado potencial significativo para superar as limitações das arquiteturas IoT convencionais.

Este capítulo está estruturado em duas seções principais. A primeira oferece uma visão geral dos trabalhos relacionados, apresentando quadros comparativos que sintetizam os problemas abordados e as soluções propostas. A segunda seção aprofunda-se na análise detalhada desses trabalhos, agrupando-os por temas relevantes para nossa pesquisa.

Ao final deste capítulo, espera-se fornecer uma compreensão clara do estado da arte em soluções para confiabilidade em IoT, estabelecendo assim o contexto necessário para a apresentação do framework proposto nessa pesquisa.

## 3.1 VISÃO GERAL DOS TRABALHOS RELACIONADOS

A Tabela 3.1 abaixo apresenta um resumo dos problemas e soluções dos trabalhos relacionados a essa pesquisa. A Tabela 3.2 mostra como cada trabalho relacionado aborda os pontos que compõem a confiabilidade da arquitetura proposta.

Além disso, buscou-se evidenciar características que validam a escolha da Blockchain e CI como solução para a automação de processos, disponibilidade, privacidade, rastreabilidade e imutabilidade dos dados, a criptografia para garantia da privacidade dos dados, assim como o uso de um GD para representação virtualizada da borda de uma rede IoT.

A partir desse capítulo também é possível entender a diferença do uso do GD, pois na maioria dos trabalhos que o adotam, o GD é utilizado como uma solução final para um problema de representação de um meio, para funcionar consoante uma IoT.

No entanto, neste trabalho ele representa a própria IoT, mais precisamente a Edge, possibilitando a virtualização de todo o ambiente dessa pesquisa.

## 3.2 ANÁLISE DOS TRABALHOS RELACIONADOS

Segundo Dai, Zheng e Zhang (2019), a tecnologia Blockchain é o complemento perfeito para IoT, dada sua interoperabilidade, provisão de rastreabilidade e imutabilidade de

**Tabela 3.1** Matriz de trabalhos relacionadosFonte: Elaborado pelo autor.

| Autores | Problema | Solução |
|---|---|---|
| **Shukla et al. (2021)** | Segurança de dados em um GD industrial | Integração de Blockchain a arquitetura do GD |
| **Zhou (2021)** | **Aumento de segurança de acesso de dispositivos na IoT** | **Integração de IoT, BLockchain e CI** |
| **Shan e Mai (2020)** | **Monitoramento seguro de dados biométricos de atletas** | **Integração de IoT com blockchain** |
| **Kebande et al. (2020)** | **Coleta segurada de dados modelos de FL** | **Framework que integra IoT, Blockchain e FL** |
| **Tripathi, Ahad e Paiva (2020)** | **Monitoramento de paciêntes remotamente** | **Framework S2HS que integra IoT e Blockchain** |
| **Cai et al. (2023)** | **Monitoramento virtual da qualidade de interiores** | **Desenvolvimento de um GD integrado com IoT** |
| **Kerrison, Jusak e Huang (2023)** | **Monitoramento virtual de pacientes em áreas rurais** | **Desenvolvimento de um GD integrado ao $HC^2$** |
| **Yu et al. (2021)** | **Limitação no volume de dados para treinamento de modelos ML** | **Integração de sistemas distribuidos com Blockchain e CI** |
| **Haque et al. (2020)** | **Privacidade e integridade dos dados de uma rede IoT** | **Integração de IoT com Blockchain** |
| **Patil et al. (2020)** | **Privacidade e integridade dos dados e dispositivos de uma rede IoT** | **Integração de IoT com Blockchain e PUF** |
| **Majeed et al. (2020)** | **Inteligência e segurança em Smart Home Automation System** | **Integração de IoT com Blockchain e AM** |
| **Leduc, Kubler e Georges (2021)** | **Eficiência e confiabilidade de *FarmMarket* na agricultura 4.0** | **Integração de IoT com Blockchain e CI** |
| **Saurabh e Dey (2021)** | **Controle de qualidade de alimentos na agricultura 4.0** | **Integração de IoT com Blockchain e RFID** |
| **Öztürk et al. (2021)** | **Segurança de dados do monitoramento de animais na *Smart Farm*** | **Integração de Global Edge Computing com Blockchain e AM** |

**Tabela 3.2** Comparação trabalhos relacionadosFonte: Elaborado pelo autor.

| Autores | TATU | FoG | Automação de Processos | Rastreabilidade | Privacidade | Imutabilidade | Disponibilidade |
|---|---|---|---|---|---|---|---|
| Zhou (2021) | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Tripathi, Ahad e Paiva (2020) | | | | ✓ | | ✓ | ✓ |
| Shan e Mai (2020) | | | | ✓ | | ✓ | ✓ |
| Kebande et al. (2020) | | | | ✓ | ✓ | ✓ | ✓ |
| Shukla et al. (2021) | | | ✓ | ✓ | | ✓ | |
| Cai et al. (2023) | | | ✓ | | | | ✓ |
| Kerrison, Jusak e Huang (2023) | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Yu et al. (2021) | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Haque et al. (2020) | | | | ✓ | ✓ | ✓ | ✓ |
| Patil et al. (2020) | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Majeed et al. (2020) | | | ✓ | | | | ✓ |
| Leduc, Kubler e Georges (2021) | | | | ✓ | | ✓ | ✓ |
| Saurabh e Dey (2021) | | | ✓ | ✓ | | ✓ | ✓ |
| Öztürk et al. (2021) | | ✓ | ✓ | ✓ | | ✓ | ✓ |
| **Este Trabalho** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

dados, além da escalabilidade de dispositivos.

Baseado nessa perspectiva, Zhou (2021) propuseram uma abordagem de integração entre uma rede IoT e Blockchain, para melhorar a segurança da estrutura de uma rede IoT, contrapondo a versão do sistema que utiliza soluções convencionais e centralizadas.

Assim, eles introduziram uma camada Blockchain entre a camada de Percepção e Aplicação, proporcionando automação, privacidade, rastreabilidade e imutabilidade dos dados, uma vez que os CI se responsabilizam pelos processos de manipulação da informação, garantindo também a validação dos usuários. Essa estrutura é apresentada na Figura 3.1.

A primeira parte da segurança é implementada na camada Marginal, composta por CI, dividida em dois módulos, sendo o primeiro módulo é responsável pelo gerenciamento dos dispositivos da IoT, assim como o sistema de pontos de confiabilidade do usuário. E o segundo módulo é responsável por analisar o comportamento dos dispositivos da rede IoT e prover regras para lidar com eles.


**Figura 3.1** Estrutura de camadas proposta por Zhou (2021)
Fonte: Adaptado de Zhou (2021)

A segunda parte da segurança ocorre na camada de Dados, responsável pela criação dos blocos e propagação da Blockchain entre os nós da rede. Cada bloco contém um conjunto de eventos de acessos dos usuários (permissão, negação e exclusão de acesso).

Como resultado, melhoraram pelo menos seis características relacionadas a estrutura de segurança da IoT com solução convencional, ao ponto de ser uma solução com melhor controle de acesso e desempenho. A Tabela 3.3 apresenta uma análise comparativa, apresentando os pontos de melhoria.

Apesar da integração das tecnologias e considerando o tempo de mineração dos blocos, a latência não foi afetada negativamente, melhorando a atuação do sistema.

**Tabela 3.3** Comparação entre as propriedades de segurança de uma IoT convencional e a solução proposta por Zhou (2021).
Fonte: Adaptado de Zhou (2021)

| **Propriedade** | **Solução baseada em nuvem** | **A solução** |
|---|---|---|
| Modelo de controle de acesso | Senha estática | Senha dinâmica |
| Modelo de armazenamento | *Clipetext* | Criptografia |
| Mecanismo de Segurança | Senha estática | Senha dinâmica, PoW |
| Número de recursos | Alto | Médio |
| Tempo real | Ordinário | Forte |
| Escala | Larga | Médio |

Tripathi, Ahad e Paiva (2020) desenvolveram um framework para ambientes Sistemas Inteligentes de Assistência Médica (do inglês *Smart Healthcare Systems*) (SHS), com finalidade de solucionar o problema de segurança que limitam o uso de IoT convencional para monitoramento remoto de pacientes, baseado no gerenciamento centralizado de dados em nuvem.

Apesar da integração com tecnologias e paradigmas modernos, o modelo convencional enfrenta diversos desafios, sendo um dos principais a garantia de segurança dos dados envolvidos na assistência médica, cuja corrupção ou apropriação podem gerar grande risco à segurança e privacidade dos *stakeholders*, principalmente os pacientes.

Desse modo, Tripathi, Ahad e Paiva (2020) propuseram o Sistema Seguro e Inteligênte para Assistência Médica (do inglês *Secured and Smart Healthcare System*) (SHS2), uma integração de SHS com Blockchain e como resultado, as informações que eram armazenadas de forma centralizada na nuvem passam a ser armazenadas de forma distribuída em uma Blockchain, garantindo a imutabilidade, segurança, rastreabilidade e disponibilidade do dado.

O trabalho de Tripathi, Ahad e Paiva (2020) tem seu núcleo de processamento na nuvem, sendo que a encriptação dos dados ocorre somente no momento de mineração de um bloco, deixando os dados vulneráveis na coleta e no tráfego até a Blockchain.

Diferente do trabalho de Tripathi, Ahad e Paiva (2020), o núcleo do processamento ocorre na Edge, sendo o dado encriptado desde a coleta, garantindo a privacidade em todas as etapas.

Outro contexto onde a segurança de dados é importante é a integração tecnológica no treino e monitoração de esportistas. Em luz desse desafio, Shan e Mai (2020) desenvolveram um framework capaz de analisar dados biométricos coletados de praticantes de esportes, buscando tornar o processo de treinamento mais eficiente e prático.

Como ao se movimentar, os sinais biométricos dos seres humanos geralmente variam, é possível coletar esses sinais a partir de sensores biométricos e baseado nas suas variações, fomentar estratégias para aprimoramento da eficiência da prática dos exercícios, utilizando sistemas de Inteligência Artificial.

Eles utilizaram um nó de sensores composto por um ECG, taxa de respiração e acelerômetro, conectados a uma rede IoT, integrada a uma Blockchain privada, tornando o processo de análise mais imutável e rastreável.

Assim, de forma automatizada os dados coletados são compartilhados com um módulo de AM para limpar e gerenciar as informações, proporcionando uma detecção, classificação e gerenciamento do estado de condicionamento físico do atleta em tempo real. Ver Figura 3.2.

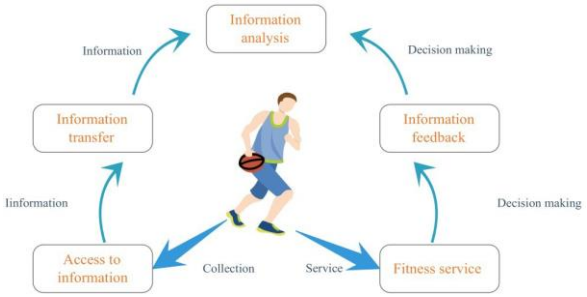



**Figura 3.2** Estrutura esquemática do processo de coleta dos dados biométricos dos esportistas usando IoT

Fonte: Adaptado de Shan e Mai (2020)

Assim como Tripathi, Ahad e Paiva (2020), Shan e Mai (2020) também concentram o processamento na nuvem, deixando toda segurança do dado a cargo da Blockchain também em nuvem, dependendo da limitada segurança provida pela IoT convencional, comprometendo a privacidade do dado nas etapas que antecedem a mineração do bloco.

Kebande et al. (2020) desenvolveram o DISFLF para resolver o problema de confiabilidade, privacidade e segurança dos dados dos usuários, a fim de possibilitar seu uso para o treinamento de modelos de AM, baseado no algoritmo AF.

O DISFLF propõe a integração de uma rede IoT com Blockchain, AF e Processamento por Comentário de Usuário (do inglês *User Feedback Process*) (UFP), onde os dados coletados por dispositivos da borda da rede são armazenados na Blockchain e um especialista humano realiza a análise e rotulação deles. Uma vez que os dados são rotulados, eles são disponibilizados para o treinamento dos modelos de AM.

Nesse sistema os mesmos dispositivos que realizam a coleta também realizam o trenamento de um modelo de AM e depois o enviam para o nó Coletor Inteligente (do inglês *Collective Intelligence Node*) (CIN), que ao receber os modelos, realiza uma composição deles resultando em um modelo global, que por sua vez retorna para os dispositivos locais, reiniciando o ciclo de aprendizado contínuo.

Como no AF o que é enviado para o CIN é apenas o resultado do treinamento dos nós locais, os dados dos usuários nunca deixam os dispositivos da borda da rede, garantindo a privacidade dos usuários. Além disso, a imutabilidade, rastreabilidade e segurança dos dados é garantida pela Blockchain, como apresentado na Figura 3.3.

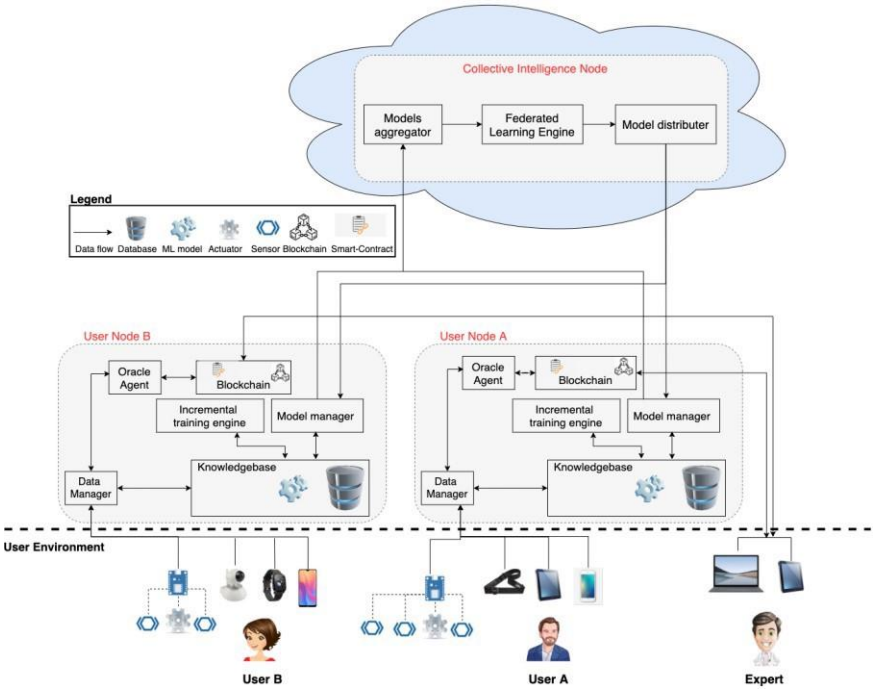



**Figura 3.3** Estrutura esquemática do DISFLF
Fonte: Adaptado de Kebande et al. (2020)

Como resultado, a abordagem proposta por Kebande et al. (2020) proporciona segurança, privacidade, imutabilidade e rastreabilidade dos dados. No entanto, por necessitar de interferência humana, não provê automação e garantia de não manipulação dos dados, uma vez que o especialista humano detém esse controle.

Diferente dos demais trabalhos, os dados dos usuários não são trafegados para nuvem, apenas os resultados dos modelos treinados na Edge e como é possível compor uma engenharia reversa para obter os dados a partir dos modelos trafegados, a privacidade é garantida, por outro lado, não serve com fonte de dados distribuível, pois o dado não pode ser reaproveitado.

Shukla et al. (2021) perceberam que a possibilidade de manipulação dos dados podiam comprometer os resultados do treinamento do modelo que responsável por replicar os processos industriais.

Desse modo, propuseram um GD para Controle de Computação Numérica ( do inglês *computer numerical control*) (CNC), integrada a uma IIOT, que além de replicar todo o processo de produção, também prevê falhas nos produtos, processos e até o desgaste das peças, possibilitando antecipar um problema antes de ir para a produção.

Para corrigir o problema, eles integraram a IIOT a uma Blockchain, que serve de fonte de dados para o treinamento de uma inteligência artificial baseado em uma composição dos algoritmos XGBoost, floresta aleatória e AdaBoost.

O sucesso da integração com a tecnologia Blockchain possibilitou os ataques de manipulação, garantindo a privacidade, segurança, rastreabilidade e imutabilidade do sistema, assim como a confiabilidade do modelo treinado.

E apesar do GD rodar na Edge, eles utilizaram Ethereum, uma abordagem de Blockchain publica para o armazenamento dos dados. Isso garante automação, rastreabilidade e imutabilidade, mas deixa a privacidade dos dados a mercê do modelo, que não usa um sistema de proteção de dados no GD, a exemplo de uma criptografia nas etapas que precedem a mineração de um bloco.

Cai et al. (2023) desenvolveram um GD para monitoramento de ambientes internos de construções, que abrange desde uma residência familiar à grandes empreendimentos como escritórios, hospitais e qualquer outro tipo de ambiente que possibilite a instalação de sensores, que tenham pessoas habitando ou desenvolvendo algum tipo de atividade no local.

A ideia principal da criação de um GD para monitoramento desses ambientes é prevenir que esses espaços alcancem níveis prejudiciais de patógenos, utilizando recursos computacionais que não dependem de interferências humanas e que proporcionam baixo custo de implantação.

A capacidade de realizar esse monitoramento assistido por um GD, possibilita prever e propor ações que impeçam a proliferação de doenças, tal como a COVID-19, que gerou uma pandemia entre os anos de 2020 e 2023, tendo como uma das medidas de prevenção da proliferação da doença a segregação das pessoas em suas próprias residências.

Essas entidades físicas podem ser mensuradas por sensores presentes nesses ambientes, o que tornou a IoT uma abordagem conveniente para essa tarefa, visto que além de ser descentralizada, pode ser integrada com o GD, tornando-o ainda mais eficiente na representação do espaço.

A coleta é realizada por dois tipos de sensores, os sensores de ambiente e sensores biológicos. Os sensores biológicos são responsáveis por coletar dados dos indivíduos presentes no espaço, tal como temperatura da pele, ritmo cardíaco e pressão sanguínea.

Já os sensores de ambiente são responsáveis por coletar dados relativos ao espaço

de análise, tal como a movimentação captada por câmeras e sensores de presença, a luminosidade e umidade, ou seja, monitoram as características do ambiente que podem ser afetadas pela aglomeração e gerar o aumento da cultura de patógenos.

Como resultado, Cai et al. (2023) desenvolveram um sistema de monitoramento utilizando GD, capaz de representar todo o espaço interno em modelos 3D, com mapeamento dos níveis patógenos presentes nesses ambientes, possibilitando disparar alerta e ações preventivas para evitar a contaminação e proliferação desses organismos. Ver Figura 3.4

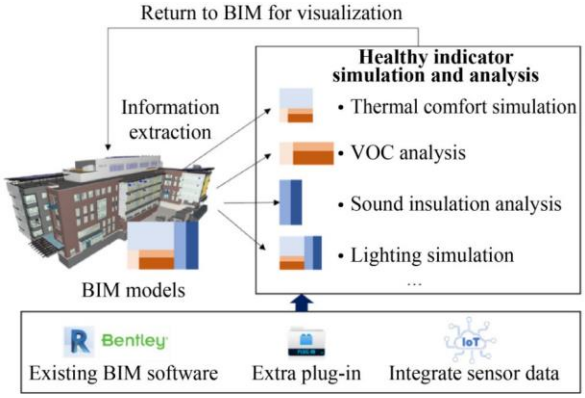



**Figura 3.4** Arquitetura 3D de um GD para ambientes interiores, combinado com IoT
Fonte: Adaptado de Cai et al. (2023)

Apesar de Cai et al. (2023) citarem uma preocupação com a privacidade e segurança, nenhuma solução foi implementada, deixando a proposta vulnerável a diversos tipos de ataques que comprometem não somente a segurança dos indivíduos, mas também a validade dos dados.

Kerrison, Jusak e Huang (2023) desenvolveram um sistema para atender às necessidades de monitoramento de pacientes em áreas rurais de modo seguro e descentralizado, baseados em IoT e integrados com *Blcockchain* para conexões de baixa largura de banda. Eles utilizaram um GD para virtualizar os componentes de rede, o qual também é responsável pela alternação do canal de comunicação.

Como as áreas rurais, em geral, apresentam densidade demográfica baixa, baixa disponibilidade de recursos de rede e grandes barreiras naturais, Kerrison, Jusak e Huang (2023) combinaram dois canais de comunicação que se alternam para garantir a disponibilidade do canal de comunicação.

Nesse contexto observa-se que mesmo em arquiteturas de baixa largura de banda, a IoT integrada à Blockchain representa uma solução eficaz, capaz de prover a segurança e os recursos necessários, sem afetar a latência de comunicação entre os dispositivos, visto que a combinação dos canais contorna a dificuldade do terreno.

Apesar de utilizarem computação baseada em nuvem e a Ethereum com Blockchain publica, eles garantem a criptografia dos dados de ponta a ponta, desde sua coleta, proporcionando segurança, privacidade, rastreabilidade, automação e imutabilidade.

Como um dos desafios da área de inteligência artificial, tem sido a melhora a atuação da classificação dos modelos, Yu et al. (2021) mostram que o aumento do volume de dados

resulta em aumento do desempenho de treinamento dos modelos de AM, favorecendo o treinamento de modelos mais eficientes na classificação.

Ao levar em consideração o fato de os dados a serem compartilhados na rede e sua qualidade, eles propuseram o Árvore de classificação ID3 Baseado em Blockchain (do inglês *Blockchain-based ID3 Decision Tree Classification*) (BIDTC) visando explorar a combinação das vantagens do *Blockchain-based ID3 Decision*, da criptografia homomórfica aprimorada e simulação de CI, que considera a privacidade e valor dos dados para classificá-los,

Com isso eles conseguiram garantir a segurança e a qualidade dos dados, possibilitando o aumento o volume de dados disponíveis para o treinamento e validando que a combinação das tecnologias é capaz de contribuir para o aprimoramento da capacidade de classificação e assim da qualidade dos modelos treinados.

O trabalho de Yu et al. (2021) valida a IoT como fonte de dados, tendo como resultado a melhora do desempenho dos modelos de AM. Além disso, utiliza a Blockchain e os CI como ferramentas capazes de garantir a privacidade, segurança, rastreabilidade e imutabilidade, que viabilizam a preservação da qualidade dos dados.

Haque et al. (2020) apresentam uma perspectiva similar à de Yu et al. (2021), de que quanto mais dados de qualidade forem utilizados no treinamento de modelos AM, maior será a eficiência da classificação. Desse modo, eles propõem uma integração de IoT com Blockchain, utilizando um sistema de criptografia homomórfica parcial.

Como resultado, eles possibilitam conexão entre provedores de dados de diferentes IoT e consumidores de dados para treinamento de modelos, garantindo a privacidade e integridade do dado e dos *stakeholder*, vencendo a limitação da insegurança de dados de uma IoT. Além disso, mostraram que a Blockchain não gerou grandes impactos no processo de treinamento e comunicação.

O trabalho de Haque et al. (2020) e Yu et al. (2021) mostram que a IoT tem a capacidade de ser uma fonte de dados. Além disso, mostram que exite a possibilidade de consumidores serem conectados a essa fonte de dados de forma segura.

A grande questão do trabalho desses trabalhos é que ele não é capaz de garantir a confiabilidade dos dados fornecidos pelos provedores, uma vez que não há como monitorar o processo de coleta e armazenamento dos dados e não foi estabelecido nenhum protocolo para validar um provedor, baseado na confiabilidade que provê.

Patil et al. (2020) afirmam que uma rede IoT convencional possui limitações para garantir a privacidade dos dados e assim dos usuários, por ser vulnerável a um grande conjunto de ataques cibernéticos, que comprometem a privacidade dos dados, o que os torna não confiáveis.

Para sanar a vulnerabilidade de uma rede IoT a ataques à privacidade e segurança dos dados, eles desenvolveram um protocolo de preservação da privacidade eficiente baseado em Blockchain e o modelo computacional secreto da função fisicamente não clonável (do inglês *the secret computational model of physically unclonable function*) (PUF).

Esse protocolo faz parte do framework desenvolvido por eles para possibilitar um processo seguro de comunicação entre usuários e dispositivos IoT, intermediado por uma Blockchain publica, utilizando o Teorema do Resto Chinês (do inglês *Chinese Remainder Theorem* (CRT) e função *hash*, possibilitando um método ágil de validação de dispositivos.

Em seu estudo, Patil et al. (2020) realizaram uma série de análises da perspectiva de segurança do modelo proposto, que garantem segurança, automação, privacidade, rastreabilidade e disponibilidade, uma vez que o próprio sistema se encarrega da validação dos dispositivos. Ver A Tabela 3.4.

**Tabela 3.4** Avaliação de segurança da proposta de Patil et al. (2020).
Fonte: Adaptado de Patil et al. (2020)

| | |
|---|---|
| **Privacidade** | **PUF oferece padrão de randômicos únicos em dissipativos IoT e os proprietários dos dados tem controle e acesso total aos mecanismos de segurança** |
| **Integridade** | **Blockchain implementa funções de hash que validam a integridade dos dados, enquanto o identificador único ID no modelo PUF valida a proveniência do dado.** |
| **Verificação rápida** | **PUF baseado em Blockchain possibilita uma rápida verificação do processo, devido ao fato de o sistema proposto extrair a computação secreta necessária do modelo PUF e armazena-la na base de dados dos mineradores** |
| **Controle de usuários** | **O proprietário do dado implanta CI para controle de acesso definindo de políticas para controle de acesso aos dados e garantia da privacidade dos usuários** |

Como resultado, a solução garante proteção contra ataques do tipo negação de serviço (do inglês *Denial of Service*) (DoS), Detecção de Ataque de Negação de Serviço (do inglês *Distributed Denial of Service Attack Detection*) (DDoS), ataques de personificação, acesso e manipulação dos dados, uma vez registrados na Blockchain. No entanto, o uso de recursos da cloud tem seu preço em desempenho e financeiro, devido à contratação de recursos.

Majeed et al. (2020) propuseram um sistema de gerenciamento de dispositivos de *Smart Home* com objetivo de prover redução de custo com consumo de energia e extensão da vida útil dos dispositivos ao possibilitar um gerenciamento do estado dos mesmos, utilizando uma integração com Raspberry Pi e Arduindo. Ver Figura 3.5

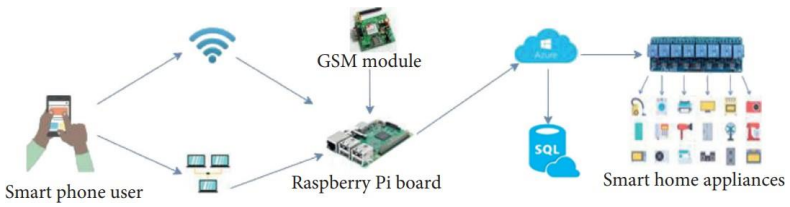



**Figura 3.5** Arquitetura do sistema de *Smart Home*
Fonte: Adaptado de Majeed et al. (2020)

A solução proporciona uma arquitetura de baixo custo, com predição de consumo, capacidade de restaurar um estado anterior em casos de desastres, inteligência artificial com capacidade de identificar o estado dos dispositivos e garantia de autenticação e identificação de usuários. A solução é baseada em nuvem e propõe resolver problemas e limitações do estado da arte.

A identificação e autenticação dos usuários é gerenciada utilizando Blockchain, visando prover uma alta disponibilidade, segurança, mecanismos descentralizados e uma forma segura de realizar transferência de dados entre dispositivos IoT, servidores, aplicações e usuários.

O trabalho Majeed et al. (2020) utilizam a Blockchain apenas como recurso de autenticação de dispositivos e usuários, não sendo o foco a imutabilidade do dado, perdendo de proporcionar rastreabilidade e imutabilidade.

Leduc, Kubler e Georges (2021) desenvolveram um *framework* para *FarmMarkets*, a fim de inovar o mercado agrícola, possibilitando que agricultores, entregadores e varejistas possam interagir de forma segura e comercializar produtos agrícolas com mais velocidade em comparação com os demais *FarmMarkets*.

Esse é um sistema que se enquadra no conceito de agricultura 4.0 e que tem se expandido com abertura do ecossistema agrícola para o uso de tecnologias como IoT. Geralmente os sistemas desenvolvidos para esse mercado se concentram na rastreabilidade, no entanto, o sistema de Leduc, Kubler e Georges (2021) foca na comercialização de produtos.

A integração da IoT com a Blockchain e CI, nesse trabalho, foi realizada com foco em aumentar a qualidade e velocidade de nos serviços oferecido pelo *FarmMarket*, além de desfrutar da segurança, privacidade, rastreabilidade, imutabilidade e disponibilidade oferecidos pela tecnologia Blockchain.

Não é mencionado o uso de nenhuma criptografia de dados, para garantia da privacidade da informação coletada e trafegada, o que torna a arquitetura vulnerável a ataques de manipulação durante a coleta ou trafego dos dados. Saurabh e Dey (2021) desenvolveram uma integração entre IoT, Blockchain e RFID para controle da qualidade de alimentos em cadeia de suprimentos.

Eles se concentraram desenvolver um sistema para garantir a confiabilidade sobre a qualidade e procedência dos alimentos, a fim de ser possível auditar todo o processo, atestando assim a confiabilidade da qualidade dos alimentos.

Eles focaram na rastreabilidade e imutabilidade da informação, a fim de garantir o monitoramento dos alimentos, deixando a segurança e a privacidade dos dados a mercê da arquitetura de uma IoT convencional. Öztürk et al. (2021) utilizaram a Arquitetura Global de Computação em Névoa (do inglês *Global Edge Computing Architecture* (GECA), proposto por Sittón-Candanedo et al. (2019), para montar um sistema de monitoramento inteligente do gado em fazendas, utilizando a integração da IoT com a Blockchain para garantir a rastreabilidade, privacidade, imutabilidade e disponibilidade dos dados.

Eles objetivaram uma arquitetura que otimizasse o tempo de resposta dos serviços, assim como possibilitar a rastreabilidade do estado de saúde do gado e dos produtos lacteis produzidos e comercializados a partir desses animais, provendo tanto aos produtores como os consumidores o acompanhamento em tempo real.

O grande detalhe desse trabalho é o uso da computação na borda da rede para otimizar o processamento e serviços de entrega da informação, mostrando que a computação em névoa pode ser uma solução eficiente para redução de custos dos serviços da computação em nuvem.

O trabalho de Öztürk et al. (2021) não deixa claro se foi utilizado uma Blockchain

publica ou integrada a arquitetura da camada de IoT, além disso, não utiliza um sistema de criptografia para garantir a privacidade dos dados, senso susceptível a taques de *man-in-the-middle*.

## 3.3 CONSIDERAÇÕES FINAIS

O ambiente descentralizado, apesar de oferecer a possibilidade de conectar de forma autônoma uma quantidade praticamente ilimitada de dispositivos, torna-se vulnerável de diversas formas.

Como o contexto convencional centralizado não apresenta soluções viáveis para os problemas de ambientes descentralizados, a estratégia atualmente mais utilizada é a integração com outras tecnologias descentralizadas. Estas trabalham no mesmo contexto descentralizado e são soluções eficientes e de baixo custo, que utilizam a heterogeneidade e o volume de dados a seu favor para prover segurança.

Seguindo essa perspectiva, a integração de IoT, CI e Blockchain apresenta múltiplos mecanismos para garantir a confiabilidade dos dados.

Capítulo 4

# RFOT: UM FRAMEWORK PARA GARANTIA DA CONFIABILIDADE DE DADOS NA NÉVOA DAS COISAS

Este capítulo apresenta a arquitetura do framework RFoT e as técnicas utilizadas para mitigar as vulnerabilidades de uma IoT convencional. Na Seção 4.3.1, é introduzido um sistema de parametrização que avalia, camada por camada, a vulnerabilidade dos dados e verifica se a arquitetura propõe uma estratégia de solução.

Esse sistema possibilitou uma análise direcionada à confiabilidade dos dados, destacando a aplicabilidade de cada tecnologia utilizada para compor o framework proposto.

## 4.1 PROPOSTA

Como apresentado no Capítulo 3, a rede IoT tem sido estudada como uma fonte provedora de dados devido à sua capacidade de oferecer grandes volumes de dados a baixo custo, de forma descentralizada e autônoma. No entanto, a vulnerabilidade da arquitetura de uma rede IoT convencional limita a confiabilidade desses dados.

As hipóteses abordadas neste trabalho baseiam-se na integração da IoT com outras tecnologias para gerar uma versão capaz de prover dados com confiabilidade, partindo do princípio de que a confiabilidade é assegurada ao garantir automação, disponibilidade, privacidade, rastreabilidade e imutabilidade dos dados. Ver Capítulo 1.1.

Neste trabalho, propõe-se o RFoT, um framework composto pela integração de Blockchain, protocolo TATU e CI para criar uma versão de IoT que ofereça automação, disponibilidade, privacidade, rastreabilidade e imutabilidade dos dados na Edge, podendo assim ser utilizada como uma fonte confiável de dados. A Figura 4.1 apresenta a arquitetura proposta.

## 4.2 METODOLOGIA

Após a etapa de pesquisa e filtragem dos trabalhos que apresentavam propostas de solução para o problema de segurança dos dados no contexto de uma IoT, foram analisadas quais

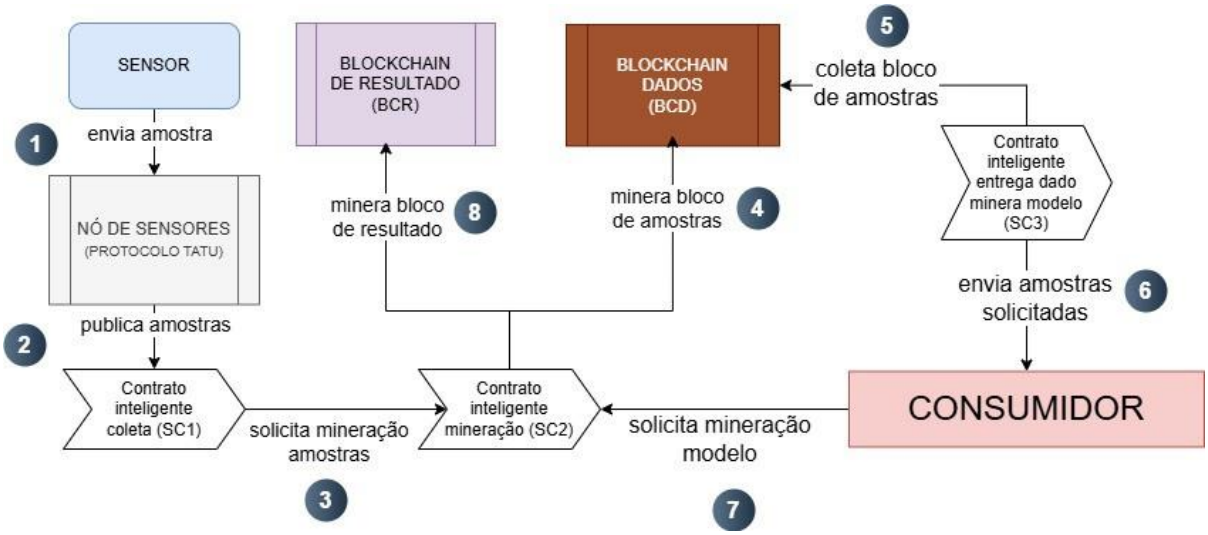



**Figura 4.1** Arquitetura da solução proposta intitulada RFoT
Fonte: Elaborado pelo autor.

soluções gerariam melhor impacto, considerando o custo de implementação e a eficiência na provisão de automação, disponibilidade, privacidade, rastreabilidade e imutabilidade.

Das estratégias analisadas, a Blockchain em conjunto com CI foram as que apresentaram melhores resultados e maior frequência nos trabalhos. Com os resultados dos estudos, concluiu-se que a implementação da solução deveria ocorrer na Edge, em vez de abranger todo o contexto da IoT, pois já existem soluções para outras camadas.

Desse modo, utilizando as ferramentas proporcionadas pelo MININET, foi desenvolvido um GD para a Edge a fim de simular os recursos da rede propostos pelo modelo FoT. A topologia de rede proposta é composta por sensores, *switches* e *gateways*.

Como o MININET oferece a possibilidade de escalar recursos, o GD representa um ambiente virtual capaz de ser adaptado para diferentes situações reais, sem a necessidade de investimento em equipamentos de uma rede real.

Com um ambiente funcional, foi possível realizar a integração do GD representando a Edge com a Blockchain e CI. Como toda a implementação foi realizada utilizando a linguagem Python, os elementos da arquitetura foram desenvolvidos utilizando o paradigma de orientação a objetos e boas práticas de programação embasadas nos princípios SOLID, DRY e Clean Code.

Após a estruturação da integração e testes de funcionamento da arquitetura, foi implementado um sistema de criptografia dos dados desde a coleta até seu registro na Blockchain, visando dificultar tanto a manipulação quanto ataques do tipo *man-in-the-middle*.

A fim de validar o nível de confiabilidade que a solução proporciona, foi desenvolvido um estudo de caso sobre o qual foram realizados testes de impacto temporal da integração das tecnologias, testes de corrupção e uma comparação com a arquitetura convencional da Edge para validar o problema e a solução proposta.

Com base nos resultados, foi desenvolvido um sistema de parametrização que permite

classificar o nível de confiabilidade da arquitetura, baseado em uma análise da gestão dos dados em cada etapa de manipulação dos dados pela Edge.

## 4.3 ARQUITETURA DO RFoT

Como o objetivo do GD neste trabalho é propor uma virtualização da Edge, as entidades físicas são representadas por amostras de temperatura e umidade, coletadas do *dataset* Intel Lab.

A arquitetura do RFoT foi organizada de forma similar à composição apresentada por Hussain et al. (2020), com adaptações voltadas para o contexto da atuação de um IoT como fonte de dados, seguindo assim, o padrão de quatro camadas: (1) Física, (2) Monitoramento e Controle, (3) Suporte Operacional e (4) Suporte ao Negócio, dividindo a responsabilidade da confiabilidade nos processos de coleta, transporte, armazenamento e disponibilização de informações. A análise abaixo apresenta uma descrição de cada uma das camadas, as vulnerabilidades identificadas e as soluções implementadas para correção.

- **Camada Física**: Como o RFoT atua como fonte de dados, foram implementados apenas recursos para simular o processo de coleta de dados pelos sensores. Aqui temos o primeiro ponto que diferencia o RFoT da arquitetura apresentada por Hussain et al. (2020), que propõe tanto sensores quanto atuadores. Desse modo, utilizou-se o Intel Lab para representar os dados das entidades físicas. Além disso, essa camada foi implementada com baixo nível de acoplamento com o *dataset*, permitindo a substituição do dataset Intel Lab por qualquer outro, sem prejudicar as camadas e processos subsequentes. As amostras coletadas são organizadas por um nó de sensores em uma mensagem em formato JSON. Este JSON é encriptado e enviado para a camada de Monitoramento e Controle.
  - **Vulnerabilidade**: coleta ou envio de dados distorcidos, que podem ter sido manipulados;
  - **Solução**: Esta pesquisa não aborda soluções para problemas nessa camada, ficando para trabalhos futuros realizar uma abordagem de integração com uma solução, a exemplo do protocolo PUF apresentado Patil et al. (2020);
- **Camada de Monitoramento e Controle**: esta camada é responsável pela conexão entre a camada Física e a camada de Suporte. Nela é implementado o método FLOW do protocolo TATU, utilizado para realizar o monitoramento e controle do processo de coleta das amostras. Esta camada é responsável por solicitar à camada Física que realize a coleta de amostras. Quando a camada física devolve as amostras, a camada de Monitoramento e Controle publica no tópico configurado para o sensor que realizou a coleta. Este processo continua se repetindo, obedecendo à configuração de tempo de coleta e publicação , como proposto no método FLOW.
  - **Vulnerabilidade**: segundo Sultan, Mehmood e Zahid (2022), o protocolo MQTT não garante a proteção da informação trafegada contra ataques do tipo *man-in-the-middle*;

  - **Solução**: a estratégia utilizada para mitigar esse problema foi a criptografia de Fernet. Sua implementação garante a confidencialidade dos dados, desde que a chave não seja comprometida. Como essa vulnerabilidade pode afetar todas as camadas, o dado permanece criptografado até o momento de entrega ao consumidor.

- **Camada de Suporte Operacional**: Nesta camada são implementados os CI e a Blockchain para garantir a imutabilidade, automação e rastreabilidade no processo de armazenamento das amostras. Sua atuação se inicia com o recebimento de uma mensagem contendo os dados das amostras coletadas. Cada vez que um pacote é recebido pelos *gateways*, o Contrato Inteligente 1 (CI1) é acionado para manipular as amostras criptografadas e realizar a montagem de uma nova transação. Com isso, o RFoT permite que o administrador determine quantas transações irão compor o dado a ser armazenado em um bloco da Blockchain de Dados (BCD).

  Uma vez que o pacote de transações é formado, o CI1 envia para o Contrato Inteligente 2 (CI2) realizar a mineração do bloco, resolvendo o desafio do *hash*, validando a consistência da BCD e atualizando a versão da Blockchain com o novo bloco. Uma vez que um bloco é minerado e adicionado na BCD, as amostras se tornam imutáveis, segundo os princípios da Blockchain.

  - **Vulnerabilidade 1**: o grande problema enfrentado nessa pesquisa é a alteração ou manipulação dos dados armazenados por uma IoT;
  - **Solução 1**: a solução para essa vulnerabilidade foi a integração da IoT com a Blockchain, por proporcionar imutabilidade e rastreabilidade;
  - **Vulnerabilidade 2**: outro problema que pode afetar a confiabilidade da fonte são falhas humanas de um especialista que pode participar diretamente em alguma das etapas;
  - **Solução 2**: a solução para essa vulnerabilidade foi a integração da IoT com CI, estabelecendo um formato operacional único e padronizado para os processos de coleta e armazenamento dos dados, automaticamente, sem a necessidade de um agente humano.

- **Camada de Suporte ao Negócio**: O objetivo dessa camada é garantir a disponibilidade da fonte, provendo uma interface entre a fonte de dados e os consumidores, recebendo requisições dos consumidores, fornecendo os dados armazenados na Blockchain e registrando os modelos gerados. Para cumprir essa tarefa, o RFoT conta com o Contrato Inteligente 3 (CI3), que automatiza o processo de recuperação de um bloco da BCD. Ao receber uma requisição do consumidor, o CI3 realiza um processo de recuperação de um bloco da Blockchain, que consiste no acionamento do algoritmo de consenso para obter a versão válida e mais atualizada da Blockchain, selecionar um bloco que ainda não foi enviado e devolvê-lo ao solicitante.

  Esse processo de coleta também faz parte do refinamento da imutabilidade proposta pela tecnologia Blockchain, pois cada vez que o algoritmo de consenso é acionado

para validar e sincronizar as versões da Blockchain. o RFoT anula os ataques de alteração de dados, garantindo que os dados originais sejam restaurados, desfazendo a corrupção da informação e o envio de amostras confiáveis ao consumidor.

Após o treinamento, o consumidor envia o modelo treinado para o CI2 realizar a mineração de um bloco da BCR. Vale lembrar que, nesta pesquisa, o consumidor é responsável por toda a informação trafegada fora do RFoT. Para melhor compreensão de como os dados são organizados na BCD e na BCR, são apresentados abaixo os parâmetros da classe que compõe um bloco de cada uma das Blockchains.

- ***index***: corresponde ao número bloco, seguindo uma sequência crescente, em que o primeiro bloco é o número um (***index*** = 1)
- ***proof***: corresponde ao número calculado pelo algoritmo *Proof Of Work*, que atende às condições estabelecidas para cálculo de *hash* do bloco. A lógica do processo de cálculo do *nounce* ou *proof* é apresentada no fluxograma 4.2.

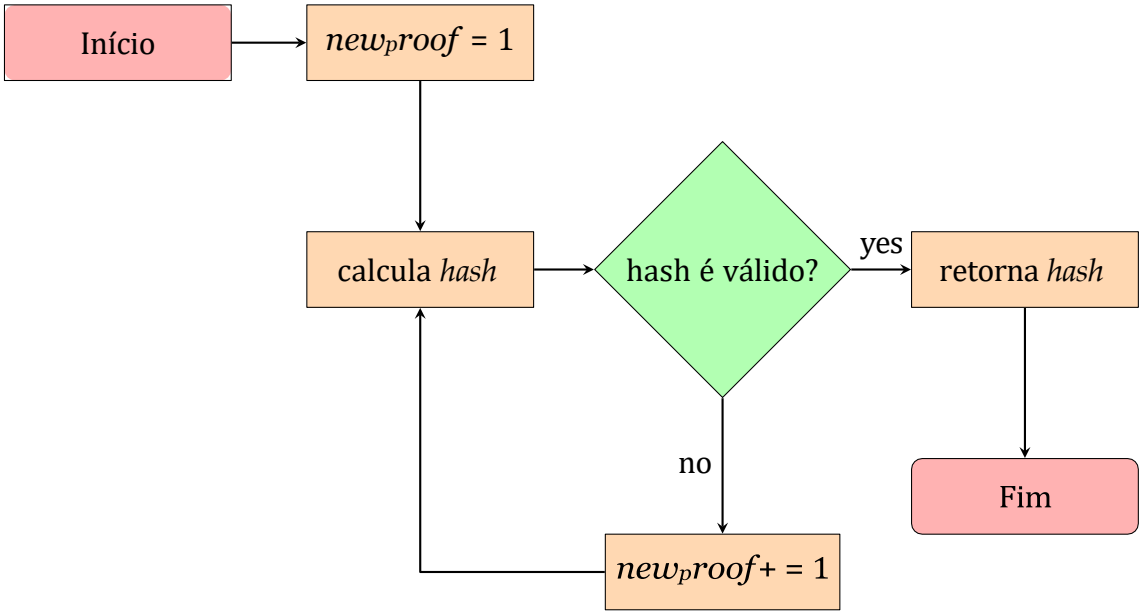



**Figura 4.2** Fluxograma do algoritmo para o cálculo do *Proof Of Work*
Fonte: Elaborado pelo autor.

Para gerar o *hash*, um valor numérico é calculado a partir da equação 4.1.

$$hv = newProof^2 - previewProof^2 \tag{4.1}$$

em que "$newProof$" é o *Proof of Work* (prova de trabalho) desejado, "$previewProof$" é o *Proof of Work* do bloco anterior e "$hv$" é o valor numérico resultante, convertido para texto.

Utiliza-se o algoritmo SHA256, criado pela *National Security Agency* (NSA) em 2001 N-able (2019), para obter o *hash* correspondente, representado por um número hexadecimal com 64 caracteres, que correspondem a $2^8$ bits.

Como apresentado no Capítulo 2, o cálculo do *Prof of Work*, nada mais é do que a solução de um desafio e nessa pesquisa, o desafio estabelecido consiste em obter um *hash* com os quatro primeiros caracteres iguais a zero.

- ***previousHash***: é o *hash* do bloco anterior
- ***timestamp***: registro da data e horário de mineração do bloco
- ***transactions***: conjunto de transações, que representam o dado armazenado na Blockchain. Uma transação, por sua vez, é constituída de:
  - ***sender***: quem está enviando o dado
  - ***receiver***: Quem está recebendo o dado
  - ***data***: dado enviado, que pode ser de dois tipos, um vetor com as amostras de coletadas pelos sensores ou o modelo treinado que representa o produto gerado pelo consumidor.
  - ***sensor*** : quando o tipo de dado se refere às medidas coletadas, o nome do dispositivo que coletou os dados é armazenado aqui.

Todos os parâmetros acima são comuns às duas Blockchains; no entanto, a BCR possui dois parâmetros adicionais: (1) *hashDataBlock*, que contém o *hash* do bloco da BCD para conectar à rastreabilidade já implementada na BCD; e (2) *hostTrainer*, que faz referência ao host consumidor da informação.

Com essa conexão entre as Blockchains, todos os elementos que interferem direta ou indiretamente no percurso do dado até o consumidor são rastreados, como mostra a Figura 4.3.

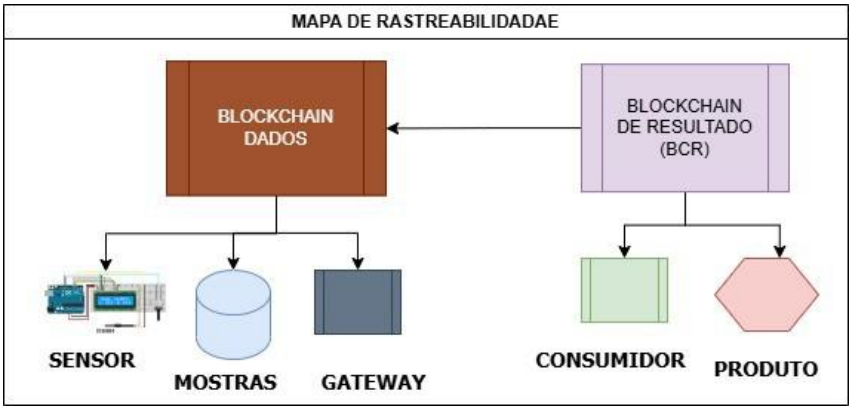



**Figura 4.3** Mapa de rastreio dos elementos que interferem no percurso do dado, desde a coleta, até a entrega a um consumidor

Fonte: Elaborado pelo autor.

- **Vulnerabilidade**: a vulnerabilidade identificada nessa camada é a mesma da camada de suporte, onde o fator humano poderia comprometer a confiabilidade no processo de comunicação com o consumidor

- **Solução**: essa vulnerabilidade é sanada pela integração com a Blockchain, que garante a imutabilidade dos dados, e CI, que além de automatizar o processo de coleta e armazenamento dos dados, torna desnecessária a atuação de atores corruptíveis.

Para entender a importância da BCR, imagine uma situação onde os sensores foram corrompidos ou simplesmente apresentaram mau funcionamento, publicando na rede valores suspeitos e esses dados sejam utilizados no treinamento dos modelos, gerando distúrbios no treinamento. Com a BCR, o consumidor poderá realizar uma análise para identificar a partir de que momento esse problema começou, a partir da data e hora de cada evento, identificando qual(is) sensor(es) está(ão) envolvidos e corrigindo o problema. A BCR também permite que o consumidor possa retomar o processo de treinamento a partir do último modelo que não recebeu dados distorcidos, sem perder o que já foi conquistado.

Outra situação seria uma tentativa de corrupção dos dados. Apesar de todo o ferramental proposto pelo RFoT para garantir a imutabilidade dos dados, não é possível afirmar que o sistema é invulnerável a fraudes. Portanto, o principal propósito da BCR é prover um meio de realizar uma auditoria sobre os modelos treinados para investigar comportamentos anômalos, a partir dos dados utilizados no seu treinamento.

Sem a BCR, não seria possível realizar uma verificação nesse nível, e caberia ao consumidor desenvolver soluções para isso. Como os dados influenciam diretamente no treinamento dos modelos, a implementação da BCR ajudará a guiar soluções para resolver a causa das anomalias, impedindo que todo um projeto seja encerrado por falta de confiabilidade.

Vale ressaltar que os processos de coleta e distribuição não são síncronos e são baseados em estímulos diferentes, pois a proposta do RFoT é atuar como uma fonte de dados e, no contexto proposto nessa pesquisa, sempre que um consumidor solicitar amostras para o treinamento, o RFoT terá a informação pronta para ser enviada. Com isso, é importante entender que o framework precisa ter sempre uma proporção ideal de dados para suprir a demanda dos consumidores.

### 4.3.1 Parametrização de Confiabilidade (PC)

Um dos desafios encontrados durante essa análise foi como representar e parametrizar a eficiência da solução proposta, uma vez que a literatura não apresenta parâmetros adequados para uma avaliação da confiabilidade dessa arquitetura.

Para superar esse desafio, foi desenvolvido um sistema de parametrização para possibilitar a metrificação do nível de confiabilidade que a arquitetura proporciona. Este sistema foi denominado Parametrização de Confiabilidade (PC) e foi utilizado para expressar a cobertura de confiabilidade proporcionada pelo RFoT e suas limitações.

O sistema de PC propõe analisar a confiabilidade da arquitetura camada a camada, seguindo o fluxo de passagem e controle sobre as amostras. Ele atesta a presença de estratégias para prover automação, disponibilidade, privacidade, rastreabilidade e imutabilidade da informação.

Assim, a nomenclatura de cada certificado inicia com o nome do *framework* e, para diferenciá-los, utilizou-se a letra "S" para certificados de segurança, que englobam automação, privacidade, disponibilidade e imutabilidade dos dados, e "R" para certificados de rastreabilidade/auditoria, conforme segue abaixo:

- **Certificado de Confiabilidade da Camada de Física (RFoT-S1)**: este certificado atesta a existência de estratégias para garantir a confiabilidade dos sensores, ou seja, estratégias que impedem a corrupção dos sensores, evitando a publicação de dados inválidos. Está relacionado à segurança do dispositivo de contato com a entidade física;
- **Certificado de Automação (RFoT-S2)**: A interferência do fator humano pode introduzir vulnerabilidades na arquitetura, pois mesmo especialistas podem cometer erros. Este certificado atesta que todo o processo, desde a coleta dos dados até a distribuição, ocorre de forma autônoma;
- **Certificado de Confiabilidade da Camada de Rede (RFoT-S3)**: atesta que a comunicação não está suscetível a ataques de alteração dos dados, pois toda a comunicação é criptografada;
- **Certificado de Imutabilidade (RFoT-S4)**: atesta que, uma vez que o dado chega à camada de suporte, ele não sofrerá alteração ou manipulação externa, ou seja, o dado é imutável;
- **Certificado de Disponibilidade (RFoT-S5)**: atesta apenas a segurança da distribuição dos dados requisitados, garantindo que o dado entregue é confiável;
- **Certificado de Rastreabilidade de Dados (RFoT-R1)**: este certificado atesta que o dado gerenciado pela IoT é auditável;
- **Certificado de Rastreabilidade de Modelos (RFoT-R2)**: atesta que o resultado produzido pelo consumidor é auditável desde a coleta dos dados;

Um dos principais benefícios do uso de certificados é a possibilidade de analisar a arquitetura por partes, tornando a análise escalável e especializada. Além disso, é possível combinar os certificados para compreender a cobertura de segurança da solução, conforme ilustrado na Figura 4.4.

Com base nesse entendimento, é possível assimilar os pontos de vulnerabilidade da arquitetura e trabalhar em estratégias para mitigar cada vulnerabilidade de uma camada específica, sem impactar negativamente as demais. Além disso, viabiliza a mensuração do nível de criticidade de cada projeto, explicitando as camadas que requerem mais investimento para atingir o objetivo de segurança desejado ou mesmo se o objetivo pode ser alcançado.

Ao analisar o RFoT, concluiu-se que podem ser atribuídos os certificados RFoT-S2, RFoT-S3, RFoT-S4, RFoT-S5, RFoT-R1 e RFoT-R2. Dessa forma, ele garante a segurança, privacidade, imutabilidade e rastreabilidade dos dados desde a publicação dos sensores até a entrega ao consumidor. Como já mencionado anteriormente, a segurança

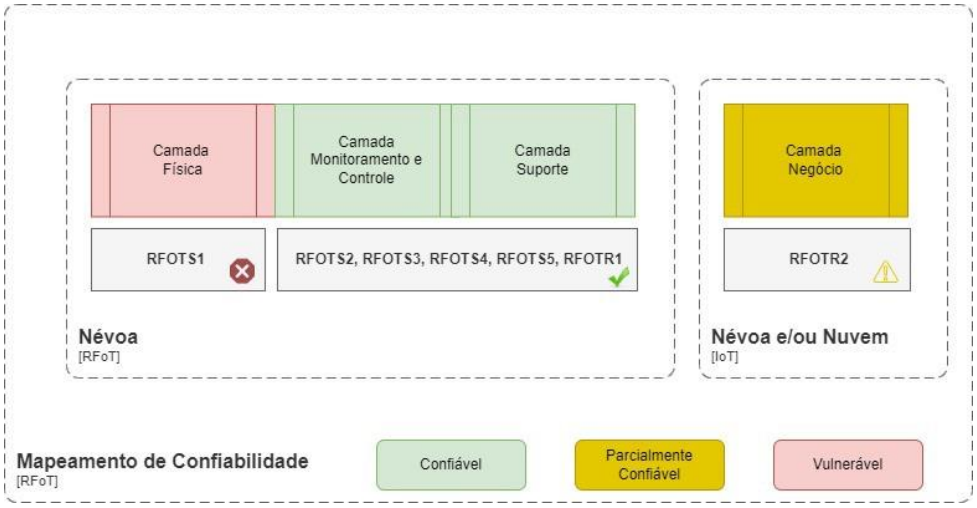



**Figura 4.4** Cobertura de Confiabilidade do RFoT
Fonte: Elaborado pelo autor

dos sensores e do processo do consumidor não faz parte do escopo desta pesquisa, pois já existem soluções para essas camadas, como o PUF apresentado por (PATIL et al., 2020).

Capítulo 5

# AVALIAÇÃO

Este capítulo apresenta os experimentos realizados para validar o problema da vulnerabilidade de uma IoT convencional e demonstrar como o RFoT oferece recursos para garantir a confiabilidade dos dados, tornando a IoT uma fonte de dados viável para o auxílio no treinamento de modelos de AM.

Dado que os dados coletados pela IoT são quantificáveis, utilizaram-se estatísticas para avaliar se a corrupção dos dados foi propagada para o consumidor e o impacto gerado nos resultados dos modelos treinados. A análise foi conduzida de forma comparativa entre as arquiteturas de uma IoT convencional e o RFoT.

Os experimentos propõem submeter ambas as arquiteturas às mesmas condições e avaliar os efeitos gerados no ator que consumiu os dados fornecidos por cada arquitetura. Assim, o experimento consistiu na simulação de quatro situações, onde as arquiteturas foram submetidas a um contexto com e sem tentativa de corrupção, seguido de uma análise comparativa das duas situações para cada arquitetura.

## 5.1 PLANEJAMENTO DE EXPERIMENTOS

O experimento desenvolvido nesta pesquisa consiste em um estudo de caso em que a IoT atua como uma fonte de dados, responsável por coletar informações de entidades físicas por meio de sensores de temperatura e umidade, para disponibilizá-las a um ator consumidor. O papel do consumidor é solicitar dados à IoT e gerar um produto que possa ser avaliado e registrado em formato de texto.

A análise da capacidade de garantia de confiabilidade é realizada com base nos dados recebidos pelo consumidor, permitindo compreender se ele recebe ou não dados corrompidos da fonte de dados e o efeito nos resultados do treinamento dos modelos.

Nos experimentos apresentados, o sistema consumidor produz modelos capazes de classificar o conforto térmico do espaço interno de um laboratório, a partir de dados de temperatura e umidade. O conforto térmico tem se tornado uma informação importante no planejamento urbano e, por isso, tem ganhado destaque no âmbito científico.

Atualmente, três índices têm sido utilizados como referência, cada um apresentando uma característica singular. Nesta pesquisa, a análise do conforto térmico foi realizada

utilizando o Intel Lab, e o índice IDT foi escolhido por apresentar uma melhor adequação aos dados. Além disso, foi feita uma adaptação da tabela de referência para simplificar a classificação do conforto térmico.

O IDT considera quatro faixas de conforto térmico, conforme apresentado na Tabela 5.1. No entanto, utilizou-se uma classificação binária para o treinamento dos modelos, visando classificar apenas os estados "Confortável" e "Desconfortável", como apresentado na Tabela 5.2.

**Tabela 5.1** Tabela do índice IDT para inferência do conforto térmico
Fonte: Elaborado pelo autor

| Faixas | IDT(°C) | Níveis de Desconforto Térmico |
|---|---|---|
| 1 | IDT < 24,0 | Confortável |
| 2 | $24 \leq IDT \leq 26$ | Parcialmente confortável |
| 3 | $26 < IDT < 28$ | Desconfortável |
| 4 | $IDT \geq 28$ | Desconfortável |

**Tabela 5.2** Tabela IDT adaptada para apenas duas faixas de conforto térmico.
Fonte: Elaborado pelo autor

| Faixas | IDT(°C) | Níveis de Desconforto Térmico |
|---|---|---|
| 1 | $IDT \leq 26,0$ | Confortável |
| 2 | $IDT > 26$ | Desconfortável |

### 5.1.1 Configuração do Gêmeo Digital

O GD representa o ambiente de simulação desta pesquisa, permitindo a construção de toda a topologia de rede que atuará como a Edge de uma IoT. Isso significa que, dentro do GD, será simulado o processo de coleta de dados por sensores, o tratamento desses dados pelos *gateways* da camada de suporte e o fornecimento desses dados para a camada de negócio, onde os consumidores estarão solicitando dados para o treinamento de modelos.

Como ambiente virtualizado, o GD é essencialmente um conjunto de *softwares* que oferecem recursos para a execução das atividades. Por questões de simplificação e custo-benefício, utilizou-se uma máquina virtual criada com o software VMWare, em um computador com processador Intel Core i5 10400F 2.90GHz, 16 GB de memória e sistema operacional Windows 10.

A máquina virtual foi criada a partir da imagem da distribuição Linux Xubuntu, por oferecer o mínimo de recursos de uma distribuição Linux, mas suficiente para a instalação e execução do MININET e das bibliotecas Python necessárias para encriptar dados, criar uma Blockchain integrável com a IoT, criar contratos inteligentes e realizar o treinamento de modelos de AM. Abaixo seguem as configurações mínimas da máquina virtual para reprodução dos experimentos:

- Memória: 4 GB

- Número de processadores: 2
- Espaço do disco rígido: 80 GB

Como apresentado no Capítulo 2, o MININET é o software que possibilita a criação e execução da topologia de rede e, como foi desenvolvido em Python, todo o código do RFoT também foi desenvolvido na mesma linguagem. Portanto, todo o código do RFoT pode ser encontrado no repositório público (SILVA; BRENNO, 2024). O *readme* apresenta um passo a passo para a instalação e configuração de algumas ferramentas e bibliotecas necessárias.

### 5.1.2 Roteiro dos experimentos

As etapas dos experimentos são:

- Coleta e armazenamento das amostras
- Acionamento do sistema consumidor
- Pré-processamento dos dados
- Execução do treinamento dos modelos
- Análise dos resultados

Para otimizar essa análise, foi projetado um consumidor composto por cinco dispositivos FoT, responsáveis pela solicitação de dados e treinamento dos modelos, um sexto dispositivo para integrar os modelos gerados e um sétimo dispositivo para monitorar os resultados dos treinamentos em tempo real, baseado no algoritmo AF.

Os cenários apresentados a seguir foram concebidos para simular todo o contexto da atuação da IoT como uma fonte de dados, desde a coleta até o treinamento dos modelos. Como a proposta desta pesquisa é apresentar o RFoT como solução para garantir a confiabilidade dos dados em uma rede IoT, foi desenvolvido um experimento composto pelos quatro cenários descritos abaixo:

- Cenário 1: Simulação do treinamento de um modelo a partir de dados de uma rede IoT convencional que não sofreu tentativa de corrupção;
- Cenário 2: Simulação do treinamento de um modelo a partir de dados de uma rede IoT convencional que sofreu tentativa de corrupção;
- Cenário 3: Simulação do treinamento de um modelo a partir de dados do RFoT sem tentativa de corrupção;
- Cenário 4: Simulação do treinamento de um modelo a partir de dados do RFoT com tentativa de corrupção;

A ideia central do experimento é validar se o RFoT e uma IoT convencional são capazes de impedir tentativas de corrupção e não propagar dados manipulados quando submetidos a um ataque. A validação dessa habilidade foi realizada com base nos dados recebidos pelo consumidor e nos resultados do treinamento dos modelos.

Se o consumidor receber dados corrompidos, isso indica que a arquitetura não foi capaz de garantir a confiabilidade dos dados. Outro fator a considerar é como essa corrupção impactou o resultado do treinamento dos modelos, a fim de explicitar os riscos e impactos do uso de uma fonte de dados sem confiabilidade.

Desse modo, as seguintes análises são propostas para avaliar o comportamento de cada arquitetura e sua capacidade de garantir a confiabilidade dos dados:

- **Análise IoT convencional**: Comparar os resultados do treinamento dos cenários (1) e (2) para entender se o consumidor recebeu dados diferentes nas duas situações e se o resultado do treinamento variou;

- **Análise RFoT**: Comparar os resultados do treinamento dos cenários (3) e (4) para validar a capacidade do RFoT em garantir que o consumidor receberá um conjunto de dados sem corrupção em ambas as situações, obtendo resultados similares no treinamento;

- **Análise de Custo Computacional**: Avaliar o tamanho das mensagens trocadas pelos dispositivos desde a coleta;

- **Análise de Custo de Arquitetura**: Apresentar exemplos de dispositivos que podem compor cada camada, seus recursos e custos de aquisição;

- **Análise de Impacto Temporal da Integração das Tecnologias**: Analisar o impacto temporal ao integrar a IoT com a Blockchain e CI.

Para compreender como os experimentos serão realizados, é necessário conhecer um pouco da arquitetura do ambiente de simulação. Assim, as subseções seguintes apresentam um esboço da composição do GD desenvolvido e os processos executados a partir dele.

### 5.1.3 Coleta e Armazenamento das Amostras

Nesta etapa, cada sensor executa um subprocesso independente e, utilizando o método FLOW do protocolo TATU, os dados são coletados do *dataset* Intel Lab, criptografados e publicados no tópico em intervalos periódicos de 10 segundos. Dessa forma, os dispositivos FoT inscritos no tópico recebem as amostras e as armazenam conforme o padrão estabelecido para cada arquitetura.

Nesta pesquisa, as amostras são armazenadas em blocos, compostos por transações que seguem o formato apresentado no Capítulo 4. O tamanho do bloco pode ser definido durante a execução do processo de coleta, permitindo ajustes conforme o contexto de treinamento.

Nos experimentos desenvolvidos nesta pesquisa, o tamanho do bloco foi definido para 20 transações, sendo que cada transação contém uma medida de temperatura e uma de umidade. Assim, após o bloco alcançar a quantidade de transações estabelecida, ele é armazenado em um arquivo no formato JSON. Além disso, durante todo o tráfego, as amostras permanecem criptografadas para garantir a privacidade da informação, minimizando a possibilidade de ataques do tipo *man-in-the-middle*.

### 5.1.4 Acionamento do Sistema Consumidor

Nesta pesquisa, um consumidor é definido como um dispositivo da rede que solicita dados à IoT e os utiliza para realizar o treinamento de um modelo de AM. Este modelo pode ser representado em formato de texto, permitindo seu registro na BCR e possibilitando sua rastreabilidade a partir dos dados recebidos. Para isso, foi escolhido um sistema de AM descentralizado, baseado no algoritmo AF, para treinar modelos capazes de classificar a sensação térmica de um ambiente de escritório.

Os modelos gerados visam treinar uma AI capaz de classificar o conforto térmico, com base nas informações de temperatura e umidade, utilizando o índice IDT como base de classificação. A escolha por esse tipo de sistema se deve ao fato de que o desempenho e a confiabilidade de uma AI dependem de dados confiáveis e de qualidade. Essas características, aliadas ao baixo custo de implementação e reprodução, favorecem a análise da confiabilidade proposta para o RFoT, permitindo validar cada camada da arquitetura da solução.

É importante destacar que o sistema consumidor escolhido nesta pesquisa não tem o propósito de resolver um problema real, mas sim de avaliar a solução de confiabilidade na borda da rede IoT. Assim, uma vez que o dado é recebido pelo sistema consumidor, não são realizadas medidas adicionais de segurança, pois o escopo de confiabilidade se aplica apenas à fonte de dados. O máximo que o RFoT consegue garantir nesse contexto é a rastreabilidade do que é gerado pelo consumidor. Este é um ponto a ser abordado na seção de trabalhos futuros (ver Seção 6.3).

### 5.1.5 Pré-processamento dos Dados

O cálculo proposto pelo índice IDT utiliza amostras de temperatura e umidade para determinar o conforto térmico. O Intel Lab oferece um conjunto de 1.032.198 amostras. No entanto, foram utilizadas apenas 20.000 amostras, uma vez que a tentativa de utilizar um conjunto maior resultou em um evento de estouro de memória, devido a limitações na arquitetura de memória do GD.

Como o tamanho do bloco foi definido para 20 transações, essas 20.000 amostras foram organizadas em um conjunto de 1.000 blocos. Assim, cada vez que um consumidor realiza uma requisição à IoT, um bloco é enviado e, antes de realizar o treinamento com os dados do bloco, este passa por um processo de pré-processamento. Esse procedimento ocorre sempre que um consumidor recebe os dados requisitados.

O pré-processamento dos dados consiste em um conjunto de ações para adequar os dados ao contexto do treinamento, eliminando ou reduzindo ruídos que interfiram na identificação de padrões. Esses ruídos podem incluir valores nulos ou valores em escalas

que se distanciam significativamente da maioria das amostras. Valores que se enquadram nessa descrição são conhecidos como *outliers*.

A análise de *outliers* foi realizada sobre as primeiras 20.000 amostras do *dataset* Intel Lab, conforme explicado na seção anterior. Como o treinamento dos modelos ocorre em ciclos e a cada ciclo um pequeno subconjunto de dados é utilizado, seria difícil visualizar a base de dados completa usando apenas um bloco. Por isso, o pré-processamento apresentado considera todo o conjunto de amostras.

O pré-processamento concentrou-se principalmente no tratamento dos *outliers*, começando com a remoção de valores nulos. A partir desse novo conjunto, foi realizada uma análise gráfica da distribuição das amostras, apresentada na Figura 5.1.

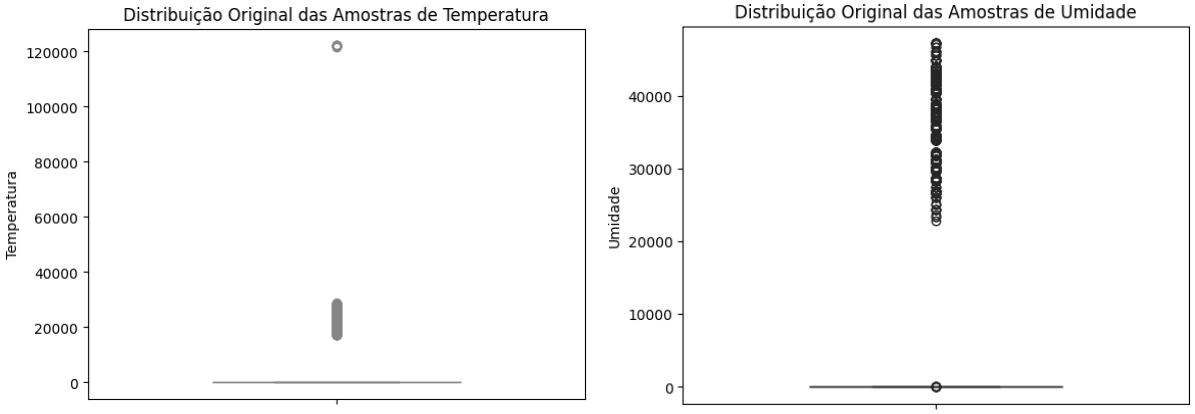



**Figura 5.1** Gráfico de distribuição da temperatura e umidade originais
Fonte: Elaborado pelo autor.

Ao observar a Figura 5.1, percebe-se a existência de alguns valores que se distanciam significativamente da maioria das amostras, resultando no deslocamento da média e do desvio padrão, dificultando a identificação de padrões pelo algoritmo de treinamento. Isso se torna ainda mais evidente ao analisar os dados apresentados na Tabela 5.3 e a frequência das faixas de amostras na Figura 5.2.

| Índice | Temperatura | Umidade |
|---|---|---|
| Quantidade | 20000,0 | 20000,0 |
| Média | 3715,9363 | 3413,1208 |
| Desvio Padrão | 8574,8014 | 10761,5580 |
| Mínimo | 17,1954 | 3, 9190 |
| 25% | 20,3412 | 33,8396 |
| 50% | 22,3600 | 3,4629 |
| 75% | 25,3686 | 41,5116 |
| Máximo | 122153,0 | 47212,0 |

**Tabela 5.3** Tabela de dados de temperatura e umidade

Embora os *outliers* geralmente representem valores problemáticos, nem sempre a melhor estratégia é removê-los. Nesta pesquisa, eles são importantes para possibilitar o

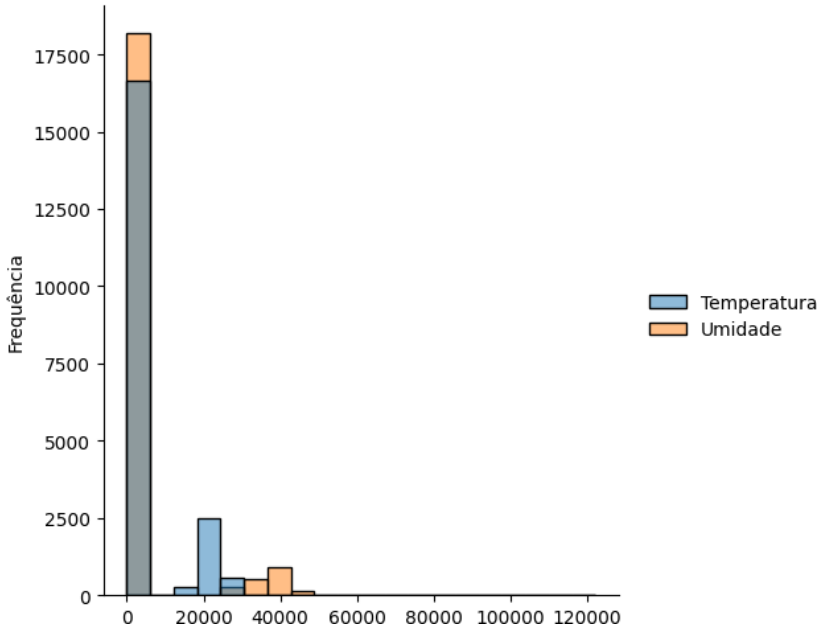



**Figura 5.2** Mapa de rastreio dos elementos que interferem no percurso do dado, desde a coleta até a entrega a um consumidor

Fonte: Elaborado pelo autor.

treinamento dos modelos para a classificação da classe denominada como "desconfortável". Assim, na tentativa de preservar o máximo de amostras possíveis com qualidade, percebeu-se que os *outliers* apresentavam valores similares aos demais nos dois primeiros dígitos, sugerindo um erro no posicionamento do separador decimal durante o registro.

Portanto, em vez de removê-los, foi criada uma função para reposicionar o separador, ajustando-o para manter o padrão de dois dígitos inteiros. No entanto, ao analisar a fórmula do índice IDT, desconsiderando o valor da umidade, seria necessário um valor de temperatura em torno de 40°C para gerar um contexto de desconforto térmico, representado pela classe "0". Assim, após o reposicionamento do separador decimal, esses valores foram acrescidos de 20 unidades para garantir que os modelos tivessem dados para aprender a classificar ambos os contextos.

Dessa forma, os *outliers* se tornaram elementos importantes na composição do conjunto de dados utilizados no treinamento dos modelos. Como resultado, a média ficou próxima do 2° quartil, o desvio padrão da temperatura foi ajustado para aproximadamente $7,9$ e o da umidade para $7,6$, com a variação de temperatura entre $17,2$°C e $48,7$°C e a umidade entre $-3,9$ e $67,2$, conforme mostrado na Tabela 5.4 e na Figura 5.3.

A Figura 5.4 apresenta os histogramas da temperatura e umidade, respectivamente, com a distribuição das amostras e suas respectivas frequências, mostrando uma maior concentração nas temperaturas entre 24°C e 26°C.

Ao realizar o cálculo do IDT sobre os dados pré-processados, obteve-se a estrutura de dados utilizada para o treinamento dos modelos, conforme apresentado na Figura 5.5.

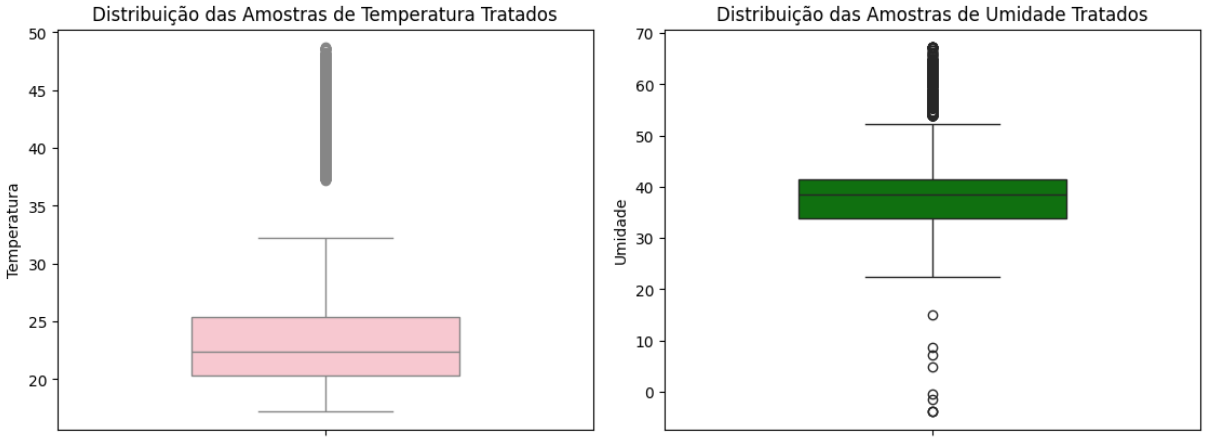



**Figura 5.3** Gráfico de distribuição da temperatura e umidade após ajustes do posicionamento do separador decimal

Fonte: Elaborado pelo autor.

| Índice | Temperatura | Umidade |
|---|---|---|
| Quantidade | 20000,0 | 20000,0 |
| Média | 25,2203 | 38,7188 |
| Desvio Padrão | 7,8574 | 7,6277 |
| Mínimo | 17,1954 | -3,9190 |
| 25% | 20,3412 | 33,8396 |
| 50% | 22,3600 | 38,4629 |
| 75% | 25,3686 | 41,5116 |
| Máximo | 48,6810 | 67,2120 |

**Tabela 5.4** Tabela de dados de temperatura e umidade após pré-processamento

### 5.1.6 Execução do Treinamento dos Modelos

O treinamento dos modelos seguiu o fluxo operacional do Aprendizado Federado (do inglês, *Federated Learning*), que divide o treinamento em três etapas, formando um ciclo de treinamento incremental.

- **Seleção do modelo:** O primeiro passo é a seleção de um modelo, chamado de "modelo primário", que será distribuído entre os dispositivos para iniciar o processo de treinamento do algoritmo AF;
- **Treinamento do modelo local:** Consiste no treinamento do modelo compartilhado, a partir de dados locais de cada dispositivo, individualmente;
- **Agregação do Modelo:** Uma vez que os modelos locais individuais são treinados, eles são enviados a um servidor central, onde ocorre o treinamento de um modelo global, que representa uma atualização do modelo anterior. Uma vez gerado o modelo global, este é compartilhado com os nós, reiniciando todo o processo, mas a partir do novo modelo.

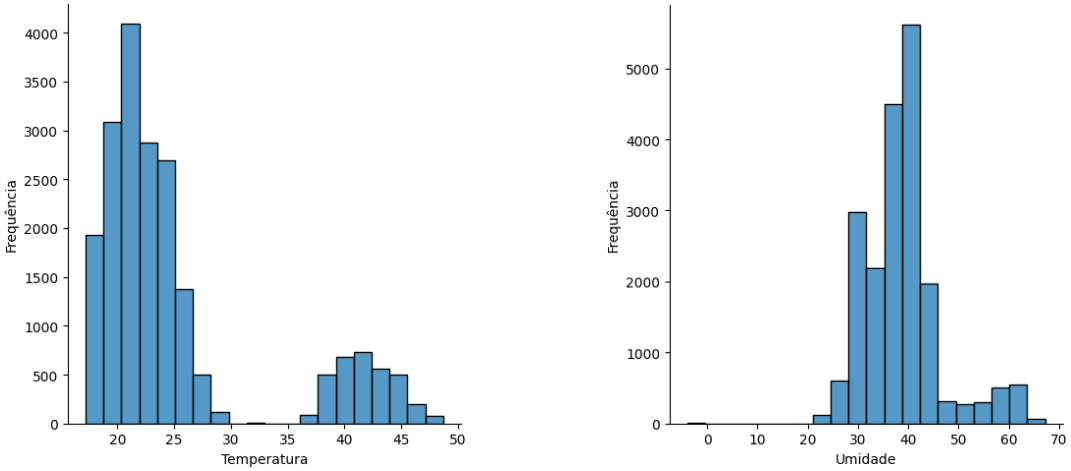



**Figura 5.4** Histograma das amostras de temperatura e umidade pós-tratamento de *outliers*
Fonte: Elaborado pelo autor.

Essa característica de aprendizado incremental utilizando vários nós simultâneos para o treinamento de um modelo de AM possibilita representar em tempo real tentativas de corrupção nos dados armazenados pela IoT e o impacto gerado nos resultados do consumidor. Assim, os cenários 2 e 4 apresentam tentativas de corrupção dos dados fornecidos a cada ciclo e com o auxílio do recurso de definição do nível Qualidade de serviço (do inglês *Quality of Service, QoS* (QoS) provido pelo MQTT, foi possível tornar a arquitetura do consumidor preparada para lidar com uma de desconexão de um cliente. Uma vez que o nó integrado só consegue prosseguir se todos os clientes previstos enviarem o modelo que treinaram, se um cliente não receber o modelo global por algum motivo, todo o sistema ficaria parado, aguardando a entrega desse cliente.

Pensando nisso, o nível de QoS do protocolo MQTT foi definido para o nível 2, que possibilita a um nó receber uma mensagem, mesmo que se conecte à rede após o envio. Isso garante a um cliente receber o modelo global, mesmo que seja conectado à rede após a publicação no tópico. Além disso, o nó integrador foi configurado para aguardar o envio dos modelos por um período determinado, seguindo com a integração, mesmo que não receba todos os modelos esperados. Outro motivo para a escolha desse nível de QoS é que apenas uma mensagem pode ser postada por vez no tópico. Desse modo, se dois ou mais clientes publicarem ao mesmo tempo, uma das mensagens seria perdida.

Cada nó cliente da arquitetura do AF realiza o treinamento dos modelos AM, utilizando uma rede neural do tipo Multilayer Perceptron (Perceptron de múltiplas camadas), implementada com auxílio da biblioteca Keras e TensorFlow versão 2.2, que oferece um conjunto de classes para o desenvolvimento do treinamento, além de ferramentas de monitoramento.

A arquitetura da rede neural foi definida com doze neurônios na camada de entrada, trinta e dois neurônios na camada oculta, com função de ativação *ReLU*, e um neurônio na camada de saída, com função de ativação *sigmoid*, como apresentado na Figura 5.6.

A função para o cálculo da taxa de perdas (*loss*) escolhida foi a "*BinaryCrossentropy*",

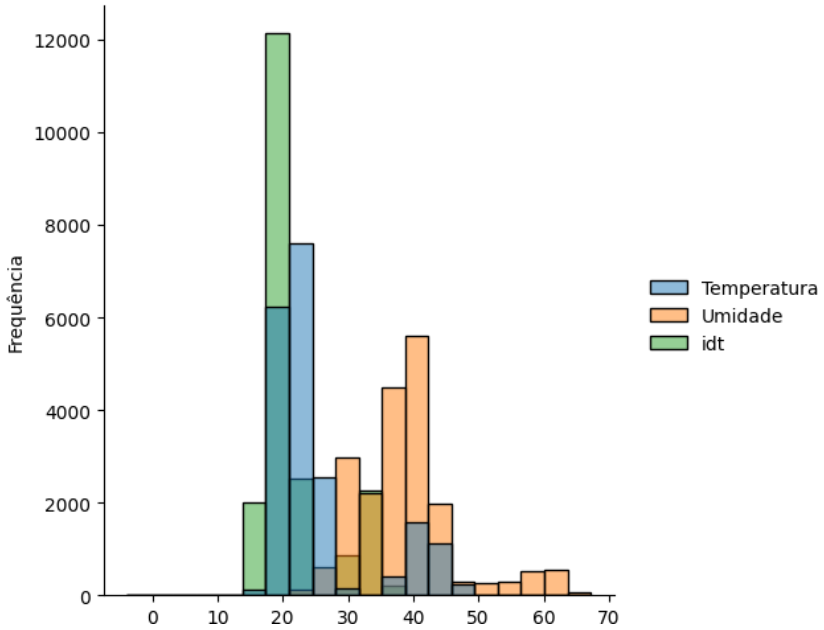



**Figura 5.5** Histograma da distribuição dos dados utilizados para o treinamento dos modelos
Fonte: Elaborado pelo autor.

por ser mais adequada para classificações binárias, ou seja, que apresentam apenas dois estados (0 ou 1). Assim, a taxa de aprendizagem foi definida para $0,01$, com taxa de decaimento calculada pela equação 5.1.

$$I_d = \frac{I_l}{bachSize} \tag{5.1}$$

A cada iteração, os nós clientes publicam o modelo atualizado em um tópico ao qual apenas o nó integrador está inscrito. O nó integrador recebe e armazena os modelos à medida que são publicados. Assim que todos os nós publicam seus modelos, o nó integrador realiza o processo de combinação dos modelos recebidos.

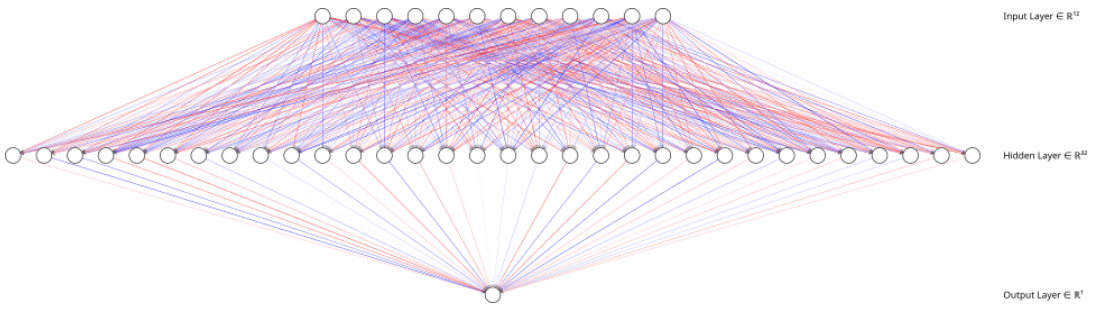



**Figura 5.6** Gráfico da arquitetura da Rede Neural implementada para geração dos modelos de Aprendizado de Máquina
Fonte: Elaborado pelo autor.

Essa combinação consiste na aplicação de cálculos matemáticos sobre o conjunto de pesos de cada modelo, com o objetivo de obter um conjunto de pesos que represente o melhor resultado, integrando todos os modelos envolvidos.

O primeiro passo para realizar a integração é o cálculo do fator de escala, conforme apresentado nas Equações 5.2, 5.3 e 5.4.

$$scalingFactor = \frac{individualFactor}{globalFactor} \tag{5.2}$$

O fator de escala, "*scalingFactor*", é determinado pelo quociente entre o fator individual (ver Equação 5.3) de um cliente e o fator global, "*globalFactor*" (ver Equação 5.4).

$$individualFactor = n \cdot y \tag{5.3}$$

O fator individual de um cliente, "*individualFactor*", é calculado pelo produto da cardinalidade, "*y*" (quantidade de pontos utilizados no treinamento), pela quantidade total de clientes, *n*.

$$globalFactor = n \sum_{i=1}^{n} x_i \tag{5.4}$$

O fator global, "*globalFactor*", é o somatório da cardinalidade de todos os clientes, "*i*", multiplicado pela quantidade total de clientes, "*n*".

O vetor de pesos de cada cliente é multiplicado pelo fator de escala. O vetor resultante é utilizado para compor uma matriz de pesos, $C_{ij}$, onde "*i*" representa a quantidade de clientes e "*j*" o tamanho do vetor de pesos dos clientes, conforme mostrado na Equação 5.5.

$$M_i = C_i \cdot scalingFactor \tag{5.5}$$

Onde $M_i$ representa uma linha da matriz de pesos e $C_i$ o conjunto ou vetor de pesos de um cliente. Assim, os pesos resultantes do modelo global são obtidos pelo somatório das linhas da matriz transposta da matriz de pesos. Veja a Equação 5.6.

$$P_i = \sum_{j=1}^{b} M_{ij}^{T} \tag{5.6}$$

Onde $P_i$ é um elemento do vetor de pesos resultante do modelo global, e $M^T$ é a matriz transposta da matriz de pesos. O somatório é calculado para cada linha da matriz transposta, resultando no vetor de pesos do modelo global.

Uma vez que o vetor de pesos é ajustado, ele é utilizado para atualizar o modelo global, que é novamente publicado e propagado para os clientes, reiniciando o processo.

### 5.1.7 Análise dos Resultados de Treinamento

A cada ciclo completado, um novo modelo global é gerado e propagado. Para monitorar esse processo, foi desenvolvido um nó de monitoramento do treinamento, que calcula o nível de eficiência do modelo global, registrando os resultados no arquivo "resultado.csv".

O papel do nó de monitoramento é entender a capacidade de classificação do modelo, a fim de avaliar se dados corrompidos podem gerar alterações da capacidade de classificação. Para isso, ele sorteia um subconjunto do *dataset* e realiza o teste de predição.

A partir dos resultados armazenados, é possível utilizar métricas de classificação e erros para analisar o desempenho do modelo. Assim, é possível validar a confiabilidade da fonte de dados baseado nos dados que foram entregues ao consumidor e na capacidade de classificação dos modelos.

Essa análise cabe tanto para uma manipulação de dados que vão prover um melhor desempenho ou pior desempenho, pois a base é se houve um fastamento dos padrões proposto no cenário que representa uma *baseline* e se o modelo está recebendo dados incomuns.

- **Taxa de perda (*Loss*)**: para o cálculo da taxa de perda, é utilizada a função "*binaryCrossEntropy*", que calcula a probabilidade de ser "**um**" para **N** pontos, sendo "**um**" o estado de acontecimento do evento Godoy (2018). A Equação 5.7 apresenta o modelo matemático dessa função:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i) \tag{5.7}$$

  Em que $y$ é o valor real e $\hat{y}$ é o valor predito;

- **Matriz de Confusão**: proporciona informações do desempenho do modelo, apresentando um cruzamento da quantidade de acertos e erros do processo de predição em comparação aos resultados reais esperados. É geralmente utilizada em treinamentos supervisionados. As combinações geradas são Verdadeiros Positivos (TP), Falsos Positivos (FP), Verdadeiros Negativos (TN) e Falsos Negativos (FN). A partir das informações proporcionadas por ela, é possível extrair métricas como a Acurácia, Precisão, *Recall* e *F1-Score* (VISA et al., 2011);

- **Acurácia**: a acurácia é a frequência de acertos na predição, ou seja, quantas vezes o valor predito foi igual ao real. A acurácia pode ser calculada através da Equação 5.8 Chicco, Tötsch e Jurman (2021):

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.8}$$

- **Precisão**: a precisão é a razão entre as instâncias positivas corretamente previstas e o total de instâncias previstas como positivas. Ela mede a capacidade do modelo de evitar rotular instâncias negativas como positivas, apresentada na Equação 5.9 (THARWAT, 2021):

$$precision = \frac{TP}{TP + FP} \tag{5.9}$$

- ***Recall***: também conhecida como sensibilidade ou taxa de verdadeiros positivos, é a razão entre as instâncias positivas corretamente previstas e todas as instâncias realmente positivas. Ela mede a capacidade do modelo de encontrar todas as instâncias positivas. Ver Equação 5.10 (POWERS, 2020):

$$recall = \frac{TP}{TP + FN} \tag{5.10}$$

- ***F1-Score***: é a média harmônica entre precisão e revocação. Ele fornece uma única pontuação que equilibra tanto a precisão quanto a revocação, sendo especialmente útil quando você tem uma distribuição desigual de classes, como apresentado na Equação 5.11 (GOUTTE; GAUSSIER, 2005):

$$F1\text{-}Score = \frac{2 \cdot (precision \cdot recall)}{precision + recall} \tag{5.11}$$

$$TVP = \frac{TP}{TP + FN} \tag{5.12}$$

$$TFP = \frac{FP}{FP + TN} \tag{5.13}$$

- **Mediana do erro absoluto (MedAE)**: representa a mediana da diferença entre o valor predito e o esperado, para todas as predições, como mostrado na Equação 5.14. Essa métrica foi escolhida por ser robusta em relação aos *outliers*, ou seja, valores que fogem do padrão geral (ABADI et al., 2022):

$$MedAE(y, \hat{y}) = median(|y_1 - \hat{y}_1|, ..., |y_n - \hat{y}_n|) \tag{5.14}$$

- **RMSE**: é uma métrica que representa o risco correspondente ao valor esperado do erro ou perda quadrática (CHAI; DRAXLER, 2014), apresentado na Equação 5.15:

$$MSE(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}(y - \hat{y})^2 \tag{5.15}$$

- **MAE**: é o valor absoluto da média do erro ou taxa de perda, como apresentado na Equação 5.16 (WILLMOTT; MATSUURA, 2005):

$$MAE(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}|y - \hat{y}| \tag{5.16}$$

## 5.2 EXECUÇÃO DE EXPERIMENTOS

Nesta seção, apresentamos a execução de cada cenário, detalhando o passo a passo do roteiro e os resultados obtidos na validação da confiabilidade que cada arquitetura proporciona.

Para diferenciar quando os dados recebidos pelo consumidor foram ou não manipulados, cada arquitetura é analisada em dois cenários: sem ataque e com ataque. A Figura 5.7 ilustra as duas arquiteturas, destacando que o RFoT se diferencia por integrar a tecnologia Blockchain e CI na camada de suporte, garantindo a privacidade e a imutabilidade dos dados.

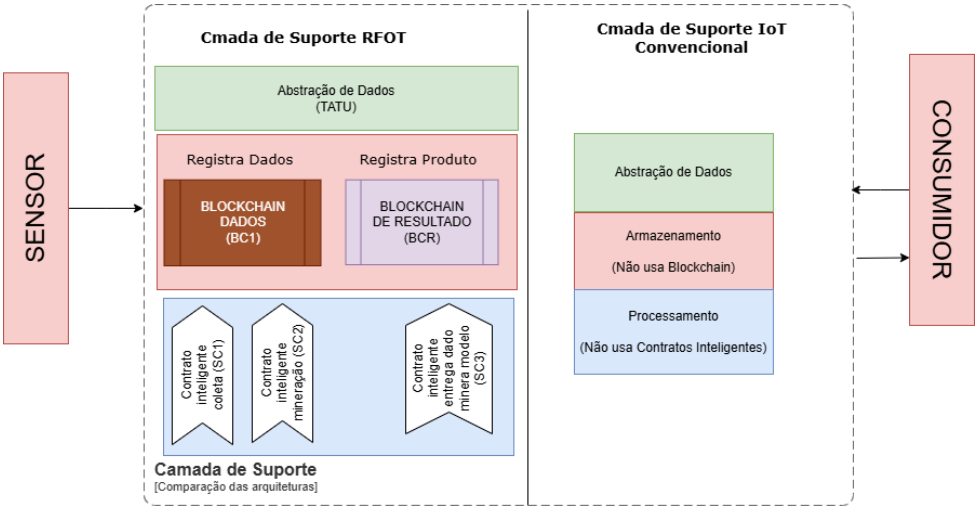



**Figura 5.7** Comparação entre a arquitetura de uma IoT convencional e do RFoT
Fonte: Elaborado pelo autor.

Para simular um cenário de ataque, desenvolvemos um algoritmo que representa um nó corrompido. Este nó é responsável por alterar os dados armazenados, sorteando valores inteiros entre um e mil para modificar as amostras já registradas. Optamos por utilizar um sistema de armazenamento em arquivo de texto no formato JSON, visando simplificar a implementação do código.

No contexto do RFoT, a privacidade dos dados é mantida ao impedir que os nós da rede tenham acesso direto à chave de criptografia. Em vez disso, eles devem solicitar a tradução da informação a outros módulos. Para acomodar essa arquitetura, implementamos um método nesses módulos que permite ao nó corrompido solicitar alterações sem acessar a informação original.

Dessa forma, os experimentos destacam as possíveis consequências a partir desse ponto, considerando que seria extremamente complexo simular um cenário em que um atacante invade o sistema e realiza alterações diretamente.

Após o treinamento, cada modelo foi registrado em um arquivo JSON. Dado que os resultados dos experimentos geraram dados extensos e complexos para serem expressos durante a execução dos cenários, criamos um *Dashboard* em formato de aplicação Web, utilizando a biblioteca "*Streamlit*".

Inicialmente, tentamos utilizar a biblioteca “Dash”, devido à sua integração com o Bootstrap 5, que oferece melhores recursos de estilização. No entanto, ela não proporcionou uma forma simples de expressar os dados em formato JSON como a “*Streamlit*”.

Por meio do *Dashboard*, foi possível apresentar os dados coletados em cada cenário, assim como os dados recebidos pelo consumidor e os resultados dos treinamentos. Em alguns casos, as informações foram dispostas lado a lado para facilitar a análise comparativa das respostas, como pode ser visto na opção de tempos de execução.


**Figura 5.8** *Dashboard* desenvolvido para análise e apresentação de resultados
Fonte: Elaborado pelo autor.

### 5.2.1 Cenário de experimentos 1

Neste primeiro cenário, simulamos o fluxo operacional de uma IoT que atua como uma fonte de dados íntegra, sem qualquer tipo de corrupção. O objetivo principal deste cenário é estabelecer uma *baseline*, apresentando um contexto ideal de operação da IoT, que, ao ser consumido, gera modelos com bom desempenho na classificação do desconforto térmico.

Os dados discutidos nesta subseção são provenientes do treinamento de um modelo global, utilizando informações fornecidas pela arquitetura de uma IoT convencional. Todos os dados recebidos da fonte são armazenados em um conjunto de dados, permitindo a comparação entre o que se espera receber e o que foi efetivamente recebido pelo consumidor.

Ao analisar a Figura 5.9, observa-se que os dados utilizados no treinamento dos modelos mantêm o mesmo padrão da base original, confirmando que o consumidor recebeu os dados sem corrupção. Além disso, os valores do índice IDT calculados também estão conforme a base de dados utilizada.

Como resultado, o modelo apresentou uma acurácia e precisão de 92%, indicando que ele acertou 92% das tentativas de predição. A métrica de classificação *recall* foi de 100%, pois nenhum falso negativo foi gerado na predição.

Isso ocorre porque todos os valores preditos ficaram acima do *trade-off*, garantindo

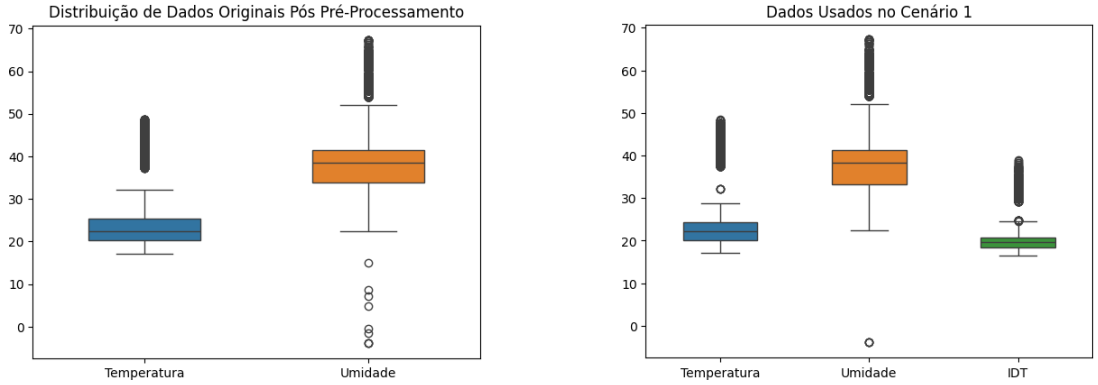



**Figura 5.9** Comparação dos dados originais e recebidos no cenário 1
Fonte: Elaborado pelo autor.

que todas as possibilidades existentes no conjunto predito fossem alcançadas. A análise do *F1-Score* ajuda a compreender a combinação desses resultados, conforme mostrado nas Figuras 5.10 e 5.12.

Conforme apresentado na Figura 5.11, todas as taxas de erro ficaram abaixo de 40%, contribuindo para um nível de classificação aceitável para esta pesquisa, uma vez que o foco não é a eficiência do modelo, mas sim como ele é afetado pelos dados utilizados no treinamento.

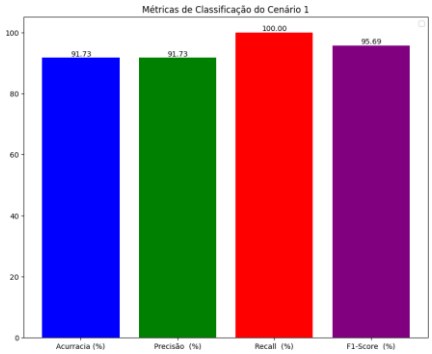



**Figura 5.10** Métricas de classificação do cenário 1

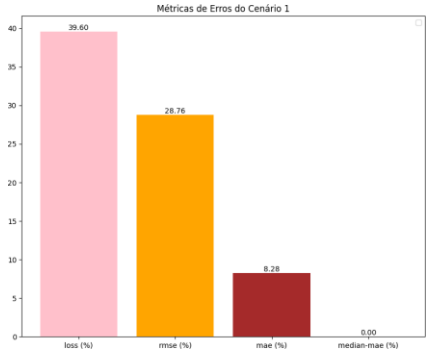



**Figura 5.11** Métricas de erros do cenário 1

Fonte: Elaborado pelo autor.

Apesar disso, o resultado do modelo foi satisfatório para a análise do que se pode esperar do treinamento de um modelo utilizando dados sem manipulação, sendo este o principal objetivo desta seção.

### 5.2.2 Cenário de Experimentos 2

Neste cenário, um algoritmo foi utilizado para simular um nó da rede que foi corrompido por um atacante, para alterar as amostras já registradas na base de dados. O nó corrompido substitui os dados por valores aleatórios entre 0 e 1000.

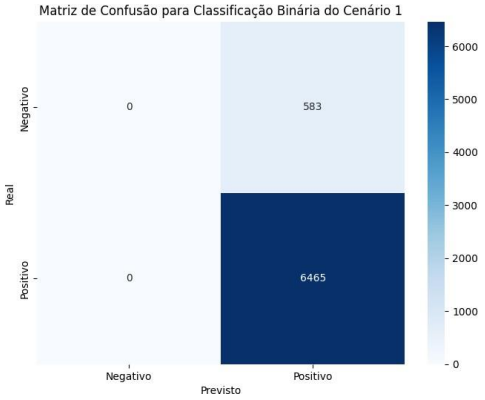



**Figura 5.12** Matriz de confusão dos resultados de predição do Cenário 1
Fonte: Elaborado pelo autor.

Espera-se que, neste cenário, o consumidor não receba os dados reais coletados pela fonte, corroborando a ideia de que uma IoT convencional não é adequada para ser utilizada como fonte de dados, pois não oferece recursos que garantam a confiabilidade das informações fornecidas aos consumidores.

Seguindo a mesma metodologia de análise aplicada no cenário 1, comparou-se a base de dados que representa o que deveria ser entregue ao consumidor com o que foi efetivamente recebido e utilizado no treinamento dos modelos. Ao observar as Figuras 5.13, nota-se uma discrepância no padrão de distribuição dos valores de temperatura e umidade.

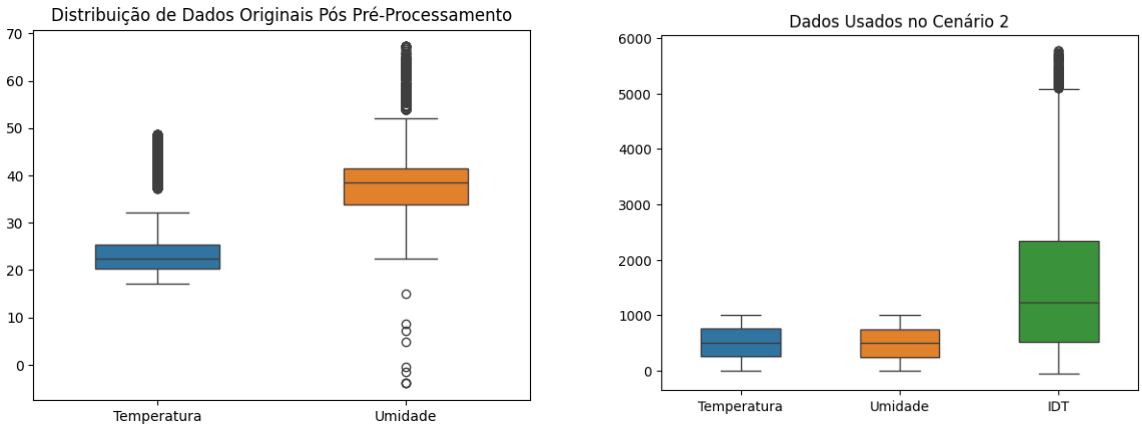



**Figura 5.13** Comparação dos dados originais e recebidos no cenário 2
Fonte: Elaborado pelo autor.

Ao analisarmos os valores apresentados na Figura 5.14(a), foi observado que o IDT está abaixo de 70. Entretanto, na Figura 5.14(b), os valores do IDT chegam a ultrapassar 6000°C. Isso caracteriza um cenário em que o consumidor recebeu dados diferentes dos coletados originalmente, indicando manipulação ou corrupção dos dados.

A geração de um ataque de manipulação de dados que provocasse uma grande discrepância em relação aos dados originais foi intencional, para evidenciar a vulnerabilidade

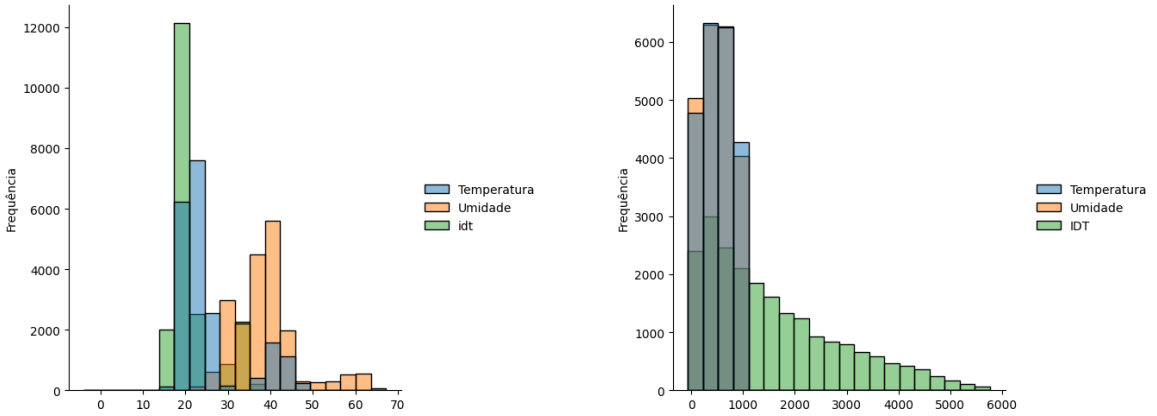



**Figura 5.14** Comparação dos histogramas dos *datasets*
Fonte: Elaborado pelo autor.

da arquitetura. Como era esperado, os modelos treinados com essas informações também apresentaram variações em relação ao cenário 1, conforme ilustrado nas Figuras 5.15 e 5.16.

Ao analisar as métricas de erro, observou-se um aumento de aproximadamente 200, 3% no *loss*, 200, 5% no RMSE, 1100, 9% no MAE e 100% no MedAE. Por outro lado, a precisão e a acurácia apresentaram uma redução de cerca de 89, 9%, sendo que a métrica que menos variou foi o *recall*, com uma redução de 0, 75%.

Essa comparação das métricas em cada cenário permite concluir que o *recall* talvez não seja uma métrica adequada para avaliar se o modelo foi afetado por dados corrompidos. Por outro lado, a métrica de erro MAE demonstrou maior sensibilidade à tentativa de ataque.

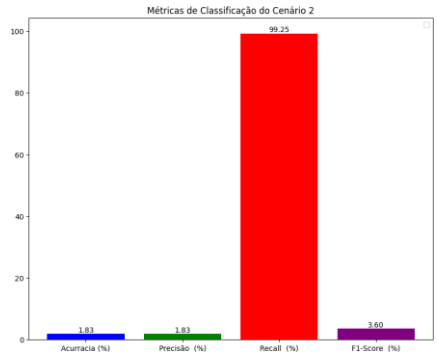



**Figura 5.15** Métricas de classificação do cenário 2

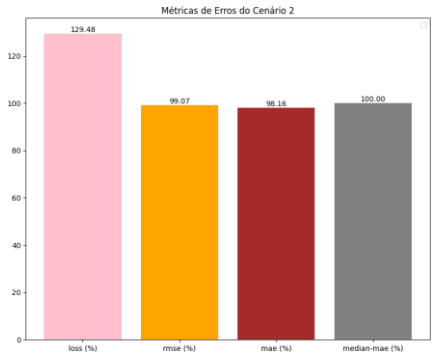



**Figura 5.16** Métricas de erros do cenário 2

Fonte: Elaborado pelo autor.

Ao examinar a matriz de confusão apresentada na Figura 5.17, observa-se que a frequência de verdadeiros positivos foi de apenas 132, enquanto o restante dos resultados consistiu em falsos positivos e falsos negativos, com uma predominância de falsos negativos. Conforme a Figura 5.15, esse resultado corresponde a apenas 1, 83% de acer-

tos em todas as tentativas, indicando que o modelo foi significativamente impactado pela corrupção dos dados, resultando em um desempenho insatisfatório.

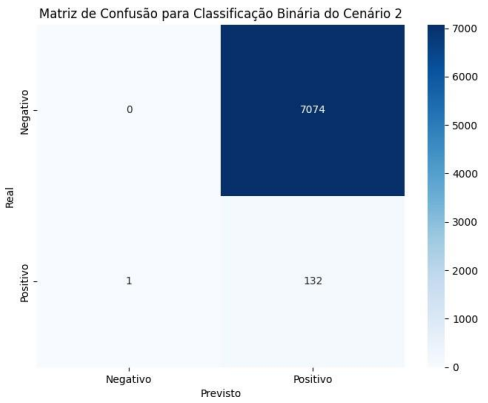



**Figura 5.17** Matriz de confusão dos resultados de predição do Cenário 2
Fonte: Elaborado pelo autor.

### 5.2.3 Cenário de experimentos 3

Neste cenário, avalia-se a capacidade do RFoT em garantir a confiabilidade dos dados, atuando como uma fonte que não foi submetida a tentativa de corrupção. Assim como no primeiro cenário, o objetivo principal é estabelecer uma *baseline* para comparação com o cenário 4, onde ocorre uma tentativa de corrupção dos dados.

Os dados apresentados nesta subseção resultam do treinamento de um modelo global, utilizando informações fornecidas pela arquitetura RFoT. Todos os dados recebidos da fonte são armazenados em um conjunto de dados, permitindo a comparação entre o que se espera receber e o que é efetivamente recebido pelo consumidor.

A Figura 5.18 ilustra a similaridade entre os dados recebidos e enviados, demonstrando que os dados utilizados no treinamento dos modelos mantêm o mesmo padrão da base original. Isso confirma que o consumidor recebeu os dados sem corrupção. Além disso, os valores do índice IDT calculados também estão em concordância com a base de dados utilizada.

Como os dados recebidos pelo consumidor foram idênticos aos do cenário 1, o desempenho de classificação obteve resultados equivalentes, com uma acurácia e precisão de 92%, *recall* de 100% e *F1-Score* de 96%. Esses resultados estão apresentados nas Figuras 5.19,5.20 e 5.21.

Ao final do treinamento de cada modelo, o consumidor solicita que o CI2 realize a mineração de um bloco na BCR, enviando o modelo treinado. Por meio dessa Blockchain, o RFoT oferece mecanismos para auditar cada modelo, uma vez que cada bloco da BCR armazena o *hash* do bloco da BCD utilizado no treinamento. Somente o modelo global não é registrado na BCR, pois é o resultado da integração dos modelos locais e não tem uma relação direta com um bloco da BCD. A Figura 5.22 apresenta uma visão da BCR.

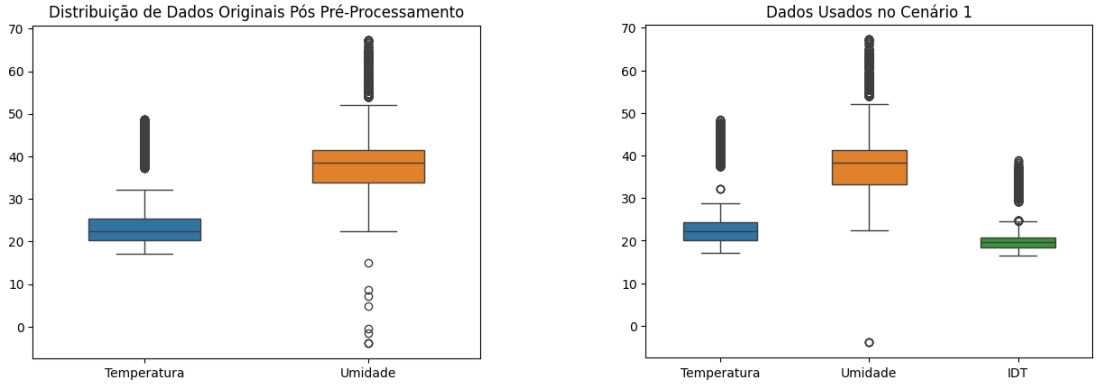



**Figura 5.18** Comparação dos dados originais e recebidos no cenário 3
Fonte: Elaborado pelo autor.

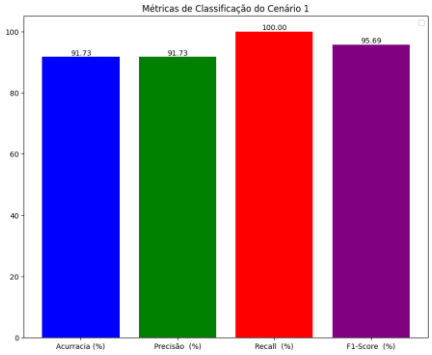



**Figura 5.19** Métricas de classificação do cenário 3

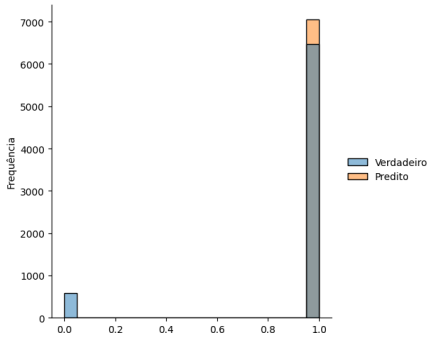



**Figura 5.20** Dados reais e preditos do cenário 3

Fonte: Elaborado pelo autor.

### 5.2.4 Cenário de experimentos 4

Após estabelecer uma *baseline* para compreender quais dados deveriam ser recebidos pelo consumidor e o desempenho de classificação dos modelos, foi desenvolvido o cenário 4. Neste cenário, o RFoT é submetido a uma tentativa de corrupção para verificar se esta se propaga até o consumidor.

Para que uma Blockchain seja completamente corrompida, o atacante precisa alterar o dado e reminerar todos os blocos subsequentes, uma vez que cada bloco está encadeado ao anterior por meio do *hash*. Além disso, para alcançar uma corrupção completa, o atacante deve conseguir comprometer pelo menos 51% das Blockchains da rede.

Nesta pesquisa, não foi possível simular uma tentativa de ataque tão abrangente, devido à sua complexidade e à falta de recursos. Assim, a tentativa de corrupção implementada neste cenário é representada pela alteração dos valores das amostras, conforme ocorreu no cenário 2 (subseção 5.2.2). Sob essa perspectiva, mesmo que todas as versões da Blockchain fossem corrompidas, nenhum dado alterado seria enviado ao consumidor, pois todas falhariam no algoritmo de consenso.

Foram geradas cinco versões de uma Blockchain válida, das quais apenas uma per-

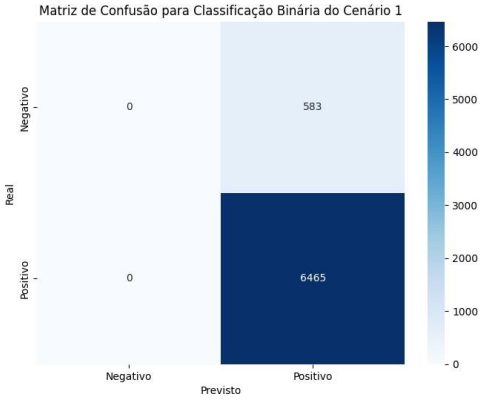



**Figura 5.21** Matriz de confusão dos resultados de predição do Cenário 3
Fonte: Elaborado pelo autor.

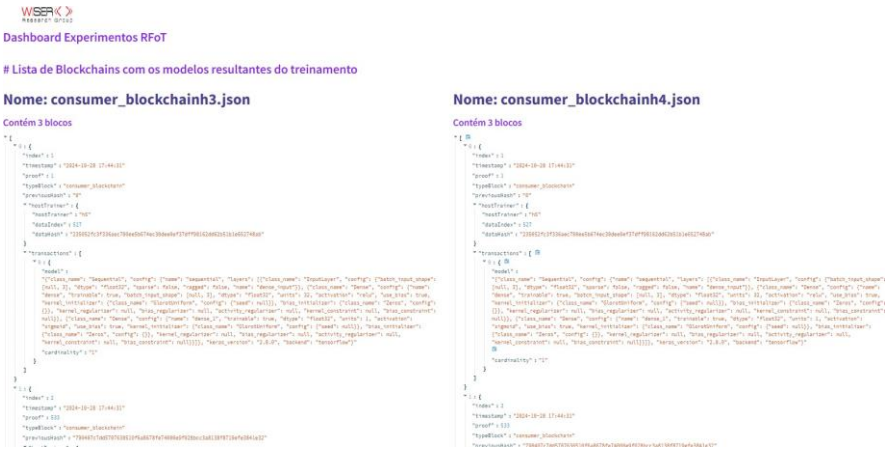



**Figura 5.22** Tela do Dashboard que apresenta as BCR geradas pelo consumidor
Fonte: Elaborado pelo autor.

maneceu intacta, já que o algoritmo de corrupção foi programado para não corromper a Blockchain número 3. Cada versão da Blockchain foi registrada em um arquivo distinto para simular nós distribuídos que compõem a rede, conforme ilustrado na Figura 5.23.

Pode-se observar que, na parte superior do *Dashboard*, é apresentada a versão da Blockchain que atende aos requisitos de uma Blockchain válida. Logo abaixo, são listadas todas as que apresentam uma quebra na cadeia, caracterizada por um *hash* que foi alterado, mas que continua sendo referenciado no bloco subsequente.

Das cinco Blockchains, a única que não aparece na lista de inválidas é a número 3, cujos dados são exibidos na seção de Blockchain válida. A Figura 5.24 apresenta a comparação dos dados entre a Blockchain válida e uma inválida.

Apesar de a maioria das Blockchains ter sido alterada, os blocos não foram reminerados. Este é um dos fatores que tornam a corrupção de dados em uma Blockchain tão desafiadora, pois o atacante precisa reminerar todos os blocos, substituindo cada *hash* que interliga os blocos da cadeia, antes que o algoritmo de consenso seja executado e

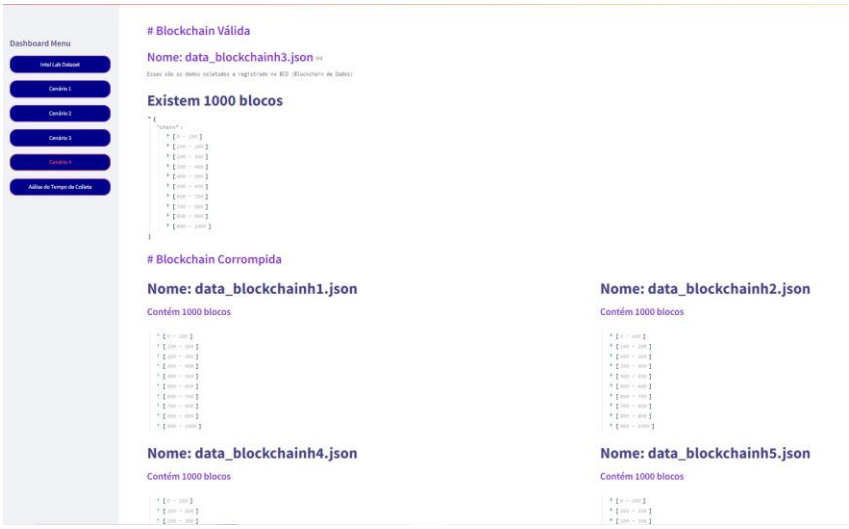



**Figura 5.23** Lista de Blockchains criadas para o experimento do cenário 4
Fonte: Elaborado pelo autor.

regularize todas as Blockchains.

Neste experimento, a refatoração das Blockchains corrompidas para a versão válida foi desativada, para demonstrar que apenas os dados de uma Blockchain válida podem ser retornados ao realizar uma requisição. Como consequência, o consumidor recebe os mesmos dados apresentados nos cenários 1 (subseção 5.2.1) e 3 (subseção 5.2.3), resultando em modelos com o mesmo desempenho de classificação.

## 5.3 RESULTADOS E DISCUSSÕES

Os experimentos apresentados nos cenários 1 e 2 evidenciam que uma das vulnerabilidades de uma IoT convencional é a incapacidade de garantir a confiabilidade dos dados coletados e armazenados. Isso ocorre porque não existem ferramentas adequadas para validar a consistência desses dados, bem como para gerir sua rastreabilidade.

Como consequência, o consumidor fica impossibilitado de utilizar a IoT como fonte de dados em contextos onde a manipulação de dados é inaceitável ou apresenta riscos ao projeto, ou à vida. Isso se deve ao fato de que os dados recebidos pela fonte podem não corresponder aos dados originalmente coletados. Além disso, as amostras utilizadas podem influenciar diretamente o produto resultante.

O fato de uma AI ser impactada pelos dados utilizados em seu treinamento é um dos motivos para a escolha de um sistema de AM para representar o consumidor. Esse efeito pode ser observado ao comparar os resultados do treinamento, utilizando as métricas de classificação dos cenários 1 e 2, conforme apresentado na Figura 5.25.

Por outro lado, a IoT pode se tornar uma fonte de dados que oferece um grande volume de informações com qualidade e aplicável a contextos de alta criticidade, desde que a confiabilidade da fonte seja garantida, proporcionando segurança, privacidade, imutabilidade e rastreabilidade dos dados.

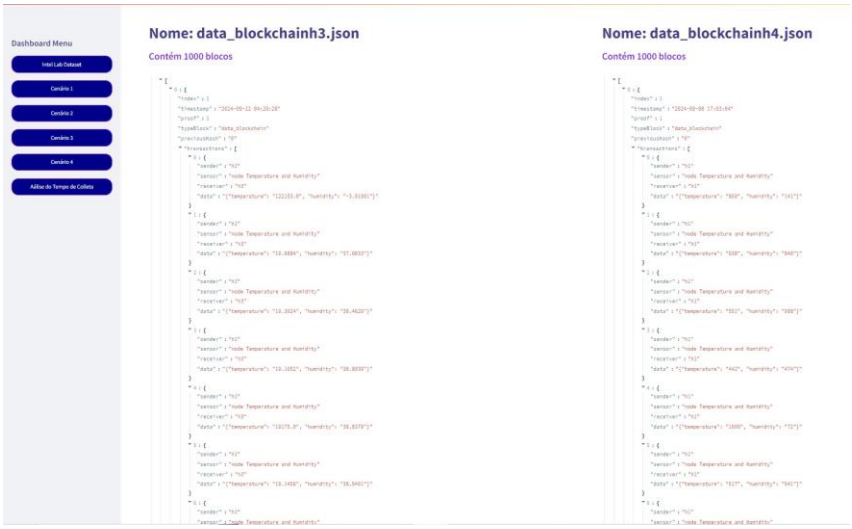



**Figura 5.24** Comparação entre Blockchain válida e inválida no cenário 4
Fonte: Elaborado pelo autor.

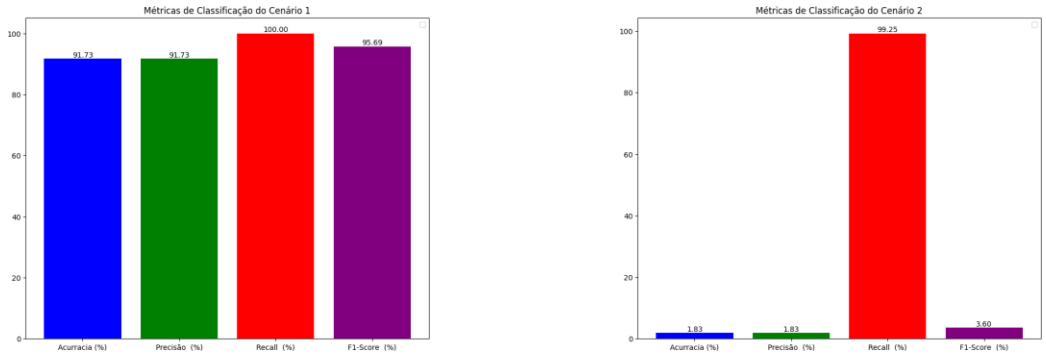



**Figura 5.25** Comparação das métricas de classificação dos cenários 1 e 2
Fonte: Elaborado pelo autor.

Portanto, o RFoT se apresenta como uma proposta de solução para tornar a IoT uma fonte confiável. Embora nesta pesquisa não tenha sido possível abordar todas as possibilidades de ataque, os cenários 3 e 4 permitiram compreender as vantagens da integração da IoT com a Blockchain e os CI.

Como resultado, os modelos treinados no cenário 4 não foram afetados, mesmo diante de tentativas de corrupção, garantindo que o consumidor recebesse exatamente os dados coletados. Isso é ilustrado na Figura 5.26.

## 5.4 ANÁLISE DE CUSTO COMPUTACIONAL

Esta seção é dividida em uma análise assintótica do processo de coleta e armazenamento dos dados e do custo de banda, baseado no tamanho dos pacotes de mensagens publicados na rede em cada fase do framework.

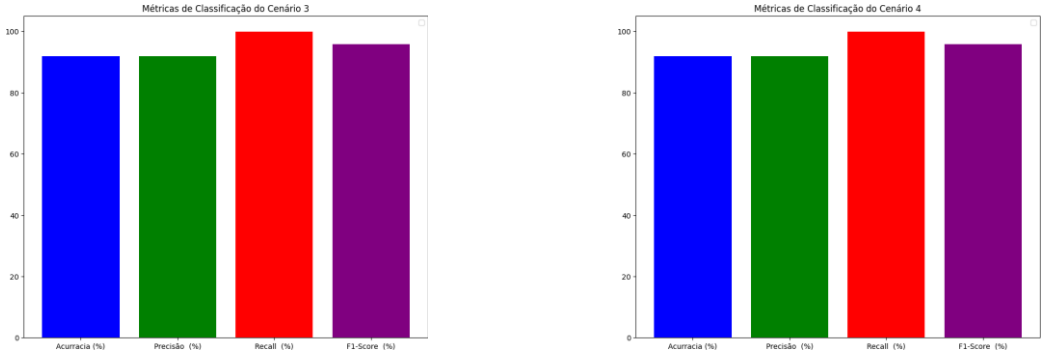



**Figura 5.26** Comparação das métricas de classificação dos cenários 3 e 4
Fonte: Elaborado pelo autor.

### 5.4.1 Análise Assintótica

A análise assintótica foi realizada utilizando a notação de Big O para o processo de coleta e armazenamento dos dados e para o processo de disponibilização da informação, a fim de compreender o impacto da execução dos algoritmos, de acordo com o crescimento da entrada. Desse modo, o custo de cada camada é apresentado no Quadro 5.1, considerando o conjunto de processos executados em cada etapa.

| Camada | Custo | Processo |
|---|---|---|
| Física | $O(1)$ | Coleta de amostras |
| Monitoramento e Controle | $O(1)$ | Publicação das amostras na rede |
| Suporte | $O(n^2)$ | Tratamento e Registro na Blockchain |
| Negócio | $O(n^2)$ | Recuperação e envio para consumidor |

Quadro 5.1: Resultado da análise assintótica para cada camada do RFoT

Os processos de coleta/armazenamento e distribuição das amostras são os dois principais processos que podem impactar o custo computacional e a latência da aplicação. O custo da coleta/armazenamento é calculado a partir do somatório do tempo de coleta da camada Física ($O(1)$), com o tempo de publicação da camada de Monitoramento e Controle ($O(1)$), o tempo de acúmulo ($O(n)$) e o tempo de registro na Blockchain ($O(n^2)$) na camada de Suporte. Este cálculo é representado através da equação 5.17. Seguindo os padrões estabelecidos pela notação Big O, o tempo de execução do algoritmo desde a coleta até o armazenamento cresce com a entrada com no máximo $O(n^2)$.

$$coleta/armazenamento = O(1) + O(1) + O(n) + O(n^2) = O(n^2) \quad (5.17)$$

Já para o processo de distribuição das amostras para um consumidor é representado pelo somatório da requisição realizada pelo consumidor ao Contrato Inteligente 3 $O(1)$, com o acionamento do algoritmo de consenso para obtenção de uma Blockchain válida $O(n^2)$ e retorno de um bloco da Blockchain de Resultados $O(n)$, resultando em um tempo de execução de $O(n^2)$, como apresentado na equação 5.18

$$distribuicao = O(1) + O(n^2) + O(n) = O(n^2) \quad (5.18)$$

### 5.4.2 Análise de Custo de Rede

Ao realizar o monitoramento do percurso das amostras na arquitetura do RFoT, foi possível mapear e quantificar a quantidade de dados compartilhada entre os dispositivos em cada etapa. Essa análise visa oferecer ao leitor uma perspectiva sobre a viabilidade da arquitetura para o cenário desejado.

É importante destacar que o cenário proposto neste trabalho foi desenvolvido com o intuito de analisar o comportamento do framework em comparação a uma arquitetura convencional, buscando determinar sua capacidade de garantir a confiabilidade da informação.

Como as amostras de temperatura e umidade são valores decimais, a análise de custo computacional e arquitetural será proporcionalmente compatível com qualquer medida que possa ser representada por um valor decimal, dado que o tamanho de uma amostra decimal equivale a 48 bytes.

A análise foi dividida em duas etapas: a primeira consiste na mensuração do tamanho das estruturas com e sem a adição das amostras, conforme ilustrado na Tabela 5.5. A análise baseou-se na coleta de apenas uma amostra de dados, que consiste em um conjunto formado por uma medida de temperatura e uma de umidade.

| Nome | Cabeçalho (bytes) | Cabeçalho+dados (bytes) |
|---|---|---|
| Nó de Sensores | 154 | 222 |
| Pacote FLOW | 184 | 384 |
| Bloco Blockchain | 384 | 427 |

**Tabela 5.5** Tabela que apresenta o tamanho em bytes das mensagens trocadas pelos dispositivos com e sem dados

A segunda etapa consistiu em uma análise do fluxo completo dos dados, registrando o tamanho em bytes das mensagens trocadas entre os principais componentes da arquitetura. Esse processo abrange desde a coleta das amostras na camada física até a camada de suporte, incluindo a mineração das amostras em um bloco da Blockchain de registro de dados. Consulte a Figura 5.27 para mais detalhes.

## 5.5 ANÁLISE DE CUSTO ARQUITETURAL

Esta seção apresenta uma análise dos dispositivos que podem compor cada camada da arquitetura, visando atender aos requisitos de capacidade de memória e processamento, focando na minimização dos custos financeiros e computacionais.

### 5.5.1 Camada 1: Sensores

- **Sensor DHT11/DHT22**

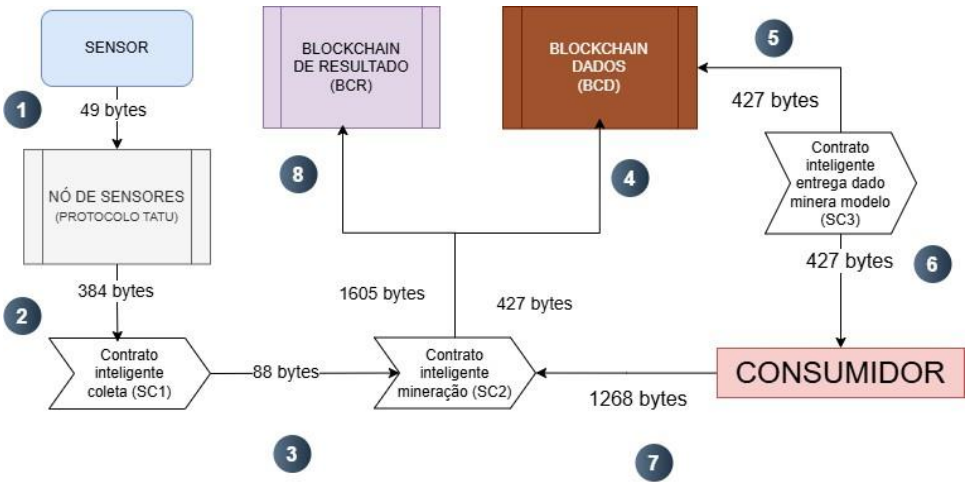



**Figura 5.27** Comparação entre Blockchain válida e inválida no cenário 4
Fonte: Elaborado pelo autor.

– **Capacidade de Memória**: Esses sensores não possuem memória interna para armazenamento de dados. Eles transmitem dados diretamente para o microcontrolador ou dispositivo de borda ao qual estão conectados.

### 5.5.2 Camada 2: Nós de Borda

- **Raspberry Pi Zero W**
  - **Memória RAM**: 512 MB
  - **Armazenamento**: Utiliza um cartão microSD para armazenamento, cuja capacidade pode variar (geralmente de 8 GB a 32 GB ou mais).
  - **Processamento**:Broadcom BCM2835 single-core 1GHz , aliado à memória de 512MB, permite que você crie aplicações controlando a GPIO de 40 pinos
  - **Considerações**: Oferece uma boa capacidade de memória para executar aplicações leves e armazenar dados localmente, especialmente quando combinado com um cartão microSD de maior capacidade.
- **ESP32**
  - **Memória RAM**: Aproximadamente 520 KB de SRAM
  - **Armazenamento Flash**: Tipicamente 4 MB, mas pode variar dependendo do modelo específico.
  - **Processamento**: Xtensa LX6 32-bit, que podem operar a uma frequência de até 240 MHz. A frequência mais comum é de 160 MHz, o que é 10 vezes mais rápido que o clock do Arduino Uno.
  - **Considerações**: Embora tenha menos memória RAM e armazenamento, é adequado para aplicações IoT que exigem baixo consumo de energia e custo.

### 5.5.3 Camada 3: Nuvem

- **AWS Free Tier (Instância t2.micro)**

  - **Memória RAM**: 1 GB
  - **Armazenamento**: O armazenamento em nuvem pode ser configurado conforme necessário, com opções de armazenamento em bloco (EBS) que variam em capacidade e custo.
  - **Considerações**: A capacidade de armazenamento e memória pode ser ajustada conforme necessário, mas é importante monitorar o uso para evitar custos adicionais.

### 5.5.4 Análise de Impacto Temporal da Integração das Tecnologias

A estratégia adotada para solucionar o problema de confiabilidade dos dados geridos na Edge de uma IoT foi a integração com outras tecnologias, visando sobrepor os pontos positivos de cada uma sobre os pontos negativos das demais.

Entretanto, essa integração pode impactar o desempenho do RFoT, dado que cada tecnologia realiza diferentes tipos de processamento. Com isso em mente, foi desenvolvido um experimento para comparar o tempo de coleta e armazenamento das amostras, para compreender o impacto da incorporação da Blockchain e dos contratos inteligentes na arquitetura da IoT.

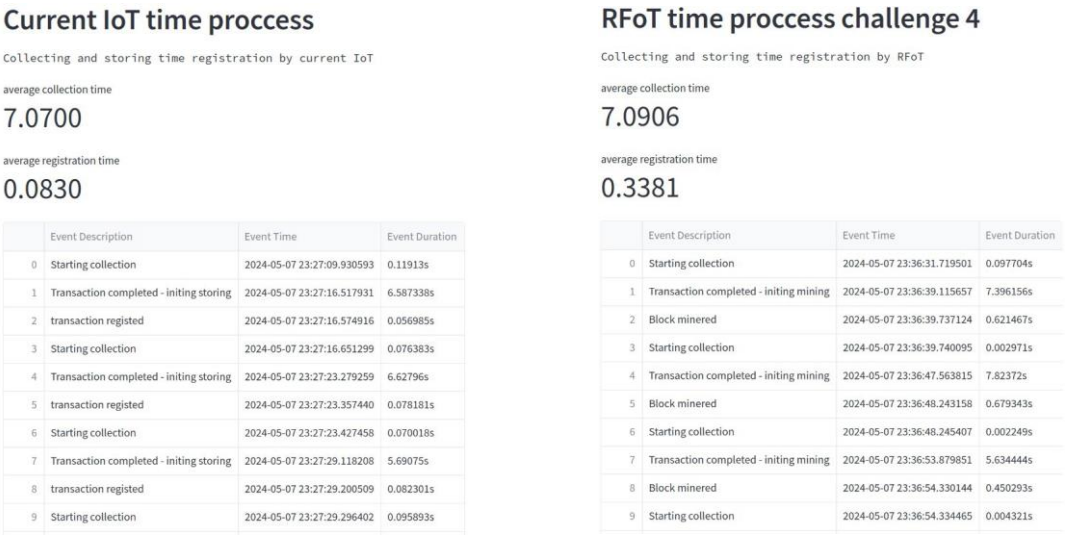
O experimento foi conduzido utilizando o GD proposto no estudo de caso, registrando a duração dos eventos de coleta e armazenamento das amostras. Isso permitiu analisar a diferença temporal entre um simples registro em arquivo e o evento de mineração de um bloco, seguido do registro em arquivo.

É importante destacar que a mineração de um bloco envolve a resolução de um desafio para obter um *hash* aceitável. O desafio utilizado neste trabalho foi a geração de um *hash* cujos quatro primeiros dígitos são zeros. Outro ponto relevante é que o tempo de coleta depende do tamanho do bloco de amostras.

Os resultados do experimento indicaram que a diferença no tempo de coleta não excedeu dois segundos, enquanto a diferença no tempo de registro foi da ordem de $10^{-1}$ segundos, conforme ilustrado na Figura 5.28.

A aleatoriedade do algoritmo de *hash* acabou contribuindo para que a média temporal dos eventos do RFoT fosse próxima da produzida em uma IoT convencional. Assim, o impacto temporal está relacionado ao tamanho do bloco de amostras especificado para a coleta e à dificuldade do desafio proposto.

Para melhor entendimento dessa variação, foram realizados os mesmos experimentos na RFoT para os desafios de *hash* iniciando com cinco, seis e sete zeros. Infelizmente, a limitação de memória do GD não possibilitou os testes com um desafio maior, uma vez que o cálculo não terminava a tempo. Ver Figuras 5.29 e 5.30.



**Current IoT time proccess**

Collecting and storing time registration by current IoT

average collection time

7.0700

average registration time

0.0830

| | Event Description | Event Time | Event Duration |
|---|---|---|---|
| 0 | Starting collection | 2024-05-07 23:27:09.930593 | 0.11913s |
| 1 | Transaction completed - initing storing | 2024-05-07 23:27:16.517931 | 6.587338s |
| 2 | transaction registed | 2024-05-07 23:27:16.574916 | 0.056985s |
| 3 | Starting collection | 2024-05-07 23:27:16.651299 | 0.076383s |
| 4 | Transaction completed - initing storing | 2024-05-07 23:27:23.279259 | 6.62796s |
| 5 | transaction registed | 2024-05-07 23:27:23.357440 | 0.078181s |
| 6 | Starting collection | 2024-05-07 23:27:23.427458 | 0.070018s |
| 7 | Transaction completed - initing storing | 2024-05-07 23:27:29.118208 | 5.69075s |
| 8 | transaction registed | 2024-05-07 23:27:29.200509 | 0.082301s |
| 9 | Starting collection | 2024-05-07 23:27:29.296402 | 0.095893s |

**RFoT time proccess challenge 4**

Collecting and storing time registration by RFoT

average collection time

7.0906

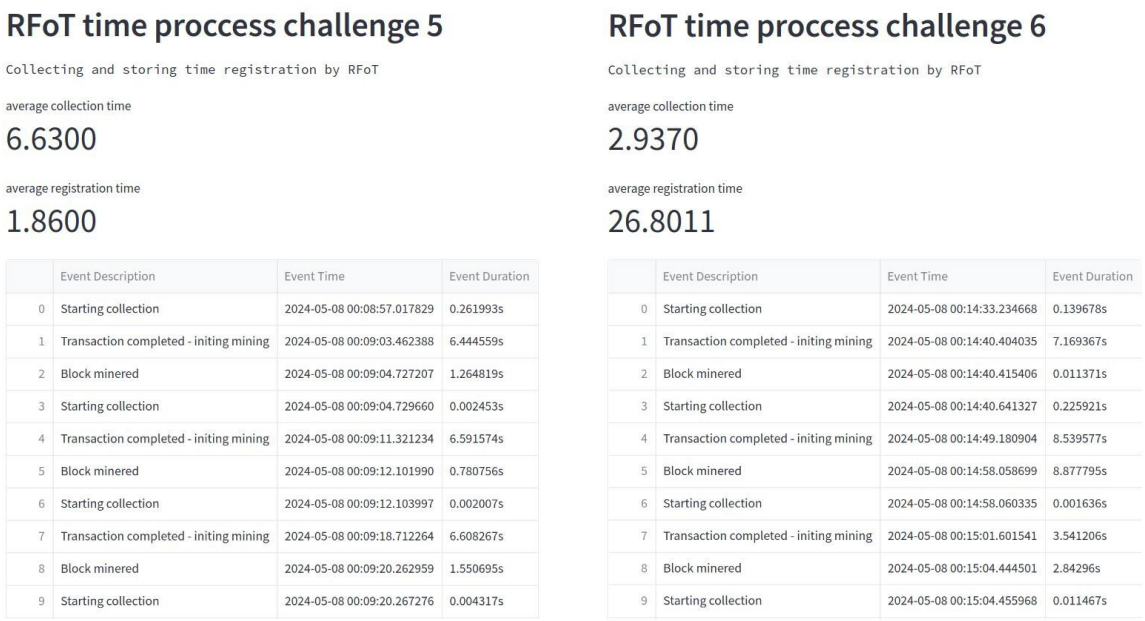
average registration time

0.3381

| | Event Description | Event Time | Event Duration |
|---|---|---|---|
| 0 | Starting collection | 2024-05-07 23:36:31.719501 | 0.097704s |
| 1 | Transaction completed - initing mining | 2024-05-07 23:36:39.115657 | 7.396156s |
| 2 | Block minered | 2024-05-07 23:36:39.737124 | 0.621467s |
| 3 | Starting collection | 2024-05-07 23:36:39.740095 | 0.002971s |
| 4 | Transaction completed - initing mining | 2024-05-07 23:36:47.563815 | 7.82372s |
| 5 | Block minered | 2024-05-07 23:36:48.243158 | 0.679343s |
| 6 | Starting collection | 2024-05-07 23:36:48.245407 | 0.002249s |
| 7 | Transaction completed - initing mining | 2024-05-07 23:36:53.879851 | 5.634444s |
| 8 | Block minered | 2024-05-07 23:36:54.330144 | 0.450293s |
| 9 | Starting collection | 2024-05-07 23:36:54.334465 | 0.004321s |

**Figura 5.28** Análise temporal dos eventos de coleta e registro dos dados na IoT convencional e RFoT

Fonte: Elaborado pelo autor.



**RFoT time proccess challenge 5**

Collecting and storing time registration by RFoT

average collection time

6.6300

average registration time

1.8600

| | Event Description | Event Time | Event Duration |
|---|---|---|---|
| 0 | Starting collection | 2024-05-08 00:08:57.017829 | 0.261993s |
| 1 | Transaction completed - initing mining | 2024-05-08 00:09:03.462388 | 6.444559s |
| 2 | Block minered | 2024-05-08 00:09:04.727207 | 1.264819s |
| 3 | Starting collection | 2024-05-08 00:09:04.729660 | 0.002453s |
| 4 | Transaction completed - initing mining | 2024-05-08 00:09:11.321234 | 6.591574s |
| 5 | Block minered | 2024-05-08 00:09:12.101990 | 0.780756s |
| 6 | Starting collection | 2024-05-08 00:09:12.103997 | 0.002007s |
| 7 | Transaction completed - initing mining | 2024-05-08 00:09:18.712264 | 6.608267s |
| 8 | Block minered | 2024-05-08 00:09:20.262959 | 1.550695s |
| 9 | Starting collection | 2024-05-08 00:09:20.267276 | 0.004317s |

**RFoT time proccess challenge 6**

Collecting and storing time registration by RFoT

average collection time

2.9370

average registration time

26.8011

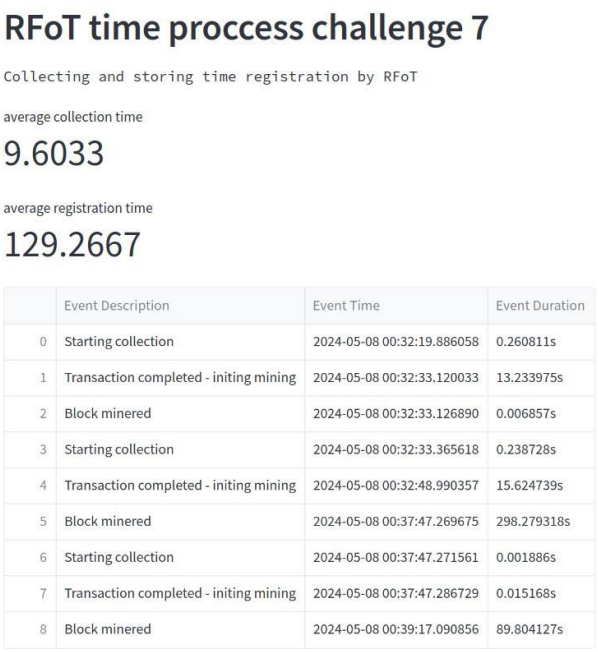
| | Event Description | Event Time | Event Duration |
|---|---|---|---|
| 0 | Starting collection | 2024-05-08 00:14:33.234668 | 0.139678s |
| 1 | Transaction completed - initing mining | 2024-05-08 00:14:40.404035 | 7.169367s |
| 2 | Block minered | 2024-05-08 00:14:40.415406 | 0.011371s |
| 3 | Starting collection | 2024-05-08 00:14:40.641327 | 0.225921s |
| 4 | Transaction completed - initing mining | 2024-05-08 00:14:49.180904 | 8.539577s |
| 5 | Block minered | 2024-05-08 00:14:58.058699 | 8.877795s |
| 6 | Starting collection | 2024-05-08 00:14:58.060335 | 0.001636s |
| 7 | Transaction completed - initing mining | 2024-05-08 00:15:01.601541 | 3.541206s |
| 8 | Block minered | 2024-05-08 00:15:04.444501 | 2.84296s |
| 9 | Starting collection | 2024-05-08 00:15:04.455968 | 0.011467s |

**Figura 5.29** Análise temporal dos eventos de coleta e registro dos dados no RFoT para os desafios de *hash* com cinco e seis zeros no início

Fonte: Elaborado pelo autor.

## 5.6 CONSIDERAÇÕES FINAIS

A implementação das arquiteturas a partir da simulação, permitiu constatar que a estrutura da IoT convencional não está apta a garantir a confiabilidade dos dados, podendo comprometer o resultado de processos que consumam os dados fornecidos por ela.



RFoT time proccess challenge 7

Collecting and storing time registration by RFoT

average collection time

9.6033

average registration time

129.2667

| | Event Description | Event Time | Event Duration |
|---|---|---|---|
| 0 | Starting collection | 2024-05-08 00:32:19.886058 | 0.260811s |
| 1 | Transaction completed - initing mining | 2024-05-08 00:32:33.120033 | 13.233975s |
| 2 | Block minered | 2024-05-08 00:32:33.126890 | 0.006857s |
| 3 | Starting collection | 2024-05-08 00:32:33.365618 | 0.238728s |
| 4 | Transaction completed - initing mining | 2024-05-08 00:32:48.990357 | 15.624739s |
| 5 | Block minered | 2024-05-08 00:37:47.269675 | 298.279318s |
| 6 | Starting collection | 2024-05-08 00:37:47.271561 | 0.001886s |
| 7 | Transaction completed - initing mining | 2024-05-08 00:37:47.286729 | 0.015168s |
| 8 | Block minered | 2024-05-08 00:39:17.090856 | 89.804127s |

**Figura 5.30** Análise temporal dos eventos de coleta e registro dos dados no RFoT para os desafios de *hash* com sete zeros no início

Fonte: Elaborado pelo autor.

A solução proposta se mostrou eficaz em prover automação, disponibilidade, privacidade, rastreabilidade e a imutabilidade da informação, tornando a IoT uma fonte de dados confiável para atuar como fonte de dados.

O RFoT Apresenta nível de segurança comparável ao proposto por Zhou (2021), ao utilizar uma estrutura similar de CI, diferenciando-se no foco de segurança dos dados ao em vez da autenticação de usuários.

A solução desenvolvida nesse trabalho supera a solução proposta por Kebande et al. (2020), ao prover a rastreabilidade desde a coleta dos dados até o treinamento do modelo global e a automatização do processo utilizando CI.

O trabalho de Shukla et al. (2021) valida a utilização de uma simulação para representar o contexto real sem perda de fidelidade. Já os trabalhos de Tripathi, Ahad e Paiva (2020) e Shan e Mai (2020), juntamente com os demais, validam a Blockchain como solução pode ser barata e eficiente.

Esse trabalhado está sendo desenvolvido na linguagem *Python* e versionado na plataforma *GitHub* em um repositório público, certificado pela plataforma *Zenodo*. Todas as configurações utilizadas, juntamente com o passo a passo para instalação do ambiente, estão explanadas no *readme* Silva e Brenno (2024).

Capítulo 6

# CONCLUSÃO

Com relação à confiabilidade dos dados, foi possível identificar uma vulnerabilidade na arquitetura de uma IoT convencional, que não apresenta meios eficazes para impedir a propagação da corrupção quando os dados são alvo de manipulação. Conforme as hipóteses apresentadas nesta pesquisa, essa arquitetura não atende aos requisitos necessários para fornecer dados confiáveis para o treinamento de modelos de AM.

A incapacidade da IoT convencional em prover confiabilidade ficou evidente na comparação dos resultados do cenário 2, onde os modelos treinados com dados corrompidos apresentaram um desempenho de classificação inferior em relação aos modelos do cenário 1, que utilizaram dados originais. Diferentemente das demais métricas de classificação, o *Recall* não apresentou grande variação, sugerindo que pode não ser uma métrica viável para identificar modelos que sofreram corrupção. Em contrapartida, as métricas de erro *loss* e MedAE destacaram o declínio dos resultados.

O RFoT demonstrou que, mesmo que um atacante consiga burlar todas as barreiras da arquitetura e realizar alterações nas informações já registradas na Blockchain, os dados originais são restaurados e enviados ao consumidor sem propagação do ataque. Sua imutabilidade está limitada no caso de o atacante corromper mais de 51% da rede, conforme estabelecem as diretrizes da tecnologia Blockchain.

Conclui-se, portanto, que o RFoT assegura a confiabilidade das amostras registradas na Blockchain, proporcionando automação, disponibilidade, privacidade, rastreabilidade e imutabilidade. Isso eleva o nível de dificuldade para a implementação de ataques de manipulação de dados e oferece um sistema de restauração das informações, caso sejam corrompidas, sem comprometer a disponibilidade.

## 6.1 RESULTADOS OBTIDOS

- A integridade do processo de pertinência dos dados foi obtida a partir da automação implementada pelos CI, reduzindo a interferência humana e a possibilidade de manipulação da informação transportada;

- A privacidade foi garantida a través da aplicação da criptografia simétrica sobre os dados, desde o processo de coleta até a pertinência. Além disso;
- A imutabilidade dos dados foi obtida a partir da tecnologia Blockchain, que protege os dados contra-ataques de alteração da informação já registrada, ou seja, uma vez que foi minerado para um bloco da rede;
- Além da imutabilidade, a Blockchain também proporciona a rastreabilidade, visto que a cadeia de dados ser uma estrutura organizada encadeadamente e cada bloco registra o momento da mineração, os dispositivos responsáveis pela coleta, mineração e consumo dos dados, além do próprio dado;
- O RFoT oferece interfaces de comunicação com os consumidores, proporcionando a segregação e encapsulamento, mantendo a disponibilidade da fonte de dados;
- A fim de compreender a cobertura de proteção proposta pelo RFoT, foi desenvolvido um sistema de parametrização que possibilidade avaliar o contexto de segurança do dado em cada etapa do processo de manipulação do dado;
- OGD desenvolvido se mostrou útil para implementação de toda a arquitetura da fonte de dados, possibilitando a realização de análises de custo computacional, arquitetural e temporal, para servir como base a novas perspectivas.

## 6.2 LIMITAÇÕES DO TRABALHO

A solução proposta garante a confiabilidade dos dados no contexto da Edge de uma rede IoT, por tanto, apenas as redes que implementam as arquiteturas dos cenários **B** e **C** (capítulo 2.1) são viáveis para implementação do RFoT.

além disso, ao analisar o RFoT, entendeu-se que ele não é capaz de garantir o certificado RFoT-S1. Isso porque, não foi implementado uma solução para garantia de confiabilidade dos sensores. Como apresentado no Capítulo 2, existem abordagens para garantir a autenticidade dos dispositivos na rede, o que levou a uma despriorização do desenvolvimento de uma solução para esse ponto nesse trabalho.

Também não foram desenvolvidas soluções de confiabilidade para as camadas que consomem os dados da Edge, desse modo, o RFoT só garante a confiabilidade do dado até a entrega da informação para um consumidor, por existir uma variedade de tipos de consumidores e por que cada um necessitar de uma análise personalizada.

## 6.3 TRABALHOS FUTUROS

Como apresentado na seção anterior, o RFoT apresenta limitações na cobertura de confiabilidade da Edge. Desse modo, como trabalho futuro propomos:

- A integração do RFoT com o PUF, para garantia de confiabilidade dos dispositivos que se conectam a rede, principalmente os sensores, uma vez que a manipulação das etapas seguintes são geridas por CI. Essa integração completaria a cobertura de confiabilidade do RFoT desde a comunicação com as entidades físicas.

- Outra integração possível seria o desenvolvimento de uma AI que ficaria responsável por monitorar e avaliar a confiabilidade e/ou a qualidade dos dados gerenciados pelo RFoT, a fim de indicar em tempo real se está acontecendo ou não uma tentativa de ataque à arquitetura, ou se um conjunto sequencial de blocos deveria ser descartado, fazendo com que o RFoT reiniciasse a Blockchain a partir de um bloco considerado confiável. Desse modo, um sistema de recuperação de desastre (considerando um ataque de possível corrupção em uma camada que esteja vulnerável) seria incluído na arquitetura.

# REFERÊNCIAS BIBLIOGRÁFICAS


ABADI, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2022. Disponível na Internet em <https://www.tensorflow.org/>. Software available from tensorflow.org.

AL-SARAWI, S. et al. Internet of things (iot) communication protocols. In: IEEE. *2017 8th International conference on information technology (ICIT)*. [S.l.], 2017. p. 685–690.

ALAM, T.; BENAIDA, M.; RAZZAQUE, M. A. A survey on the roles of communication technologies in iot-based personalized healthcare applications. *IEEE Access*, IEEE, v. 10, p. 12474–12513, 2022.

ALI, M.; KARIMIPOUR, H.; TARIQ, M. Integration of blockchain and federated learning for internet of things: Recent advances and future challenges. *Computers & Security*, Elsevier, p. 102355, 2021.

Amazon Web Services. *O que é IoT? - Internet das Coisas*. 2023. <https://aws.amazon.com/pt/what-is/iot/>. Acesso em: 10 de outubro de 2023.

ANDRADE, L.; SERRANO, M.; PRAZERES, C. The data interplay for the fog of things: A transition to edge computing with iot. In: *2018 IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2018. p. 1–7. Cited By :7.

ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. *Computer networks*, Elsevier, v. 54, n. 15, p. 2787–2805, 2010.

BATISTA, E. P.; PRAZERES, C. V. S.; FIGUEIREDO, G. B. *Modelagem, Balanceamento De Carga E Predição De Fluxos No Paradigma Névoa Das Coisas*. Dissertação (mestrado) — Universidade Federal do Recôncavo da Bahia, dec 2017.

BONOMI, F. et al. Fog computing and its role in the internet of things. In: ACM. *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. [S.l.], 2012. p. 13–16.

CAI, J. et al. Digital twin for healthy indoor environment: A vision for the post-pandemic era. *Frontiers of Engineering Management*, v. 10, n. Issue Number, p. 300 – 318, 2023.

CHAI, T.; DRAXLER, R. R. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, Copernicus GmbH, v. 7, n. 3, p. 1247–1250, 2014.

CHAMOLA, V. et al. A comprehensive review of the covid-19 pandemic and the role of iot, drones, ai, blockchain, and 5g in managing its impact. *IEEE Access*, IEEE, v. 8, p. 90225–90265, 2020.

CHICCO, D.; TÖTSCH, N.; JURMAN, G. The matthews correlation coefficient (mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData Mining*, BioData Mining, v. 14, n. 1, p. 1–22, 2021.

DAI, H.-N.; ZHENG, Z.; ZHANG, Y. Blockchain for internet of things: A survey. *IEEE Internet of Things Journal*, IEEE, v. 6, n. 5, p. 8076–8094, 2019.

DEY, N. et al. (Ed.). *Internet of Things and big data analytics toward next-generation intelligence*. [S.l.]: Springer, 2021.

DINH, T. T. A. et al. Untangling blockchain: A data processing view of blockchain systems. *IEEE Transactions on Knowledge and Data Engineering*, v. 30, n. 7, p. 1366–1385, 2018.

ELMAMY, S. B. et al. A survey on the usage of blockchain technology for cyber-threats in the context of industry 4.0. *Sustainability (Switzerland)*, v. 12, n. 21, p. 1–19, 2020. Cited By :6. Disponível em: <www.scopus.com>.

GARG, S. et al. A survey on security and privacy challenges in internet of things. *IEEE Internet of Things Journal*, IEEE, v. 8, n. 13, p. 10367–10383, 2021.

GODOY, D. *Understanding binary cross-entropy / log loss: a visual explanation*. 2018. Disponível na Internet em <https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a>.

GOUDARZI, S. et al. Data collection using unmanned aerial vehicles for internet of things platforms. *Computers & Electrical Engineering*, Elsevier, v. 89, p. 106903, 2021.

GOUTTE, C.; GAUSSIER, E. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In: SPRINGER. *European conference on information retrieval*. [S.l.], 2005. p. 345–359.

GRIEVES, M. Digital twin: manufacturing excellence through virtual factory replication. *White paper*, v. 1, n. 2014, p. 1–7, 2014.

GROUP, O. C. A. W. *OpenFog Reference Architecture for Fog Computing*. 2017. Disponível na Internet em <https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf>.

HABER, S.; STORNETTA, W. S. How to time-stamp a digital document. *Journal of Cryptology*, v. 3, p. 99–111, 1991.

HAMDAN, S.; AYYASH, M.; ALMAJALI, S. Edge-computing architectures for internet of things applications: A survey. *Sensors*, MDPI, v. 21, n. 22, p. 7580, 2021.

HAQUE, R. U. et al. Privacy-preserving k-nearest neighbors training over blockchain-based encrypted health data. *Electronics (Switzerland)*, v. 9, n. 12, p. 1–20, 2020. Cited By :3. Disponível em: <www.scopus.com>.

HOLST, A. *Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2023, with forecasts from 2022 to 2030*. 2024. Disponível na Internet em <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>.

HUSSAIN, F. et al. Machine learning in iot security: Current solutions and future challenges. *IEEE Communications Surveys Tutorials*, v. 22, n. 3, p. 1686–1721, 2020.

IBM. *O que é a tecnologia blockchain*. 2021. Disponível na Internet em <https://www.ibm.com/br-pt/topics/what-is-blockchain>.

JIANG, Y.; LIAN, Z. High performance and scalable byzantine fault tolerance. In: *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. [S.l.: s.n.], 2019. p. 1195–1202.

KEBANDE, V. R. et al. Leveraging federated learning blockchain to counter adversarial attacks in incremental learning. In: *PervasiveHealth: Pervasive Computing Technologies for Healthcare*. [S.l.: s.n.], 2020. Cited By :1.

KERRISON, S.; JUSAK, J.; HUANG, T. Blockchain-enabled iot for rural healthcare: Hybrid-channel communication with digital twinning. *Electronics*, v. 12, n. 9, p. 2128, 2023.

KOSBA, A. et al. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In: *2016 IEEE Symposium on Security and Privacy (SP)*. [S.l.: s.n.], 2016. p. 839–858.

KRAFT, E. M. The air force digital thread/digital twin-life cycle integration and use of computational and experimental knowledge. In: *54th AIAA aerospace sciences meeting*. [S.l.: s.n.], 2016. p. 0897.

LANTZ, B. et al. *Mininet*. [S.l.]: GitHub, 2022. Disponível na Internet em <https://github.com/mininet/mininet/commits/master>.

LEDUC, G.; KUBLER, S.; GEORGES, J. . Innovative blockchain-based farming marketplace and smart contract performance evaluation. *Journal of Cleaner Production*, v. 306, 2021. Cited By :2. Disponível em: <www.scopus.com>.

LEKHASY, S. *Simple Demo Blockchain*. 2021. Disponível na Internet em <https://tools.superdatascience.com>.

LI, H. et al. Blockchain-based data preservation system for medical data. *Journal of Medical Systems*, v. 42, p. 141, 2018. Disponível em: <https://doi.org/10.1007/s10916-018-0997-3>.

LIN et al. An iot-based smart system with an mqtt broker for individual patient vital sign monitoring in potential emergency or prehospital applications. *Hindawi*, v. 2022, n. 7245650, p. 2090–2840, jan 2022.

MADDEN, S. *Intel Lab Data*. 2004. Disponível na Internet em <http://db.csail.mit.edu/labdata/labdata.html>.

MAHMUD, R.; RAMAMOHANARAO, K.; BUYYA, R. Application management in fog computing environments: A taxonomy, review, and future directions. *ACM Computing Surveys*, ACM New York, NY, USA, v. 55, n. 4, p. 1–35, 2023.

MAJEED, R. et al. An intelligent, secure, and smart home automation system. *Scientific Programming*, Hindawi, v. 2020, 2020.

MOHAMMADI, M. et al. Deep learning for iot big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, IEEE, v. 20, n. 4, p. 2923–2960, 2018.

N-ABLE, B. *SHA-256 Algorithm Overview*. 2019. Disponível na Internet em <https://www.n-able.com/blog/sha-256-encryption#:~:text=The%20SHA%2D256%20algorithm%20is,that%20is%20256%20bits%20long.> Software available from tensorflow.org.

NAUMAN, A. et al. Multimedia internet of things: A comprehensive survey. *IEEE Access*, IEEE, v. 8, p. 8202–8250, 2020.

NESHENKO, N. et al. Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations. *IEEE Communications Surveys & Tutorials*, IEEE, v. 21, n. 3, p. 2702–2733, 2019.

NGUYEN, D. C. et al. Blockchain for 5g and beyond networks: A state of the art survey. *Journal of Network and Computer Applications*, v. 166, 2020. Cited By :55. Disponível em: <www.scopus.com>.

NGUYEN, T.; NGUYEN, H.; NUGUYEN GIA, T. Exploring the integration of edge computing and blockchain iot: Principles, architectures, security, and applications. *Journal of Network and Computer Applications*, v. 226, p. 103884, 2024. ISSN 1084-8045. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1084804524000614>.

NOVO, O. et al. Capillary networks - bridging the cellular and iot worlds. In: *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. [S.l.: s.n.], 2015. p. 571–578.

PATIL, A. S. et al. Efficient privacy-preserving authentication protocol using PUFs with blockchain smart contracts. *Computers & Security*, v. 97, p. 101958, 2020. Cited By: 8. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167404820302157>.

PONGNUMKUL, S.; SIRIPANPORNCHANA, C.; THAJCHAYAPONG, S. Performance analysis of private blockchain platforms in varying workloads. In: *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. [S.l.: s.n.], 2017. p. 1–6.

POWERS, D. M. W. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020. Disponível em: <https://arxiv.org/abs/2010.16061>.

PRAZERES, C.; SERRANO, M. Soft-iot: Self-organizing fog of things. In: *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. [S.l.: s.n.], 2016. p. 803–808. Cited By :36.

ROSEN, R. et al. About the importance of autonomy and digital twins for the future of manufacturing. *IFAC-PapersMisc*, v. 48, n. 3, p. 567–572, 2015. ISSN 2405-8963. 15th IFAC Symposium onInformation Control Problems inManufacturing. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2405896315003808>.

SAURABH, S.; DEY, K. Blockchain technology adoption, architecture, and sustainable agri-food supply chains. *Journal of Cleaner Production*, v. 284, 2021. Cited By :19. Disponível em: <www.scopus.com>.

SHAN, Y.; MAI, Y. Research on sports fitness management based on blockchain and internet of things. *Eurasip Journal on Wireless Communications and Networking*, v. 2020, n. 1, 2020. Cited By :3.

SHUKLA, A. et al. Digital twin-based prediction for cnc machines inspection using blockchain for industry 4.0. In: *IEEE International Conference on Communications*. [S.l.: s.n.], 2021.

SILVA, E. N.; BRENNO. *SilvaEngComp/RFoT: RFoT_2024_1*. Zenodo, 2024. Disponível em: <https://doi.org/10.5281/zenodo.11450536>.

SITTóN-CANDANEDO, I. et al. A review of edge computing reference architectures and a new global edge proposal. *Future Generation Computer Systems*, v. 99, p. 278–294, 2019. ISSN 0167-739X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167739X1930264X>.

SULTAN, A. B. M.; MEHMOOD, S.; ZAHID, H. Man in the middle attack detection for mqtt based iot devices using different machine learning algorithms. In: *2022 2nd International Conference on Artificial Intelligence (ICAI)*. [S.l.: s.n.], 2022. p. 118–121.

SYED, A. S. et al. Iot in smart cities: A survey of technologies, practices and challenges. *Smart Cities*, MDPI, v. 4, n. 2, p. 429–475, 2021.

THARWAT, A. Classification assessment methods. *Applied Computing and Informatics*, Emerald Publishing Limited, v. 17, n. 1, p. 168–192, 2021.

TIINSIDE. *Ataques às Cidades Inteligentes e Prisão de Mais de 400 Hackers: Relatório Relaciona Principais Eventos no Mundo*. 2024. Acessado em: dia mês ano. Disponível em: <https://tiinside.com.br/03/04/2024/ataques-as-cidades-inteligentes-e-prisao-de-mais-de-400-hackers-relatorio-relaciona-principais-eventos-no->.

TRIPATHI, G.; AHAD, M. A.; PAIVA, S. S2hs- a blockchain based approach for smart healthcare system. *Healthcare*, v. 8, n. 1, 2020. Cited By :24.

VISA, S. et al. Confusion matrix-based feature selection. In: *MAICS*. [s.n.], 2011. v. 710, p. 120–127. Disponível em: <https://www.researchgate.net/publication/260385dcf_Confusion_Matrix-based_Feature_Selection>.

WILLMOTT, C. J.; MATSUURA, K. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, Inter-Research Science Center, v. 30, n. 1, p. 79–82, 2005.

WOO, J. et al. *An Overview of State-of-the-Art Technologies for Data-Driven Construction*. Springer, Cham, 2021. 1323-1334 p. (Lecture Notes in Civil Engineering, v. 98). Cited By :3. Disponível em: <www.scopus.com>.

WOOD, G. et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, v. 151, n. 2014, p. 1–32, 2014.

WU, B. et al. Battery digital twins: Perspectives on the fusion of models, data and artificial intelligence for smart battery management systems. *Energy and AI*, v. 1, p. 100016, 2020. ISSN 2666-5468. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2666546820300161>.

YU, J. et al. Blockchain-based decision tree classification in distributed networks. *Intelligent Automation and Soft Computing*, v. 29, n. 3, p. 713–728, 2021. Disponível em: <https://www.techscience.com/iasc/v29n3/43036>.

ZHENG, Z. et al. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, Inderscience Publishers (IEL), v. 14, n. 4, p. 352–375, 2018.

ZHOU, Y. A solution for internet of things based on blockchain and edge computer. In: MACINTYRE, J.; ZHAO, J.; MA, X. (Ed.). *The 2020 International Conference on Machine Learning and Big Data Analytics for IoT Security and Privacy*. Cham: Springer International Publishing, 2021. p. 264–270. ISBN 978-3-030-62746-1. Disponível em: <https://link-springer-com.ez10.periodicos.capes.gov.br/chapter/10.1007/978-3-030-62746-1_39>.

ZYSKIND, G.; NATHAN, O.; PENTLAND, A. Decentralizing privacy: Using blockchain to protect personal data. In: *Proceedings of the 2015 IEEE Security and Privacy Workshops (SPW)*. San Jose, CA, USA: [s.n.], 2015.

ÖZTüRK, M. et al. *Livestock welfare by means of an edge computing and iot platform*. Springer International Publishing, 2021. 156-165 p. (Advances in Intelligent Systems and Computing, v. 1239 AISC). Disponível em: <www.scopus.com>.