UNISINOS

Programa de Pós-Graduação em

Computação Aplicada

Doutorado Acadêmico

Alexandre Luis de Andrade

VSAC: UM FRAMEWORK DE COMPRESSÃO ADAPTATIVA PARA DADOS DE SAÚDE

São Leopoldo, 2025

Alexandre Luis de Andrade

VSAC: UM FRAMEWORK DE COMPRESSÃO ADAPTATIVA PARA DADOS DE SAÚDE EM CIDADES INTELIGENTES

Tese apresentada como requisito parcial para a obtenção do título de Doutor pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos — UNISINOS

Orientador:
Prof. Dr. Rodrigo da Rosa Righi

São Leopoldo
2025

A554v Andrade, Alexandre Luis de.
VSAC : um framework de compressão adaptativa para dados de saúde em cidades inteligentes / por Alexandre Luis de Andrade. – 2025.
124 f. : il. ; 30 cm.

Tese (doutorado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, RS, 2025.
"Orientador: Dr. Rodrigo da Rosa Righi".

1. Internet das coisas (IoT). 2. Sinais vitais. 3. Compressão de dados adaptativa. 4. Compressão de dados híbrida. 5. Cidades inteligentes. 6. Saúde. I. Título.

CDU: 004.9:614

Catalogação na Publicação (CIP):
Bibliotecário Alessandro Dietrich - CRB 10/2338

(Esta folha serve somente para guardar o lugar da verdadeira folha de aprovação, que é obtida após a defesa do trabalho. Este item é obrigatório, exceto no caso de TCCs.)

RESUMO

A crescente adoção de dispositivos vestíveis e de tecnologias digitais aplicadas à saúde tem reforçado a demanda por mecanismos eficientes de transmissão e armazenamento de dados fisiológicos em sistemas de monitoramento em larga escala. Nesse contexto, um dos principais desafios está na capacidade de gerenciar grandes volumes de dados dos sinais vitais — como frequência cardíaca, frequência respiratória, temperatura corporal e pressão arterial — gerados continuamente por dispositivos heterogêneos, garantindo a integridade clínica, baixa latência e o uso otimizado da rede. As soluções atuais frequentemente tratam as técnicas de compressão de forma isolada e carecem de mecanismos adaptativos que respondam às condições clínicas de cada indivíduo monitorado. Esta pesquisa busca preencher essa lacuna por meio de uma abordagem integrada e sensível ao contexto para a compressão de dados vitais em infraestruturas de cidades inteligentes.

O objetivo deste estudo foi desenvolver e validar o *framework* VSAC (Vital Sign Adaptive Compressor), um *framework* de compressão adaptativa para a transmissão de sinais fisiológicos, combinando técnicas de compressão com e sem perdas. O *framework* foi projetado para ajustar dinamicamente seus parâmetros de compressão com base no tipo de sinal e na prioridade clínica, garantindo o equilíbrio entre a fidelidade dos dados e a eficiência de transmissão.

A metodologia envolveu o desenvolvimento de um protótipo funcional em Python, com compressão em dois estágios, aplicado a conjuntos de dados reais de frequência cardíaca coletados por dispositivos vestíveis com diferentes cadências de amostragem. A avaliação foi realizada em três cenários operacionais — apenas compressão sem perdas, apenas compressão com perdas e compressão híbrida — utilizando conjuntos de dados com diferentes tamanhos e densidades. As métricas adotadas foram: taxa de compressão, tempo de compressão e distorção do sinal, com testes conduzidos para assegurar a reprodutibilidade e a robustez dos resultados.

Os resultados confirmaram que a estratégia híbrida e adaptativa proposta pelo *framework* VSAC superou os métodos estáticos tradicionais. O *framework* alcançou taxas de compressão de até 46,3% superiores às abordagens apenas sem perdas, mantendo níveis de distorção inferiores a 10% na maioria dos casos, especialmente em conjuntos de dados médios e grandes. O tempo de compressão também foi significativamente reduzido com o uso do LZW no estágio final, particularmente para arquivos de maior porte.

Conclui-se que a abordagem híbrida e adaptativa de compressão, incorporada no *framework* VSAC, oferece uma solução robusta e escalável para a transmissão eficiente de sinais vitais em ambientes de saúde inteligentes. Seu mecanismo sensível ao contexto permite a priorização em tempo real e o uso eficiente dos recursos de rede e computação, mostrando-se uma ferramenta promissora para políticas públicas de saúde baseadas em infraestruturas conectadas e decisões clínicas oportunas em cidades inteligentes.

Palavras-chave: Internet das Coisas. Sinais vitais. Compressão de dados adaptativa. Compressão de dados híbrida. Cidades inteligentes.

ABSTRACT

The increasing adoption of wearable devices and smart health technologies has highlighted the need for efficient transmission and storage of physiological data in urban-scale health monitoring systems. In this context, one of the major challenges is the ability to manage large volumes of vital-sign data—such as heart rate, respiratory rate, body temperature, and blood pressure—generated continuously by heterogeneous devices, while ensuring clinical integrity, low latency, and optimized network usage. Current solutions often treat compression techniques in isolation and lack adaptive mechanisms responsive to the clinical condition of each monitored individual. This research addresses that gap by proposing an integrated and context-aware approach for compressing vital-sign data in smart city infrastructures.

This study aimed to develop and validate the VSAC (Vital Sign Adaptive Compressor), an adaptive *framework* for managing the transmission of physiological signals using a combination of lossy and lossless data compression techniques. The *framework* was designed to dynamically adjust its compression parameters based on the signal type and clinical priority, ensuring a balance between data fidelity and transmission efficiency.

The methodology consisted of designing and implementing a two-stage compression prototype in Python, applied to real-world datasets of heart rate collected from wearable devices with different sampling intervals. The evaluation included three operational scenarios—lossless-only, lossy-only, and hybrid compression—tested across datasets of varying sizes and densities. Performance was measured using standard metrics: compression rate, compression time, and distortion. The tests were repeated ten times per scenario to ensure measurement stability.

The results confirmed that the hybrid and adaptive strategy proposed by VSAC outperformed traditional static methods. The *framework* achieved compression rates up to 46.3% higher than lossless-only approaches while maintaining distortion levels below 10% in most cases, especially in medium and large datasets. The compression time was significantly reduced with the use of LZW in the final stage, particularly for large files.

The study concludes that the adaptive and hybrid compression approach embodied in VSAC offers a robust and scalable solution for the efficient transmission of vital-sign data in smart healthcare environments. Its context-aware mechanism enables real-time prioritization and efficient resource usage, making it a promising tool for public health policies that rely on connected infrastructures and timely clinical decision-making in smart cities.

Keywords: Internet of Things. Vital Signs. Adaptive data compress. Hybrid data compress. Smart cities.

LISTA DE FIGURAS

Figura 1: Previsão de crescimento do mercado global de IoT (bilhões de dispositivos IoT conectados) 24

Figura 2: Exemplo de painel de acompanhamento de saúde pública em tempo real para monitorar e consolidar várias fontes de dados de sinais vitais. Este painel foi desenvolvido no grupo de pesquisa do autor e orientador. Podemos analisar regiões geográficas críticas em uma cidade inteligente de acordo com as medições de muitos indivíduos e seus sinais vitais, permitindo que o governo e os hospitais atuem de forma proativa na implantação de iniciativas de saúde.) 25

Figura 3: Exemplo de um processo de coleta e envio de dados de sinais vitais através de sensores em uma rede WBAN. 34

Figura 4: Comparação entre estratégias tradicionais (a) e descentralizadas (b) para monitoramento remoto de saúde baseado em IoT. Em (c), está uma visão geral da arquitetura baseada em árvore com base na estrutura (b), onde os serviços de saúde podem ser executados na *Edge, Fog e Cloud*. Em (b) e (c), os aplicativos do usuário podem acessar os serviços de saúde de qualquer lugar. Além disso, em (c), temos uma arquitetura muito próxima da distribuição geográfica de uma Cidade Inteligente, com quarteirões, bairros e comunidades (permitindo consultar dados de um bairro, por exemplo, consultando apenas um determinado nó de *Fog*). 40

Figura 5: Cenários possíveis de transmissão e armazenamento da informação, onde (1) não aplica técnica de compressão; em (2) os dados são compactados e então transmitidos para serem armazenados e em (3) os dados são comprimidos e então transmitidos, contudo, são descompactados antes de serem armazenados. 41

Figura 6: Dois principais métodos de compressão, Lossless e Lossy. O método Lossy resulta em maior taxa de compressão, contudo os dados restaurados a partir da descompressão não correspondem aos do arquivo original. 43

Figura 7: Em (a) está apresentado o algoritmo de compressão Boxcar/Backslope, onde o ponto A é o último ponto armazenado, o ponto B viola apenas o Backslope, o ponto C viola apenas o Boxcar e o ponto D viola os dois critérios e portanto é armazenado. Em (b) está o algoritmo Swinging Door Trending, com uma área de cobertura entre o último valor armazenado (A) e o último valor recebido (F), nesse caso os valores B, C e D serão descartados, enquanto o ponto E será armazenado. 44

Figura 8: Visão arquitetural da solução através da combinação de Internet das Coisas, *Fog Computing* e *Cloud Computing*. 46

Figura 9: Arquitetura em três camadas com destaque para os nós da *Fog* adotando uma arquitetura em árvore. Fonte: [Rodrigues et al., 2023]. 47

Figura 10: Sequência do processo de filtros para seleção dos artigos. 50

Figura 11: Ilustração do caso de uso do *framework* VSAC no qual um cidadão caminha utilizando um *smartwatch* que monitora continuamente seus sinais vitais. (1) Os dados coletados são transmitidos em uma estação de base próxima. (2) Ao chegar à camada *Edge*, os sinais são processados pelo *framework* VSAC, que aplica compressão adaptativa com base na criticidade clínica da informação. (3) Os dados comprimidos são então encaminhados à camada *Fog–Cloud*, onde são armazenados e analisados por um Sistema de Monitoramento de Saúde. (4) Esse sistema aplica um score de alerta clínico (como o NEWS2) e envia uma notificação de retorno ao *framework* VSAC para ajustar dinamicamente os parâmetros de compressão. (5) Em situações de emergência, os dados são enviados (6) para VSAC e, ao verificar a prioridade, notifica o *framework* (7) e pode também direcionar alertas (8) em tempo real para centros médicos ou unidades de atendimento. 62

Figura 12: *framework* de compressão de dados VSAC desenvolvido na camada *Edge*, recebe dados dos sensores distribuídos, processa os dados conforme notificações recebidas do sistema de monitoração desenvolvido na camada *Cloud*. . 66

Figura 13: O *framework* VSAC em destaque a partir da camada *Edge*, onde estão posicionados os *Edge Controllers* no Projeto MHD. Os componentes que formam o *framework* integram um processo em que a informação dos sinais vitais é recebida em **Adaptive Compress**, responsável pelo primeiro estágio de compressão adaptativo, que irá executar a redução de dados conforme parâmetros de **SetUp**. Este componente é ajustado conforme as notificações recebidas a partir do sistema de monitoramento em **Notifiction Manager**. Após a compactação dos dados em **Adaptive Compress**, são enviados para *Assemble*, que irá agrupá-los em pacotes e disponibilizar em formato de texto ASCII para **Format File**, responsável pela formatação dos arquivos. Os arquivos agora em formato JSON seguem para o último estágio de compressão em **Output Compress**, para então serem transmitidos enviados para o sistema de saúde, onde serão restaurados, armazenados e utilizados para diagnóstico, sinalizações e monitoração. 70

Figura 14: Formato de um pacote recebido de uma estação de coleta de dados com as *strings* contendo os sinais vitais de um grupo de pessoas monitoradas. Sendo STX o caractere indicador de início do arquivo, $Base$ o identificador da estação de coleta de dados e em ID_i estão as *strings* com as informações de saúde das pessoas monitoradas. Cada *string* contendo os sinais vitais de uma pessoa monitorada tem como finalizador o caractere ETX 72

Figura 15: Módulo de compressão adaptativa, *Adaptive Compress* recebe pacotes Pck_{in} originados dos sensores e são processados no módulo *Extractor*. Aqui, são separados de cada *ID* apenas os dados de sinais vitais e enviados para serem compactados no *SDT* de acordo com a parametrização de *SetUp*. Os dados compactados resultantes desta etapa são então reagrupados em *Integrator* no pacote original, que havia sido previamente separado em *Extractor*. 73

Figura 16: Fluxograma para o processo de decisão de ajuste dos parâmetros do módulo **SDT**. 74

Figura 17: Módulo **SDT** atuando em tempo de execução com diferentes valores de parâmetros $desvio$ e Tm. Em **(a)** os parâmetros estão com valor *default* e **(b)** os valores foram ajustados conforme notificação. No exemplo acima, para o cenário **(b)**, a notificação recebida em **SDT** faz com que os valores dos parâmetros sejam ajustados para uma nova janela de compressão, com área menor que em **(a)**, levando a uma taxa de compressão maior. 75
Figura 18: Os pacotes de dados originados de **Assemble** são recebidos no módulo **Format File** para serem formatados para o padrão JSON. Na figura está exemplificado um arquivo de saída contendo coletas de dados de duas estações base, identificadas através de "base_id". O arquivo é então enviado para o módulo **Output Compress**, que consiste na última etapa do *framework*. Nesse módulo os dados serão compactados através dos algoritmos *lossless* LZW ou Huffman Coding, para então serem enviados para a *Cloud* ao sistema de monitoramento de saúde. 77
Figura 19: Fluxograma para o processo de decisão na entrada do módulo **Output Compress**. Primeiramente verifica prioridade do arquivo, comparando $FileRate$ com o valor estabelecido para o *framework* no parâmetro P_H. Prioridade baixa é direcionado para o método LZW de compressão. Para prioridade alta, mais uma verificação é feita, o tamanho $FileSz$ é comparado com o tamanho limite estabelecido pelo *framework*, T_H, se estiver no limite inferior será direcionado para a compressão Huffman, porém, se o tamanho exceder o limite definido, será utilizado LZW. 78
Figura 20: Representação dos algoritmos ao longo do fluxo de dados, destacando sua distribuição nos módulos *Adaptive Compress*, *SetUp* e *Assemble*. 84
Figura 21: Infraestrutura do protótipo para validação do *framework* com duas etapas formadas por um módulo adaptativo com algoritmo de compressão com perdas e o segundo módulo composto por dois algoritmos de compressão sem perdas. 87
Figura 22: Gráfico em escala logarítmica que apresenta as propriedades dos seis *datasests* utilizados para validação do protótipo de acordo com a quantidade de leituras (aquisição de dados) e respectivo tamanho. 89
Figura 23: Estrutura para o cenário de testes contemplando apenas algoritmos sem perdas, Huffman Coding e LZW. 94
Figura 24: Estrutura para o cenário de testes de um processo de compressão apenas algoritmos com perdas, no qual cada arquivo de entrada será testado com os desvios de 2, 3, 4 e 5. 95
Figura 25: Estrutura para o cenário de testes contemplando dois métodos de compressão em sequencia, sendo que no segundo estágio estão os algoritmos sem perdas, Huffman Coding e LZW. 96
Figura 26: Desempenho comparativo dos algoritmos Huffman e LZW no Cenário 1. À esquerda (a), observa-se a variação da Taxa de Compressão em função do tamanho dos arquivos de entrada, demonstrando que Huffman é mais eficiente em arquivos pequenos, enquanto LZW se destaca em arquivos maiores. À direita (b), o gráfico mostra o Tempo de Compressão para ambos os algoritmos, destacando a superioridade de LZW em termos de velocidade, acentuando-se à medida que o volume de dados aumenta. 99

Figura 27: Gráfico comparando os resultados da Taxa de Compressão nos diferentes cenários de teste. Se verifica o melhor desempenho do cenário que emula os dois estágios de compressão conforme o *framework* VSAC. 102
Figura 28: Relação entre a taxa de distorção e o $desvio$. Para o mesmo arquivo de entrada contendo dados de frequência cardíaca, com 36 kB, em (a) com um desvio de 2, o resultado é um PRD de 50% e em (b) com um desvio de 5, o PRD resulta em 56%. 103
Figura 29: PRD para os seis arquivos de entrada com diferentes tamanhos de arquivo, de acordo com a variação do parâmetro $desvio$. 104
Figura 30: Desempenho da Taxa de Compressão (TC) versus Taxa de Distorção (PRD) para cada conjunto de dados. 105

LISTA DE TABELAS

Tabela 1: Valores de referência de sinais vitais para EWS, onde: BRP = Taxa de Respiração, SpO2 =Saturação Periférica de Oxigênio, TC = Temperatura Corporal, FC= Frequência Cardíaca, PS = Pressão Sanguínea Diástole, SO = Suplemento de Oxigênio 37

Tabela 2: Nível de criticidade da saúde do paciente. 37

Tabela 3: Trabalhos relacionados 50

Tabela 4: Trabalhos relacionados analisados de acordo com o problema a ser resolvido, abordagem adotada técnica desenvolvida para obter o resultado. 55

Tabela 5: Análise dos trabalhos relacionados quanto aos objetivos e técnicas utilizadas. 58

Tabela 6: Possibilidades de aquisição de dados de sinais vitais. 65

Tabela 7: Comparativo de desempenho dos métodos Huffman Coding e LZW para diferentes tamanhos de arquivo texto, sendo TC= Taxa de Compressão, T= Tempo de Compressão e TD= Tempo de Descompressão. 67

Tabela 8: Módulos que compõem o *framework* com as respectivas entradas e saídas. . . 71

Tabela 9: Campos que formam a *string* de dados com as informações de sinais vitais coletados de uma pessoa monitorada. 73

Tabela 10: Formato do *header* do arquivo resultante da saída de **Assemble**. 76

Tabela 11: Nomenclaturas utilizadas para os atributos do arquivo de saída do módulo **Format File**. 76

Tabela 12: Principais parâmetros utilizados no *framework*, classificados quanto ao tipo. Os parâmetros podem ser estáticos ou se modificar em tempo de execução, permitindo o comportamento adaptativo do *framework*. 78

Tabela 13: Correspondência entre algoritmos conforme módulo em que está inserido, sua função principal e o tipo de operação. 83

Tabela 14: Principais métricas adotadas nos trabalhos relacionados. 91

Tabela 15: Tipos de arquivos utilizados para o teste do *framework*, com medidas de cada indivíduo durante um período de 24 horas. 92

Tabela 16: Resultado do cenário 1 de testes, aplicando apenas algoritmos de compressão sem perdas, tendo como entrada os arquivos de diferentes tamanhos. Os resultados deste teste colocam lado a lado os resultados de LZW e Huffman Coding. 94

Tabela 17: Resultado do cenário 2 de testes, aplicando apenas algoritmo de compressão com perdas (Swinging Door Trending), foram testadas diferentes configurações de taxa de compressão, através do parâmetro $desvio$. 95

Tabela 18: Resultado do cenário 3 de testes, contemplando os dois estágios de compressão do *framework*. Para o primeiro estágio, com algoritmo com perdas (Swinging Door Trending), foram testadas diferentes configurações de taxa de compressão, através do parâmetro $desvio$. No segundo estágio são utilizados algoritmos sem perdas, sendo LZW e Huffman Coding, de acordo com o tamanho dos arquivos. No final, uma tabela denominada *Output* apresenta a Taxa de Compressão total de cada configuração e o tempo total despendido nos dois estágios. 97

Tabela 19: Tabela comparativa com os três cenários de testes aplicados no protótipo. Os valores em destaque correspondem aos melhores resultados de cada cenário, sendo em destaque os do cenário 3 que foram o que retornou as melhores taxas de compressão para todos os arquivos. 102
Tabela 20: Síntese comparativa dos cenários de compressão aplicados ao sistema de monitoramento de saúde, com destaque para o cenário híbrido. 103
Tabela 21: Taxas de distorção para seis arquivos de entrada de acordo com a variação do desvio do parâmetro. 104
Tabela 22: Tomando a Tabela 4 como referência, esta tabela apresenta as propriedades e características do VSAC em comparação com os trabalhos relacionados analisados. 107

LISTA DE SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
FAPERGS	Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul
IOT	Internet Of Things
ML	Machine Learning
IDC	International Data Corporation
ZB	Zettabytes
IoHT	Internet of Health Things
HPC	High-Performance Computing
VSAC	Vital Sign Adaptive Compressor
WBAN	Wireless Body Area Network
SBC	Single Board Computing
BPM	Batimentos Por Minuto
ECG	Eletrocardiograma
BRPM	Breath Per Minute
BRPM	Breath Per Minute
VFC	Variação da Frequência Cardíaca
EWS	Early Warning Score
QoS	Quality of Service
LZW	Lempel-Ziv-Welch
DC	Desvio de Compressão
MHD	Minha História Digital
ICOLAB	Instituto Colaborativo de Blockchain
CS	Compressive Sensing
WSN	Wireless Sensor Network
WLAN	Wireless Local Area Network
TC	taxa de compressão
RLE	Run-Length Encode
CNN	Convolutional Neural Network
JSON	Java Script Object Notation
SDT	Swinging Door Trending
MB	Megabyte
kB	Kilobyte

SUMÁRIO

AGRADECIMENTOS **21**

1 INTRODUÇÃO **23**
1.1 Motivação 26
1.2 Hipótese e Questão de Pesquisa 28
1.3 Objetivos 30
1.4 Organização do texto 31

2 FUNDAMENTAÇÃO TEÓRICA **33**
2.1 Monitoramento remoto de sinais vitais 33
2.1.1 Tipos e Propriedades de Sinais Vitais 34
2.2 Arquitetura Sistema de Monitoramento de Saúde 37
2.3 Compressão de Dados 39
2.3.1 Tipos de compressão de dados 42
2.3.2 Combinando técnicas de compressão de dados 44
2.4 Projeto MHD 45

3 TRABALHOS RELACIONADOS **49**
3.1 Metodologia de seleção dos trabalhos relacionados 49
3.2 Descrição do Estado da Arte 51
3.2.1 Adaptação da Compressão ao Contexto e aos Recursos Computacionais Disponíveis 51
3.2.2 Uso de Estratégias Híbridas que Combinam Métodos Com e Sem Perdas . 52
3.2.3 Integração da Compressão com Mecanismos Analíticos e Arquiteturas Distribuídas do Tipo Edge–Fog–Cloud 53
3.3 Análise e oportunidades 54

4 *FRAMEWORK* VSAC **61**
4.1 Caso de Uso 61
4.2 Decisões de Projeto 63
4.2.1 Métodos de compressão 63
4.2.2 Aquisição de dados 64
4.2.3 Arquitetura e funcionamento do *framework* 64
4.2.4 Formatação dos dados 66
4.2.5 Adoção de Compressão Adaptativa em Cenários Reais 68
4.2.6 Síntese das decisões de projeto e aplicabilidade 68
4.3 Arquitetura VSAC 69
4.3.1 Visão geral 69
4.3.2 Visão detalhada 72
4.4 Algoritmos e operação 79

5 METODOLOGIA E AVALIAÇÃO **85**
5.1 Protótipo 86
5.2 Cargas de teste 87
5.3 Métricas 88
5.4 Cenários de teste 90

6 RESULTADOS E ANÁLISE **93**
6.1 Ensaios 93
6.1.1 Cenário 1: Sem perdas 93
6.1.2 Cenário 2: Com perdas 94
6.1.3 Cenário 3: Híbrido 95
6.2 Análise dos Resultados 96
6.2.1 Avaliação da Taxa de Compressão e Tempo de Compressão 96
6.2.2 Avaliação da Taxa de Distorção do Sinal 102
6.2.3 Comparação de Resultados do VSAC 107
6.3 Discussão dos Resultados 107

7 CONCLUSÃO **109**
7.1 Análise da Hipótese 110
7.2 Contribuições 111
7.3 Limitações 112
7.4 Publicações 112
7.5 Trabalhos Futuros 113

REFERÊNCIAS **115**

"Buscai primeiro o reino de Deus, e a sua justiça, e todas estas coisas vos serão acrescentadas."
Mateus 6:33

AGRADECIMENTOS

A Deus, por ter guiado meus passos durante toda a realização deste trabalho, concedendo-me as graças necessárias, a sabedoria para tomar as decisões corretas e a força para perseverar diante dos desafios. A Ele, agradeço pela saúde, pela inspiração e por iluminar meu caminho, permitindo que esta jornada acadêmica fosse concluída com propósito e determinação.

À minha família, base sólida e fonte de inspiração, expresso minha mais profunda gratidão. De forma especial, à minha esposa Graziela, cujo incentivo constante, compreensão e apoio incondicional foram determinantes para a realização deste trabalho. Aos meus pais, pelo exemplo de dedicação e valores transmitidos; e, em especial, à memória de meu pai, que não pôde presenciar este momento, mas que, estou certo, partilha desta conquista em espírito, com alegria e orgulho.

Aos meus companheiros felinos, Galileu e Tiffany, pela silenciosa presença durante as longas horas de escrita e por algumas contribuições inesperadas na forma de caracteres indecifráveis ao atravessarem o teclado.

E, de modo muito especial, ao professor doutor Rodrigo da Rosa Righi, pela orientação exemplar, marcada pelo incentivo, clareza e compromisso. Sua condução inspiradora foi decisiva para transformar esta pesquisa em um trabalho consistente, relevante e concretizado.

1 INTRODUÇÃO

A transformação digital tem impulsionado o uso de tecnologias digitais em diversas áreas. Esse impulso se deu, em especial, pela adoção em larga escala de dispositivos e sensores conectados, conhecidos como Internet das Coisas, ou *Internet of Things* (IoT), juntamente com a evolução das tecnologias de comunicação [Azad et al., 2025, Bhadula and Sharma, 2023]. Esses fatores permitiram que objetos do cotidiano - tais como eletrodomésticos e veículos - e dispositivos da indústria, medicina e agricultura pudessem ser conectados à Internet, integrando o mundo físico ao virtual [Ficili et al., 2025].

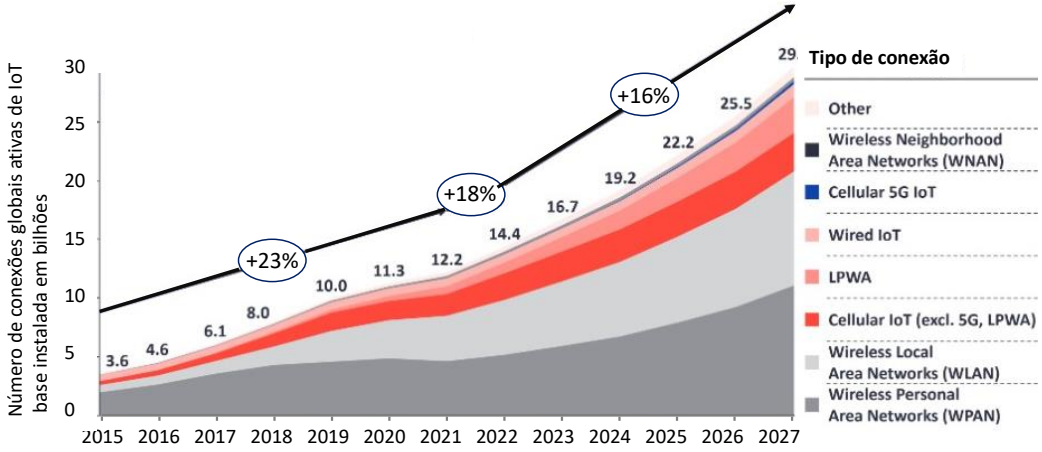
Na medida em que esses sensores permitem coletar as mais diversas informações do entorno, no contexto em que estão inseridos - tais como localização, movimento, temperatura e umidade, eles passam a dotar os objetos de diversas capacidades, tornando-os "inteligentes"[Padmanaban et al., 2023]. Esses objetos, capazes de capturar as informações, também podem transmitir dados através de uma infraestrutura para serem armazenados. Os dados armazenados podem, então, ser analisados a partir de técnicas de *Machine Learning* (ML) e Inteligência Artificial, apoiando tomadas de decisão mais assertivas [Vasanth et al., 2025, Kumar and Suresh, 2023, Lalouani et al., 2021, Lee and Yoon, 2021].

Neste contexto, a indústria tem investido cada vez mais em processos de automação e infraestruturas escaláveis, levando a um aumento exponencial de sensores conectados, conforme destacado na Figura 1. Nesse gráfico, verificamos a projeção de crescimento dos dispositivos para os próximos anos e também o quanto se desenvolveu nos anos recentes. Esse crescimento da base de dispositivos, consequentemente, gera uma grande volumetria de dados. Destaca-se também a popularização dos dispositivos *wearables*, que contribui para a geração de uma grande volumetria de dados. Segundo a International Data Corporation (IDC), em 2025, estima-se que haverá 41,6 bilhões de dispositivos IoT conectados, gerando mais de 79 Zettabytes (ZB) de dados no ano [Lu et al., 2020]. Contudo, ao lidar com tal volumetria, deparamo-nos com dois desafios principais: a transmissão e o armazenamento desses dados.

Quanto à transmissão, devemos considerar que os dispositivos IoT utilizam, basicamente, redes sem fio via rádio frequência para a conexão e a transmissão de dados na rede. Caso a rede não esteja adequadamente dimensionada, isso pode resultar na perda de pacotes e na necessidade de retransmissão. Para sistemas que são sensíveis ao tempo e à qualidade da informação, tais como *healthcare* para monitoramento de sinais vitais, a transmissão dos dados representa um fator crítico para um diagnóstico preciso e para a eficiência no tempo de reação [Idrees and Khlief, 2023].

Ao tratar do armazenamento desses dados, é importante considerar que um dos principais elementos da transformação digital é a computação em nuvem. A computação em nuvem foi um fator chave para a aceleração da transformação, pois trouxe mais agilidade, facilidade de alocação de dados e escalabilidade, atendendo rapidamente às necessidades do negócio [Dwivedi et al., 2022]. Nesse contexto, o armazenamento de dados na nuvem constitui um grande

Figura 1: Previsão de crescimento do mercado global de IoT (bilhões de dispositivos IoT conectados)



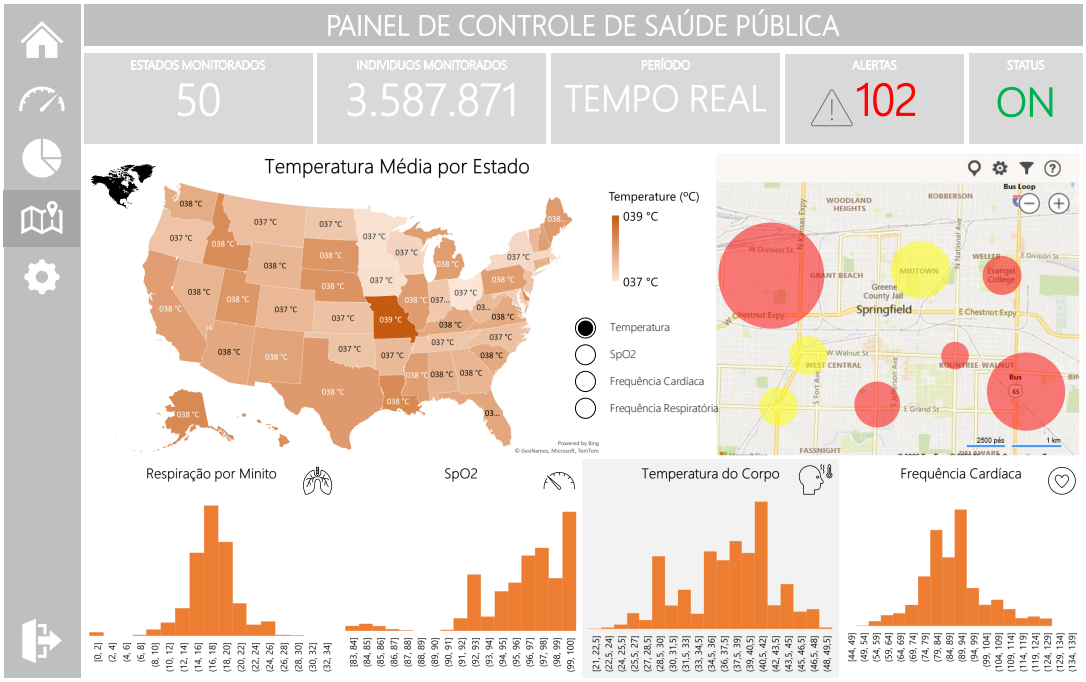
Fonte: Adaptado de https://iot-analytics.com/number-connected-iot-devices/

avanço, devido aos seus vastos recursos de armazenamento e escalabilidade, oferecendo segurança da informação, facilidade de acesso e recuperação em caso de desastres. Contudo, existe um custo financeiro associado que precisa ser considerado, demandando estratégias adequadas de armazenamento e recuperação dos dados [Aggarwal and Lan, 2020].

Entre as áreas que mais vêm se desenvolvendo no uso de dados, destaca-se a área da saúde, sendo esta a que os dados são um fator crítico e determinante para o sucesso. A área da saúde vem passando por um processo de transformação devido à utilização das tecnologias de IoT [Krishna et al., 2023, Renó et al., 2023]. A rápida adoção dessas novas tecnologias vem revolucionando as aplicações de saúde, permitindo uma forma mais personalizada, preventiva e colaborativa de cuidar dos pacientes [Verma et al., 2025]. Nos hospitais, a utilização de diversos sensores que geram informações sobre os sinais vitais dos pacientes está associada ao conceito de *Internet of Health Things* (IoHT). Dispositivos vestíveis de IoT, ou *wearables*, como sensores de temperatura corporal e de movimento, por exemplo, permitem monitorar em tempo real as condições individuais de pacientes em tratamento dentro de hospitais [Rodrigues et al., 2022]. O monitoramento por meio de *wearables* permite avanços significativos na detecção de eventos críticos, o que é possível devido à previsibilidade que se pode obter com o uso dos dados coletados.

Ainda neste contexto, os sistemas de monitoramento de saúde remoto têm passado por um grande processo de aceleração, bastante impulsionado pelas demandas que surgiram durante a COVID-19 [Castiglione et al., 2021]. Tais sistemas se destacam em relação aos sistemas tradicionais de monitoramento de saúde, na medida em que os pacientes podem ser monitorados durante 24 horas através destes sensores, enquanto um médico poderia observar um paciente por um tempo limitado durante o dia. Contudo, problemas críticos de saúde podem ocorrer a qualquer momento [Senthilkumar et al., 2018]. Pacientes assistidos por IoT podem ser acessados pela equipe médica remotamente, e seu estado de saúde pode ser supervisionado continuamente,

Figura 2: Exemplo de painel de acompanhamento de saúde pública em tempo real para monitorar e consolidar várias fontes de dados de sinais vitais. Este painel foi desenvolvido no grupo de pesquisa do autor e orientador. Podemos analisar regiões geográficas críticas em uma cidade inteligente de acordo com as medições de muitos indivíduos e seus sinais vitais, permitindo que o governo e os hospitais atuem de forma proativa na implantação de iniciativas de saúde.)



Fonte: Adaptado de projeto Minha História Digital de autor e orientador

possibilitando antecipar situações de risco à saúde e aplicar as devidas contramedidas [Fischer et al., 2020].

Por exemplo, a Figura 2 ilustra um painel de controle de sinais vitais que agrupa dados de fontes diferentes para fornecer uma visão geral das informações de sinais vitais de regiões distribuídas geograficamente. Esta figura, em particular, usa dados fictícios e representa os esforços de alguns autores para reunir dados de sinais vitais, permitindo notificações baseadas em limites e atualizações online para ajudar o governo a preparar iniciativas públicas de maneira adequada.

Associando estas iniciativas de monitoramento de saúde com pacientes assistidos remotamente ao contexto de cidades inteligentes, surgem muitas oportunidades para inovações técnicas, uso eficiente de recursos e avanços sociais. A ideia de usar redes de sensores baseadas em IoT para aplicações de saúde é promissora e tem o potencial de minimizar ineficiências na infraestrutura existente [Ghazal et al., 2021]. Nesse âmbito, o monitoramento dos sinais vitais da população permite o desenvolvimento de algoritmos que geram oportunidades a partir dos dados por meio da IA, ajudando a fortalecer as políticas públicas e a trazer segurança à população. Ainda, aplicar métodos preditivos e detectar mudanças no comportamento dos sinais vitais ajuda a entender como a dinâmica desses dados é alterada em condições de doença, revelando padrões potenciais de degradação fisiológica [Almeida and Nabney, 2017].

Para alcançar estes objetivos, surgem desafios relativos à aquisição de dados, como métodos e técnicas adequadas para capturar sinais corporais, reuni-los e transportá-los para monitoramento remoto [Mavrogiorgou et al., 2022]. A aquisição e a transferência de dados desempenham um papel crucial em um sistema de monitoramento da saúde. Para fornecer diagnósticos, relatórios e alertas precisos, este sistema depende de dados confiáveis, muitas vezes provenientes de dispositivos heterogêneos e diferentes tipos de ambientes [Mavrogiorgou et al., 2022]. Além disso, explorar técnicas e tecnologias para coletar dados consistentes, combinadas com avanços em sensores leves e tecnologias vestíveis, é uma forma de fornecer eficiência, economia de energia e produtividade [Hinze et al., 2022].

1.1 Motivação

Dessa forma, para possibilitar o avanço de áreas como a nova saúde digital, que faz uso extensivo de sensores e gera um grande volume de dados, é necessário encontrar métodos que mitiguem os fatores que podem degradar o desempenho e a qualidade do diagnóstico médico. Ainda, elaborar um *framework* eficiente de compressão de dados permite o gerenciamento de grandes volumes de dados em tempo real, mantendo a qualidade do serviço e possibilitando a gestão de falhas. O *framework* se apresenta como habilitador de serviços de monitoria de saúde em um contexto de cidades inteligentes [Poongodi et al., 2021], que depende de um *framework* de gestão da coleta de dados distribuídos geograficamente.

No âmbito da saúde, o diagnóstico preditivo constitui um fator chave para poder antecipar e evitar situações de risco que podem ser irreversíveis. Estudos forneceram evidências de que análises quantitativas revelam anormalidades nos sinais vitais ocorrendo em pacientes várias horas antes de um agravamento [Mok et al., 2015]. O processo de análise das informações do paciente depende do recebimento de dados sem atraso, íntegros e com acurácia, permitindo antecipar situações antes que elas aconteçam [Rodrigues et al., 2022]. Aplicar métodos preditivos e detectar mudanças de comportamento ajuda a entender como a dinâmica dos sinais vitais é alterada em condições de doença, revelando padrões potenciais de degradação fisiológica [Almeida and Nabney, 2017].

Embora muitos trabalhos estejam sendo realizados na área de transmissão de dados na saúde, em particular na *Internet of Health Things* (IoHT), a literatura aborda, de modo mais abrangente, diferentes arquiteturas e *middlewares*. Tais abordagens propõem modelos e soluções para enfrentar desafios de baixa latência, ocupação de largura de banda e alto *throughput* [Kashani et al., 2021, Sobin, 2020, Bansal and Kumar, 2020, Krishnamoorthy et al., 2021]. Nesse contexto, a arquitetura mais amplamente adotada em sistemas de monitoramento de saúde é composta por três camadas: *Edge-Fog-Cloud*. Nessa perspectiva de camadas, os dados são tratados nas duas primeiras camadas (*Edge-Fog*), antes de serem enviados para a *Cloud*, reduzindo a quantidade de pacotes na rede.

Contudo, ainda nessa abordagem, verifica-se que os dados que trafegam na *Fog* se bene-

ficiam de uma estrutura de telecomunicações fixa e dedicada, o que otimiza a velocidade de transmissão. Entretanto, há um grande desafio na quantidade de dados que trafegam entre as camadas *Edge-Fog*, onde a comunicação entre essas duas camadas normalmente é lenta [Alharbi et al., 2023, Ahmad et al., 2023, Mattia et al., 2023], sendo muitas vezes sem fio. Portanto, é na transmissão de pacotes da *Edge* para a *Fog* que podem surgir gargalos. Nesse ponto, faz-se necessária uma solução baseada na gestão e otimização de pacotes de dados para não limitar a escala de sensores distribuídos em uma cidade inteligente.

Além de uma abordagem de arquitetura, conforme verificado na pesquisa de trabalhos relacionados no Capítulo 3, outro modo adotado para lidar com a alta volumetria de dados na monitoração de sinais vitais consiste na utilização de técnicas de compressão de dados. Embora menos abundante, essa literatura apresenta diferentes métodos que se apoiam em técnicas de compressão, combinando soluções amplamente adotadas ou propondo modificações em algoritmos já conhecidos para obter um desempenho melhor em situações específicas.

Os trabalhos apresentados em [Hanoune and Lysmos, 2020, Kalaivani et al., 2020, Passos et al., 2020, Lu et al., 2020, Khlief and Idrees, 2022] utilizam técnicas de compressão que, quando combinadas, geram novos métodos para obter mais eficiência na redução dos dados do que seria alcançada apenas com um único método. Nesses casos, foram combinadas técnicas *lossy* e *lossless*, sendo que uma primeira etapa normalmente aplica *lossy*. Com isso, mantém-se o controle da taxa de compressão para não perder informações que possam comprometer a qualidade do resultado. Nesses casos, como estamos tratando de sinais vitais, esse resultado está diretamente relacionado à precisão de um diagnóstico ou alerta de saúde.

Para essas pesquisas que combinam as técnicas de compressão em duas ou mais etapas, no estágio seguinte ao *lossy*, aplica-se um segundo ciclo de redução para ganhar mais eficiência, mas agora sem perdas. Esse processo é assim estruturado, pois a primeira etapa já lidou com as perdas de compressão nos níveis aceitáveis, de acordo com as necessidades de cada sistema em que está sendo aplicado.

Os modelos que combinam técnicas de compressão com e sem perdas, na etapa responsável pela compactação com perdas, frequentemente utilizam processos que mantêm esse módulo atualizado sobre o nível de perda aceitável. Em [Xiao et al., 2018, Azar et al., 2019, Khlief and Idrees, 2022, Amiri et al., 2022], a taxa de compressão do módulo em que se faz a redução de dados com perdas é estabelecida de modo a permanecer dentro dos níveis que não comprometem a qualidade da informação restaurada. Nas propostas, encontram-se casos em que a informação é atualizada em tempo real e em que é feita por meio da configuração de parâmetros antes de executar o processo.

Na literatura pesquisada, percebe-se um equilíbrio em relação à abordagem do tipo de compressão utilizada para sinais vitais. Em [Xiao et al., 2018, Azar et al., 2019, Amiri et al., 2022] utiliza-se a compressão do tipo *lossy*, com controles de tolerância para o nível de perdas. A compressão do tipo *lossless* se verifica em [Das and Rahman, 2021, Idrees et al., 2022, Hossein and Sheikh, 2022, Idrees and Khlief, 2023], e uma solução híbrida em [Hanoune and Lysmos,

2020, Kalaivani et al., 2020, Passos et al., 2020, Lu et al., 2020, Khlief and Idrees, 2022].

Uma visão mais detalhada sobre a literatura na área será apresentada na Seção 3.2. Com base nesta análise, pode-se afirmar que, apesar de existirem vários trabalhos no domínio da compressão de dados para sinais vitais, não foram encontradas abordagens capazes de endereçar as principais necessidades de um sistema de monitoramento de saúde. Mais especificamente, existe uma lacuna de pesquisa sobre como as soluções exploram as estratégias de transmissão e como esse processo pode ser adaptável. Em especial, considerando características intrínsecas dos sinais vitais, tais como o tipo de sinal vital, a periodicidade, a priorização de acordo com a necessidade do paciente e sua localização.

Dessa forma, a motivação científica para o presente trabalho reside na necessidade de propor uma solução para os problemas de transmissão, latência e armazenamento de dados em sistemas de monitoramento na área da saúde, com base em dispositivos de IoT. As soluções encontradas na literatura para esse problema não tratam de modo eficiente a transmissão dos dados dos sinais vitais, considerando suas particularidades e as dos pacientes. Dado este cenário, justifica-se o desenvolvimento de um novo modelo de compressão adaptativo para a transmissão de sinais vitais originados da IoT para sistemas de monitoramento de saúde.

1.2 Hipótese e Questão de Pesquisa

Conforme visto nos parágrafos anteriores, a integridade dos dados constitui um fator crítico para que um sistema de monitoramento de saúde possa prover uma solução com qualidade e acurácia, capaz de oferecer um diagnóstico preciso para tomadas de decisão assertivas. Considerando a alta volumetria de dados — originada não apenas do número de indivíduos monitorados, mas também de diversos dispositivos IoT — e a heterogeneidade dos dados — quanto aos diferentes tipos, características e prioridades dos sinais vitais — várias abordagens surgem como alternativas na literatura. Abordagens como elasticidade computacional [Zhang et al., 2020], balanceamento de carga [Pallewatta et al., 2019], computação de alto desempenho *High-Performance Computing* (HPC) [Rizvi et al., 2022], agregação de mensagens [Rizvi et al., 2022], e compressão de dados [Khlief and Idrees, 2022], que vêm ganhando espaço. De fato, as técnicas de compressão de dados não constituem algo novo; contudo, ressurgem nesse novo contexto como uma alternativa eficiente e de baixa complexidade de implantação [Gia et al., 2019a] [Khlief and Idrees, 2022] [Das and Rahman, 2021]. Assim sendo, para nortear esta pesquisa, a seguinte afirmação define a atual hipótese da tese [Andrade et al., 2024]:

Hipótese. *Um modelo de gestão de dados voltado a aplicações de saúde em cidades inteligentes, que combina técnicas de compressão de dados com perdas e sem perdas por meio de estratégias adaptativas, apresenta melhor desempenho na transmissão eficiente e na preservação da qualidade dos sinais vitais do que modelos que utilizam apenas compressão com perdas ou apenas compressão sem perdas.*

A hipótese propõe que um modelo de compressão de dados que combina técnicas de compressão com perdas e sem perdas, empregando **estratégias adaptativas**, pode **superar em desempenho** abordagens que utilizam exclusivamente compressão com perdas ou apenas compressão sem perdas. Essa hipótese parte do entendimento de que, em sistemas de monitoramento de saúde – especialmente em ambientes distribuídos como cidades inteligentes –, é necessário encontrar um equilíbrio entre a eficiência na transmissão de dados, a preservação da qualidade da informação clínica e o desempenho computacional.

Modelos que aplicam apenas compressão com perdas geralmente alcançam maiores taxas de redução de dados, mas podem comprometer informações clínicas relevantes. Já os que utilizam apenas métodos sem perdas preservam completamente os dados; porém, geram arquivos maiores, dificultando a transmissão em redes com baixa largura de banda. A abordagem híbrida apresentada nesta tese busca o melhor dos dois cenários: utilizar uma compressão inicial com perdas, ajustada dinamicamente conforme a criticidade do paciente e do sinal, seguida de uma compressão sem perdas que assegura a integridade da informação remanescente. Com base na hipótese, foi definida a seguinte questão de pesquisa a ser trabalhada nesta tese:

Quais estratégias poderiam ser adotadas em um processo adaptativo de gestão de compressão de dados de sinais vitais que garantam eficiência na transmissão, de modo que o sistema de monitoramento de saúde consiga prover informações para diagnóstico preditivo e alertas de problemas com as pessoas monitoradas ?

Esta questão de pesquisa proposta envolve a busca por **estratégias adaptativas** no contexto da **gestão de compressão de dados de sinais vitais**, com o objetivo de garantir **eficiência na transmissão** e, ao mesmo tempo, assegurar a qualidade necessária para **o diagnóstico preditivo** em sistemas de monitoramento de saúde.

O termo **adaptativo** refere-se à capacidade do sistema de ajustar dinamicamente seu comportamento de compressão de acordo com o contexto clínico ou operacional. Isso inclui, por exemplo, a variação dos parâmetros de compressão com base na criticidade do estado de saúde do paciente, na qualidade do sinal recebido ou nas condições de rede disponíveis. Um sistema adaptativo não opera de forma fixa ou estática, mas sim como um agente sensível ao ambiente, capaz de se reconfigurar para priorizar informações mais relevantes ou reduzir o volume de dados quando necessário, sem comprometer a integridade clínica.

Já o conceito de **gestão** implica a coordenação inteligente de múltiplos fatores no processo de compressão — como a escolha do algoritmo adequado, a priorização de sinais com maior relevância clínica, o balanceamento entre compressão e latência, e a integração com os mecanismos de decisão do sistema de monitoramento. A gestão eficaz da compressão em tempo real exige um controle preciso sobre os módulos envolvidos, considerando tanto os requisitos técnicos quanto os objetivos clínicos do sistema.

A **eficiência na transmissão** é um requisito central em arquiteturas distribuídas de saúde conectada, onde a largura de banda e a estabilidade da rede podem ser limitadas, especial-

mente na camada *Edge*. Reduzir o volume de dados transmitido sem perder a qualidade do sinal é essencial para garantir a escalabilidade do sistema e permitir o monitoramento contínuo de múltiplos pacientes. A compressão eficiente, nesse contexto, deve ser medida não apenas em termos de taxa de compactação, mas também considerando o tempo de processamento, o consumo energético e a preservação das informações relevantes.

Por fim, o foco em **diagnóstico preditivo e geração de alertas** demanda que o sistema preserve os padrões e variações significativas nos sinais vitais. Mesmo comprimidas, as informações clínicas essenciais — como picos, irregularidades ou tendências anômalas — devem ser mantidas para que os algoritmos de análise e os profissionais de saúde possam identificar riscos iminentes. A compressão adaptativa, portanto, não é apenas uma questão técnica, mas uma parte fundamental de um sistema inteligente que precisa ser confiável para apoiar decisões clínicas em tempo real.

Dessa forma, a questão de pesquisa articula um conjunto de requisitos complexos e interdependentes que exigem soluções inovadoras no cruzamento entre compressão de dados, computação adaptativa, monitoramento de saúde e tecnologias de cidades inteligentes. Portanto, o foco desta pesquisa é identificar qual modelo para a gestão de dados de sinais vitais, com origem na captura de informações em dispositivos IoT da área da saúde e disponibilizados por meio da *Edge Computing*, transmitindo de modo eficiente e com qualidade para que sejam monitorados e gerem alertas para atendimento médico. Essas informações também ficam disponíveis para serem analisadas por meio de ML e IA para buscar tendências e gerar alertas que permitam ações preventivas.

1.3 Objetivos

Com o propósito de disponibilizar um modelo de compressão de dados que seja adaptativo e capaz de oferecer uma transmissão eficiente dos dados de sinais vitais originados de sensores alocados na camada *Edge*, este trabalho propõe o *framework Vital Sign Adaptive Compressor* (VSAC). O VSAC considera as propriedades dos sinais vitais que se deseja monitorar. Para este trabalho, são adotadas a temperatura corporal, a pressão arterial, os batimentos cardíacos por minuto e a frequência respiratória, os quais serão utilizados como base de informação em um sistema de monitoramento da saúde. A escolha desses tipos de sinais vitais deve-se à sua ampla adoção na literatura e por serem considerados básicos para a predição e diagnóstico da saúde de um paciente [Bose and Kumar, 2019] [Bautista et al., 2018]. Estes dados devem ser tratados pelo modelo conforme a prioridade definida para um paciente em particular e também precisam considerar as propriedades de cada tipo de sinal vital, permitindo que sejam exploradas características que possibilitem a otimização de seu envio. Essa habilidade do *framework* deve propiciar a economia no consumo da banda de rede e diminuir o tamanho dos arquivos a serem armazenados para análises futuras. Dessa forma, a seguinte afirmação define o objetivo principal desta pesquisa:

Desenvolver um framework adaptativo para a gestão do envio de dados de sinais vitais para um sistema de monitoramento de saúde que utilize técnicas de compressão de dados como estratégia de otimização da transmissão de arquivos na rede, considerando as propriedades intrínsecas desses sinais e a sinalização de prioridade.

Para completar o objetivo principal, a pesquisa deve atender a objetivos secundários. A definição desses objetivos está alinhada aos desafios enfrentados em sistemas semelhantes, conforme identificado na literatura, bem como aos seus respectivos resultados qualitativos e quantitativos. Assim, os objetivos secundários não apenas complementam o objetivo principal da pesquisa, mas também apoiam o desenvolvimento do modelo e as aplicações necessárias para sua validação.

a) possibilitar que um sistema de monitoramento de saúde apresente alertas de situações de risco;

b) o modelo deve contribuir para a redução do consumo de banda de rede.

c) garantir que a informação esteja íntegra e confiável para que o sinal possa ser restaurado com qualidade;

d) gerar eficiência para armazenamento de dados em uma infraestrutura computacional de cidade inteligente.

e) alcançar uma taxa de compressão superior a 60% nos sinais vitais processados, com taxa de distorção inferior a 10%.

1.4 Organização do texto

O restante da tese está estruturado da seguinte forma: após o capítulo de Introdução, o Capítulo 2 trará a Fundamentação Teórica utilizada para este trabalho, apresentando os conceitos fundamentais necessários à compreensão dos temas abordados por esta pesquisa. Após, o Capítulo 3 trará uma relação de Trabalhos Relacionados a este, destacando uma pesquisa bibliográfica dos trabalhos mais recentes. O Capítulo 4 é central neste documento e descreve o *framework* VSAC. Este capítulo apresentará uma visão preliminar do método de compressão de dados e da estratégia a ser adotada para atingir os objetivos propostos, assim como das decisões de projeto. O Capítulo 5 apresenta a metodologia de avaliação do *framework*, definindo os cenários de testes planejados, juntamente com os parâmetros, a infraestrutura e a carga de trabalho que serão utilizados nos testes, bem como as métricas que serão empregadas na análise do sistema proposto. No Capítulo 6 os resultados dos testes são apresentados e analisados. Por fim, o Capítulo 7 apresenta as considerações finais desta tese, com as contribuições esperadas e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

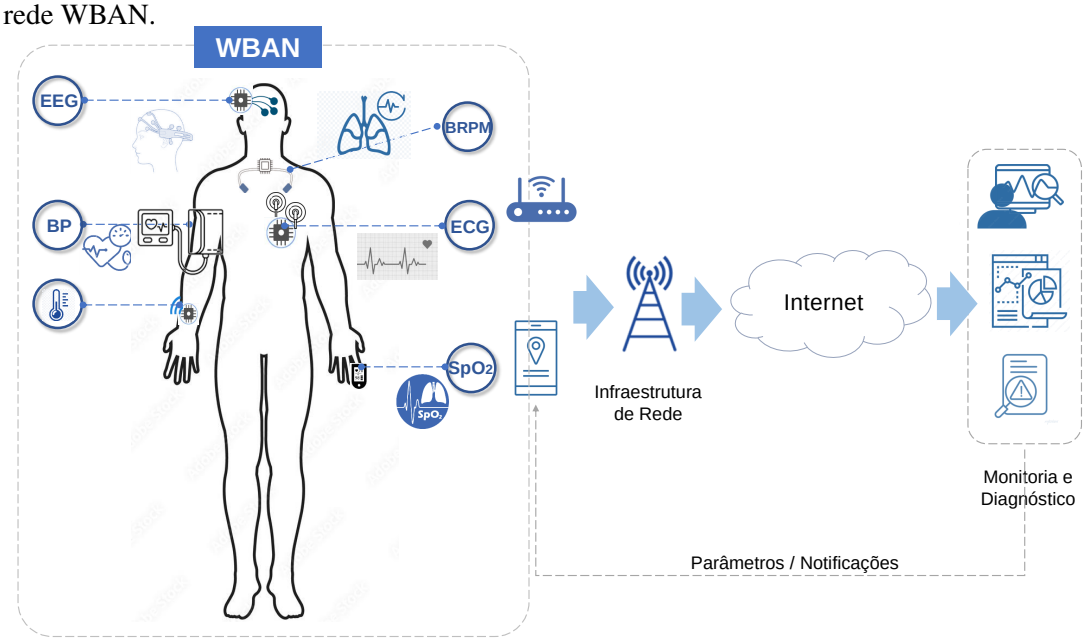
O objetivo deste capítulo é apresentar os conceitos fundamentais para o desenvolvimento deste trabalho, sendo dividido em três seções principais. Primeiramente, na Seção 2.1, serão apresentados os conceitos referentes à aquisição de dados de sinais vitais, abordando como se dá esse processo e sua relação com a IoT, destacando as principais características dos tipos de sinais que serão considerados neste trabalho e como isso pode apoiar a tática utilizada no *framework* proposto. A seguir, na Seção 2.2, serão analisadas as arquiteturas mais adotadas na literatura relacionadas aos Sistemas de Monitoramento de Saúde, nas quais se identificam oportunidades para otimizar a transmissão de dados. Por fim, na Seção 2.3, serão descritas técnicas e conceitos de compressão de dados e as oportunidades de combinação desses.

2.1 Monitoramento remoto de sinais vitais

Na literatura atual, é possível identificar diferentes técnicas e métodos para a aquisição de dados de sinais vitais com a IoT, com os quais busca-se obter um sistema de monitoramento da saúde eficiente, sem limitar as atividades diárias dos pacientes. Além disso, a IoT permitiu que os sistemas de monitoramento de sinais vitais iniciassem uma mudança de paradigma na abordagem de gerenciamento de saúde, passando do diagnóstico reativo ao prognóstico proativo [Sahu et al., 2022a]. Pequenos dispositivos eletrônicos conectados ao corpo podem medir a temperatura corporal, a frequência cardíaca, a pressão arterial, a frequência respiratória, a saturação periférica de oxigênio e a frequência do pulso [Sahu et al., 2022b]. Um exemplo de arquitetura adotada para adquirir esses tipos de dados é a Wireless Body Area Network (WBAN), na qual os sinais do corpo podem ser coletados e transportados para o monitoramento remoto. O padrão IEEE 802.15.6 foi proposto para comunicações mais realistas e precisas em ambientes WBAN [Şen et al., 2021].

A Figura 3 exemplifica esse contexto no qual os dispositivos vestíveis podem coletar os sinais vitais de uma pessoa e disponibilizá-los em um aparelho móvel, tal como *smartphone*, ou dispositivos conhecidos como *Single Board Computing* (SBC) - como Arduino ou Raspberry Pi - que irão agrupar as informações e transmiti-las através de uma infraestrutura de internet. No destino, as informações permanecem disponíveis para monitoramento e diagnóstico. Nesse processo de coleta e envio, os dispositivos SBC e *smartphones* podem atuar como um *gateway*, sendo que o primeiro envia dados para a internet e pode interagir com os sensores que estiverem disponíveis na residência do paciente. Já o *smartphone* acompanha a pessoa que está sendo monitorada onde quer que ela esteja, coletando dados permanentemente, além de estar disponível para receber alertas em tempo real. A qualquer momento, o usuário pode receber notificações sobre seu estado de saúde. Esse fluxo de informação tem como objetivo disponibilizar uma arquitetura proativa, na qual se pode alertar os usuários sobre um eventual problema futuro, com base em diagnóstico preditivo, permitindo que eles busquem tratamento em tempo hábil.

Figura 3: Exemplo de um processo de coleta e envio de dados de sinais vitais através de sensores em uma rede WBAN.



Fonte: Elaborado pelo autor

2.1.1 Tipos e Propriedades de Sinais Vitais

Nesta seção, serão explorados os principais tipos de sinais vitais, suas respectivas características e como se dá a aquisição desses dados no corpo humano, considerando que o monitoramento é essencial para a observação da saúde de uma pessoa. É preciso, contudo, ressaltar que existem limites inferiores e superiores para cada sinal vital, ou seja, limites que uma determinada métrica deve observar. Além disso, esses limites variam de acordo com a idade da pessoa, seu estado de saúde e se possui doenças crônicas, etc. Os sinais vitais aqui discutidos são relevantes para nos fornecer *insights* sobre as sequelas do longo período da pandemia de COVID-19, o que é especialmente importante para pacientes com doenças crônicas. A literatura nos mostra que cinco parâmetros de saúde são sinais vitais essenciais para detectar a evolução de doenças, que são: (i) frequência cardíaca; (ii) temperatura corporal; (iii) saturação periférica de oxigênio (SpO2); (iv) frequência respiratória; e (v) pressão arterial.

a) **Frequência Cardíaca:** mede os batimentos por minuto (BPM) do ciclo cardíaco de uma pessoa. Varia fisiologicamente ao longo do dia em um indivíduo saudável, de acordo com a atividade física, o consumo de alimentos com cafeína e as emoções, por exemplo. Entretanto, a faixa de frequência cardíaca adequada para um indivíduo em repouso é de 50 a 90 BPM, podendo ser menor em pessoas que praticam atividades físicas [Nanchen, 2018]. Segundo [Karjalainen and Viitasalo, 1986], com um aumento de 1°C na temperatura corporal, há aproximadamente um aumento de 8,5 BPM. Os métodos tradicionais para medir a frequência cardíaca incluem o eletrocardiograma (ECG) e a palpação do pulso radial. No entanto, tais métodos, além de dependerem de um profissional para serem avaliados,

também apresentam pouca mobilidade para acesso e dificuldade de monitoramento das atividades diárias. O ECG é um método preciso, mas deve ser realizado em um local específico, pois a medição pelo pulso radial pode ser imprecisa. Assim, as iniciativas atuais buscam explorar diferentes estratégias de fontes bioelétricas para medir a frequência cardíaca com precisão. Bandas inteligentes geralmente usam fotopletismografia (PPG) para medir a frequência cardíaca [Castaneda et al., 2018]. O monitoramento contínuo pode prever algumas patologias, como arritmias, anemia e hipertireoidismo. Assim, a detecção precoce da frequência cardíaca pode auxiliar na percepção de um sinal de gravidade em pacientes, seja em casa ou no hospital.

b) **Temperatura Corporal:** é um dos sinais vitais mais utilizados para entender a saúde de uma pessoa. A temperatura é utilizada principalmente para observar se uma pessoa possui um processo inflamatório, o início de uma doença ou reações de partículas contra vírus ou bactérias [Chow et al., 2020]. Normalmente, uma temperatura corporal acima de 37,3°C caracteriza febre, indicando que o corpo está tentando combater uma doença ou infecção [Zhang et al., 2022]. A temperatura corporal é captada de diferentes maneiras: axilar, oral e retal, usando termômetros tradicionais [Kanegaye et al., 2016]. Soluções recentes utilizam tecnologias que também podem medir a temperatura da pele, uma vez que a temperatura da pele varia frequentemente para regular e estabilizar a temperatura central do corpo.

c) **Saturação Periférica de Oxigênio (SpO2):** o nível de SpO2 mede a porcentagem de oxigênio transportada pelas moléculas de hemoglobina no sangue periférico de um indivíduo. Pode estar diminuída quando ocorre uma infecção, na qual as citocinas inflamatórias impedem que a troca gasosa eficiente ocorra nas membranas respiratórias [Meleveedu et al., 2020]. Taxas abaixo de 95% de sangue oxigenado indicam um alerta de que o indivíduo pode estar começando a ter falta de ar. Os níveis de oxigênio geralmente permanecem na mesma taxa durante todas as atividades diárias. Abordagens padrão para calcular SpO2 usam sinais PPG compostos por sensores de luz vermelha e infravermelha aplicados às extremidades do corpo [Kumar V and Reddy, 2021].

d) **Frequência Respiratória:** este sinal vital analisa a taxa de respiração por minuto, do inglês *Breath Per Minute (BRPM)*. A literatura apresenta que a frequência respiratória esperada está entre 12 e 20 BRPM [Touw et al., 2017]. De fato, uma redução na BRPM pode indicar que uma determinada pessoa tenha alguma doença respiratória. Isso ocorre porque, quando uma doença ataca os pulmões, torna-se difícil respirar adequadamente e regularmente. Uma coleção de dispositivos vestíveis pode monitorar o BRPM por meio de sensores piezoresistivos e inerciais [De Fazio et al., 2021]. Além disso, ao analisar a literatura, observamos que a maioria das abordagens requer sensores no tórax, no abdômen, no pescoço ou no nariz do paciente. Também se observa o crescimento de novas

soluções que exploram algoritmos para obter a frequência respiratória a partir de sensores ópticos embarcados em *wearables*, como *smartwatches* e pulseiras.

Além dos fatores citados acima, outra medida bastante utilizada como indicador para monitorar a saúde é a variação da frequência cardíaca (VFC) [Tiwari et al., 2021]. A VFC é a variação entre o intervalo de tempo entre dois batimentos no ciclo cardíaco e é uma das principais formas de avaliar o bom funcionamento do coração e a regulação do sistema nervoso autônomo; por isso, é uma medida relevante para identificar e auxiliar no prognóstico de diversas patologias. Tem uma grande variação interpessoal, e um valor alto representa uma resistência mais significativa ao estresse, enquanto um valor baixo pode indicar doença, estresse, depressão ou ansiedade. A principal forma de detecção da VFC é por meio da monitorização contínua no hospital ou na enfermaria, através do eletrocardiograma [Schuurmans et al., 2020]. Assim, a falta de mobilidade é a principal dificuldade no monitoramento contínuo. Desta forma, com faixas inteligentes vestíveis que contam com medições de frequência cardíaca, é possível determinar e medir a VFC usando medições de frequência cardíaca e, assim, ajudar a monitorar pacientes tanto em ambientes hospitalares quanto domésticos [Bourdillon et al., 2022].

Entre os métodos adotados para avaliar o risco de degradação da saúde de um paciente, destaca-se o *Early Warning Score* (EWS), que tem como base uma escala de alerta, conforme apresentado na Tabela 1. Trata-se de um método baseado em um sistema de atribuição de pontos (scores) de acordo com medidas dos sinais vitais, sendo a principal finalidade a identificação precoce do risco do paciente. A origem desse método é do final da década de 1990, quando estudos mostraram que a deterioração intra-hospitalar e a parada cardíaca eram frequentemente precedidas por um período de anormalidades crescentes nos sinais vitais. Desse modo, o método consiste em somar os valores agregados da Tabela 1 e associar o nível de criticidade da saúde do paciente, conforme categorizado na Tabela 2.

Os valores de referência para os sinais vitais descritos acima podem variar de acordo com o sexo, a idade ou o estado de saúde da pessoa. Contudo, são indicativos de limites que podem ser utilizados como gatilhos para a tomada de decisão em um sistema de monitoramento de saúde. Esses gatilhos podem gerar mensagens de alerta ao serem detectadas mudanças ao longo do período de monitoramento e serem explorados como oportunidades para a gestão do tempo de aquisição de dados. Ou seja, se os dados estiverem dentro dos limites dos gatilhos de cada sinal vital, respeitando o estado de saúde e a condição física da pessoa monitorada, eles não precisam ser transmitidos. Esse procedimento de uso de gatilhos pode tornar mais eficiente a utilização da estrutura de rede para a transmissão de dados.

Quanto à periodicidade de leitura dos sinais vitais, conforme RCoP [2012] e a pontuação da tabela EWS, a recomendação para pacientes com pontuação 0 é uma frequência mínima de monitorização de 12 horas, aumentando para 4 a 6 horas com pontuações de 1–4. A frequência do monitoramento deve aumentar para um mínimo de 1 hora para pacientes com uma pontuação de 5 a 6 e deve haver monitoramento contínuo dos sinais vitais para aqueles com pontuação agregada de 7 ou mais.

Tabela 1: Valores de referência de sinais vitais para EWS, onde: BRP = Taxa de Respiração, SpO2 =Saturação Periférica de Oxigênio, TC = Temperatura Corporal, FC= Frequência Cardíaca, PS = Pressão Sanguínea Diástole, SO = Suplemento de Oxigênio

Sinal Vital	**3**	**2**	**1**	**0**	**1**	**2**	**3**
BRPM	≤ 8		9-11	12-20		21-24	$25 \geq$
SpO2	≤ 91	92-93	94-95	$96 \geq$			
TC	≤ 35		35,1-36	36,1-38	38,1-39	$39,1 \geq$	
FC	≤ 40		41-50	51-90	91-110	111-130	$131 \geq$
PS	≤ 90		91-100	101-110	111-219		$220 \geq$
SO		S		N			

Fonte: https://www.york.ac.uk/crd/publications/effectiveness-matters/early-warning-systems/

Tabela 2: Nível de criticidade da saúde do paciente.

EWS Score	**Risco Clínico**
Agregado 1 - 4	Baixo
Agregado 5 - 6	Médio
Agregado ≥ 7	Alto

Fonte: https://www.york.ac.uk/crd/publications/effectiveness-matters/early-warning-systems/

Neste método de aquisição de sinais vitais pelos sensores, processar os dados, pelo menos parcialmente, na camada de borda ou *Edge*, antes de serem enviados para a nuvem, constitui uma boa estratégia para o uso da infraestrutura de rede. Essa tática oferece vantagens, como a redução de requisitos de largura de banda, na medida em que, a partir de dados brutos, extraem-se informações mais compactas [Scrugli et al., 2021]. Isso representa uma menor necessidade de largura de banda de comunicação e um menor consumo de energia relacionado à transmissão de dados sem fio, especialmente considerando que, em média, 80% da potência de um sensor é consumida na transmissão de dados [Latif et al., 2020]. Entretanto, não se consegue avançar muito nessa direção, dada a característica dos dispositivos IoT, que possuem capacidade limitada de processamento, armazenamento e, sobretudo, de bateria [Thakker et al., 2021]. Esses fatores restringem a capacidade de ter aplicativos embarcados e levam à necessidade de buscar outras abordagens para conseguir obter eficiência na transmissão dos dados adquiridos dos sensores.

2.2 Arquitetura Sistema de Monitoramento de Saúde

Escolher a arquitetura adequada é essencial para um sistema eficiente e confiável de monitoramento de saúde [Pise et al., 2023]. Fatores como limitação de recursos dos sensores IoT, latência e ocupação da banda de rede são próprios desses sistemas e, se não forem adequadamente contornados, podem degradar a qualidade e comprometer o resultado [Wen et al., 2023]. Ao pesquisar na literatura, o estado-da-arte apresenta diversos modelos de arquitetura para aquisição e transmissão de sinais vitais através de sensores IoT. Tais modelos, na sua maior parte, recaem sobre uma estrutura de três camadas: *Edge, Fog e Cloud*.

A primeira camada, *Edge*, muitas vezes referida como camada de sensores, é onde o dispositivo IoT é alocado para a leitura do sinal vital. Em alguns casos, também são considerados dados do ambiente em que está localizado, tais como umidade e temperatura, para compor o *dataset* de informações de contexto [Hassen et al., 2020, Nooruddin et al., 2020, Gia et al., 2019b, Sood et al., 2021, Debauche et al., 2019]. Monitorar as condições da sala onde o paciente se encontra enriquece a base de dados para um diagnóstico mais preciso [Paganelli et al., 2021] [Qian et al., 2022]. Ainda nessa camada, alguns autores sugerem uma camada próxima em que é alocada mais inteligência, explorando a possibilidade de alguns dispositivos que possuem uma capacidade mínima para processar e armazenar dados, denominada orvalho ou *Dew*. A *Dew* é uma computação em nuvem de pequena escala perto da fonte de dados [Medhi et al., 2021]. Quando possível, ela fornece uma IoT de assistência médica inteligente e de baixa latência; contudo, precisa de dispositivos com melhor capacidade de processamento e memória do que os sensores estáticos comuns que coletam dados. Na medida em que os dados adquiridos pelos sensores podem ser processados, mesmo que parcialmente na borda, antes de serem enviados para a nuvem, já se obtêm ganhos com a redução de requisitos de largura de banda [Scrugli et al., 2021].

A segunda camada é a que recebe os dados de saúde dos dispositivos. O uso da *Fog computing* entre os sensores e a *Cloud computing* para coletar e processar os dados reduz o tamanho dos pacotes transmitidos de ponta a ponta [Aladwani, 2019]. Além disso, uma intensidade mais baixa do sinal de rádio necessária para a transmissão de dados sem fio reduz significativamente o consumo de energia do nó do sensor [Anzanpour et al., 2020]. Ainda nesta camada, os dados são processados e podem ser comprimidos [Gia et al., 2019b, Li, 2020], criptografados [Paganelli et al., 2021, Li, 2020] ou mesmo aplicados a alguma inteligência capaz de fornecer notificações ou serviços de categorização. *Gateways* inteligentes nesta camada podem ser habilitadores de métodos para lidar com a heterogeneidade de dispositivos [Mavrogiorgou et al., 2019, Alhassoun et al., 2019] e prover meios para obter tempos de resposta a alertas mais rápidos.

Na camada de *Fog*, os nós distribuídos geograficamente oferecem uma infraestrutura de processamento de dados próxima. Estes nós podem oferecer vários serviços de integridade, fornecendo notificações para controladores de borda e gerando relatórios para administradores públicos. Ao contrário dos controladores de borda, que não se comunicam entre si, os nós de *Fog* podem trocar dados e descarregar o processamento de dados [Chabi Sika Boni et al., 2022]. Além disso, esses nós formam uma árvore hierárquica na qual os nós pais agregam informações dos nós filhos para produzir o conhecimento da região.

A terceira camada corresponde à *Cloud computing*, que possibilita a utilização de vastos recursos de armazenamento e processamento dos dados. Devido a essa capacidade escalável, é nessa camada que a informação é armazenada e colocada à disposição para mineração de dados, relatórios médicos e algoritmos de IA para análises futuras e diagnósticos. É adequada também para serviços e tarefas que não requerem resposta imediata ou alto desempenho, tais

como *dashboards* [Bandopadhaya et al., 2020], ou para mecanismos de compartilhamento de registros médicos [Sood et al., 2021]. A literatura também destaca que, para sistemas de saúde que demandam baixa latência e tempo de resposta, a inteligência se move em direção à borda da rede e próxima dos dispositivos para melhorar a qualidade de serviço (QoS) do sistema, diminuindo a latência [Moghadas et al., 2020, Vora et al., 2017].

Outra abordagem encontrada na literatura para a arquitetura de IoT separa as três camadas em: camada de percepção - *perception layer*, camada de rede - *network layer*, e camada de aplicação - *application layer* [Inbamani et al., 2022]. A partir dessa perspectiva, na camada de percepção, é feita a aquisição dos dados dos sensores; os pacotes de dados são enviados através da camada de rede para a terceira camada, que fornece serviços específicos de aplicativos para o usuário [Rahmani et al., 2021].

Por fim, a arquitetura de um sistema de monitoramento de saúde pode seguir o modelo tradicional descrito acima, normalmente referido na literatura como arquitetura baseada em árvore, podendo adotar uma estratégia centralizada ou descentralizada, conforme destacado na Figura 2.2. Nesta figura, estão representadas essas estratégias voltadas para um sistema de monitoramento da saúde. Cabe destacar o potencial a ser explorado em uma arquitetura descentralizada, na qual os aplicativos do usuário podem acessar os serviços de saúde em diversos locais, permitindo mobilidade ao usuário, com uma arquitetura que viabiliza essa situação na medida em que está próxima da distribuição geográfica de uma Cidade Inteligente.

2.3 Compressão de Dados

A compressão de dados não é uma ciência nova. Logicamente, existem novas técnicas e soluções aplicadas a tipos de arquivos modernos, mas, em geral, as soluções existentes no mercado datam de mais de duas décadas. Apesar de não ser uma novidade, a compressão de dados ganha relevância no cenário atual, onde as áreas de Inteligência Artificial, Big Data, Cidades Inteligentes e IoT se destacam. Em particular, vale ressaltar que tais demandas geram muitos dados, os quais, muitas vezes, precisam ser armazenados e transferidos entre uma máquina e outra através da rede, como já mencionado nas seções anteriores.

Para ilustrar o contexto da compressão de dados, na Figura 5 estão apresentados três cenários possíveis de transmissão e armazenamento de dados: (1) sem a aplicação de técnicas de compressão, (2) com compressão de dados e (3) comprimindo e aplicando descompressão antes de armazená-los. Nesses cenários, estão destacados os tempos envolvidos nas diferentes etapas, sendo o tempo necessário para transmissão (T_t), para compressão (T_c) e o tempo para descompressão de dados (T_d). Com isso, os tempos totais envolvidos nos cenários apresentados são $T_1 = T_t$, $T_2 = T_c + T_t$, e $T_3 = T_c + T_t + T_d$. De modo que, se dimensionado adequadamente, o esperado é $T_3 > T_2 > T_1$.

Diante disso, a ideia de compressão de dados consiste em diminuir o tamanho de um arquivo (o qual é composto por um fluxo de dados ou mesmo por um arquivo tradicional no sistema de

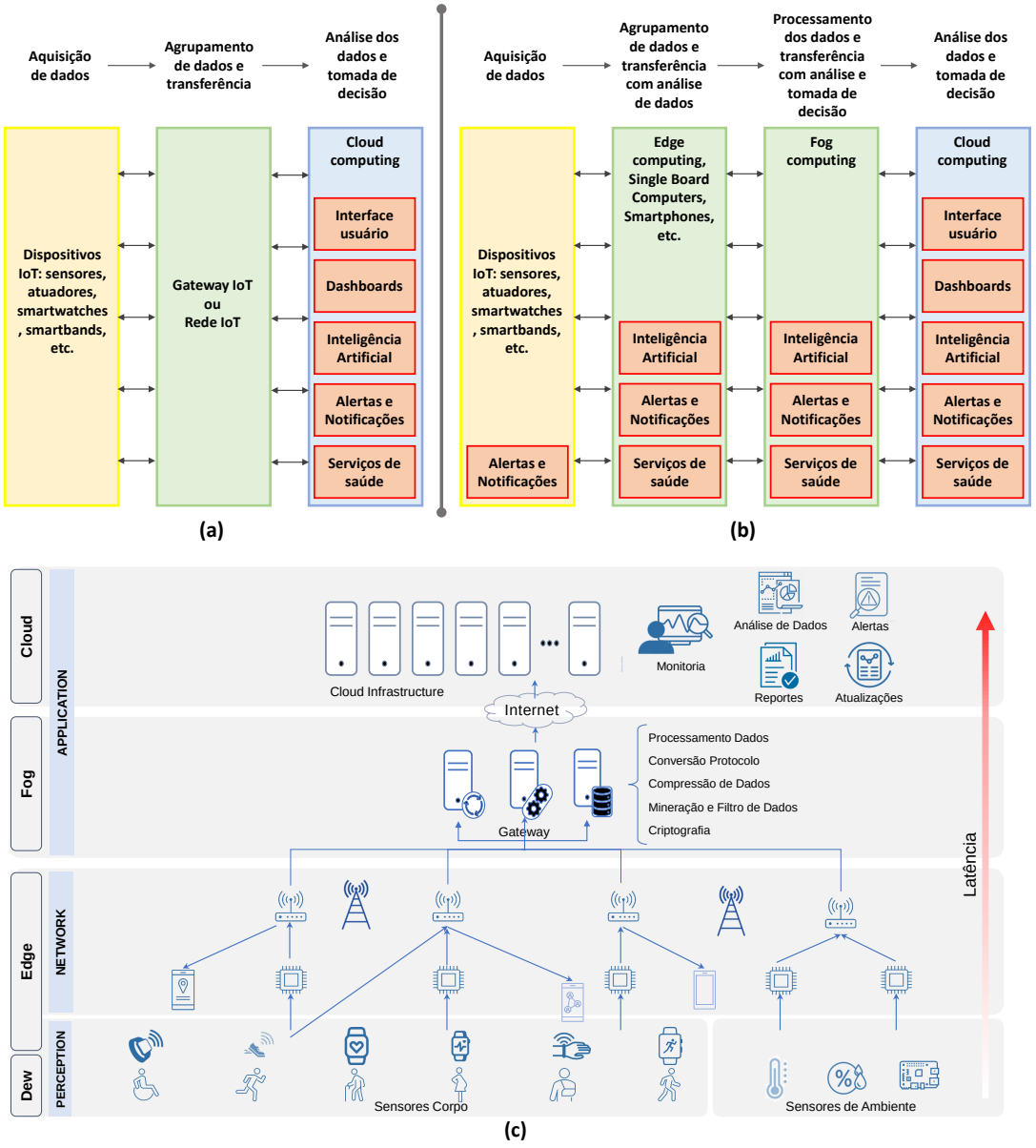


Figura 4: Comparação entre estratégias tradicionais (a) e descentralizadas (b) para monitoramento remoto de saúde baseado em IoT. Em (c), está uma visão geral da arquitetura baseada em árvore com base na estrutura (b), onde os serviços de saúde podem ser executados na *Edge, Fog e Cloud*. Em (b) e (c), os aplicativos do usuário podem acessar os serviços de saúde de qualquer lugar. Além disso, em (c), temos uma arquitetura muito próxima da distribuição geográfica de uma Cidade Inteligente, com quarteirões, bairros e comunidades (permitindo consultar dados de um bairro, por exemplo, consultando apenas um determinado nó de *Fog*).

Fonte: Elaborado pelo autor

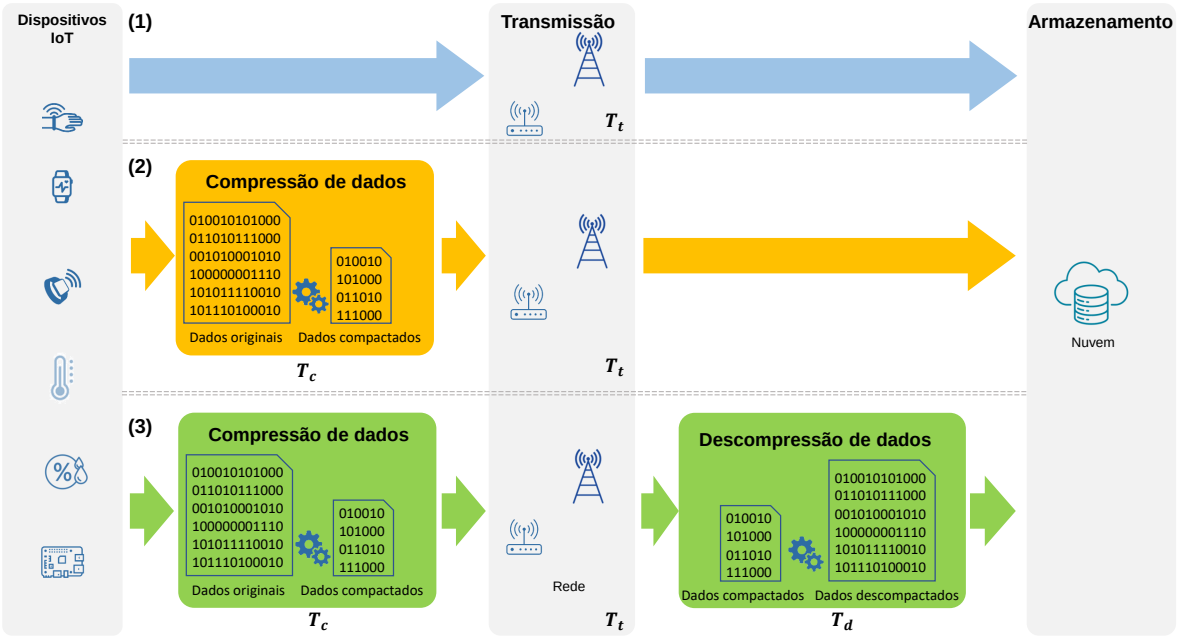


Figura 5: Cenários possíveis de transmissão e armazenamento da informação, onde (1) não aplica técnica de compressão; em (2) os dados são compactados e então transmitidos para serem armazenados e em (3) os dados são comprimidos e então transmitidos, contudo, são descompactados antes de serem armazenados.

arquivos do sistema operacional). Tal diminuição fará, portanto, com que se ocupe menos espaço em disco para armazenamento e menos largura de banda para sua transferência pela rede. Apesar de serem claros os benefícios da técnica, vale a pena atentarmos para uma série de questões. Para explicá-las, vamos usar as notações abaixo:

- $tempo_comunicacao(dados)$ - Função que retorna o tempo para a transferência de uma região de $dados$. Perceba que $dados$ pode estar na sua forma original ou não comprimida. Por questões de simplicidade, não estamos levando em consideração a tecnologia de rede empregada, tampouco os protocolos utilizados na comunicação.

- $tempo_armazenamento(dados)$ - Função que retorna o tempo de armazenamento em disco para uma quantidade de dados denotada por $dados$. Por questão de simplicidade, não estamos considerando a tecnologia de disco rígido empregada.

- $tempo_compressao(dados, tecnica)$ - Função que apresenta o tempo para aplicar o motor de compressão usando a estratégia $tecnica$ sobre uma região de $dados$.

- $tempo_descompressao(dados, tecnica)$ - Procedimento que retorna o tempo para aplicar o motor de descompressão, utilizando a estratégia $tecnica$ sobre uma região de $dados$.

Com as colocações acima, é possível denotar ao menos dois cenários para explorar o potencial de compressão de dados: (i) cenário de somente transferência de dados; (ii) cenário de transferência e armazenamento. No cenário (i), a ideia é que a compressão seja pertinente para

reduzir o tempo de comunicação. Ela é particularmente interessante em redes de larga cobertura ou naquelas que tendem a possuir um alto nível de ruído, como algumas transferências por rádiofrequência.

Tanto para o cenário (i) quanto para o cenário (ii), a noção de grão é importante para que se possa entender as inequações apresentadas. Uma vez que estamos analisando o tempo de transferência pela rede de computadores, tempo de armazenamento em memória secundária e os tempos dos motores de compressão e descompressão, o formato e o tamanho dos dados são cruciais. Por exemplo, na transferência de 100 bytes, possivelmente tenhamos uma situação em que o uso da técnica de compressão pode ser proibitivo, uma vez que a própria chamada de compressão e descompressão acarretaria uma sobrecarga que não seria superada pelos ganhos de comunicação na transferência da forma comprimida. Entretanto, a manipulação de arquivos de 1 Mbyte pode ser profícua com a compressão, desde que o ganho com a transmissão exceda a sobrecarga das traduções de original para comprimido e vice-versa. Aqui vale ressaltar que as CPUs nas máquinas de origem e destino desempenham um papel crucial, uma vez que contribuem para a velocidade das execuções dos algoritmos de compressão e descompressão de dados. Por fim, o tipo de dados de um arquivo é importante para denotar o potencial de compressão. Usando o algoritmo empregado no WinZip, enquanto em arquivos texto ASCII é possível atingir taxas de compressão de até 73%, bibliotecas DLL ou arquivos EXE possuem taxas de compressão de até 50% (em média)[1].

2.3.1 Tipos de compressão de dados

Os algoritmos de compressão de dados são tradicionalmente classificados em dois tipos: com perdas (ou irreversíveis), do inglês *lossy*, ou sem perdas (ou reversíveis), do inglês *lossless* [Shirani, 2008]. Embora o primeiro algoritmo geralmente forneça um resultado mais significativo no que se refere à taxa de compressão, esse algoritmo não possibilita reconstruir totalmente os dados originais. Por outro lado, os sistemas de compressão sem perdas reduzem o tamanho dos dados transmitidos ou armazenados, focando em reduzir a redundância de informações da origem, preservando a entropia da informação e permitindo a reconstrução integral dos dados originais. Comparativamente, a taxa de compressão dos algoritmos *lossless* é menor do que a do algoritmo *lossy*.

Os algoritmos *lossless* comprimem o arquivo geralmente por codificação de repetição, basicamente substituindo uma palavra ou caractere que aparece com maior frequência por um código que utiliza menos espaço de armazenamento. Esses algoritmos comumente utilizam estratégias de codificação de redundância mínima ou métodos de dicionário. A técnica de redundância mínima consiste em representar os caracteres com o maior número de ocorrências usando menos bits. Um exemplo de algoritmo muito conhecido que aplica essa técnica é a Codificação de Huffman. Quanto ao método do dicionário, aplica-se um arquivo como um di-

[1] Detalhes em: https://kb.corel.com/en/125890

cionário de expressões com seus valores correspondentes em bits; como exemplo de algoritmo que utiliza essa técnica, temos o Lempel-Ziv-Welch (LZW). Outros algoritmos *lossless* amplamente conhecidos são ZIP e RAR, e apresentam maior rendimento, como citado na seção anterior, quando aplicados a arquivos de texto ASCII.

Os algoritmos *lossy*, por outro lado, eliminam uma certa quantidade de dados que não é perceptível e, desse modo, não permitem que a informação seja restaurada em sua forma original. Contudo, reduzem significativamente o tamanho, conforme ilustrado na Figura 6. O esperado deste tipo de algoritmo é que permita uma análise similar entre os dados comprimidos e os originais, sem eliminar informações relevantes e com a mais alta taxa de compressão possível. Ou seja, essa técnica é benéfica se a qualidade dos dados não for uma prioridade, mas sim o envio ou o armazenamento de dados. Esse tipo de compactação de dados é usado principalmente para comprimir dados multimídia, como voz, vídeo e imagens.

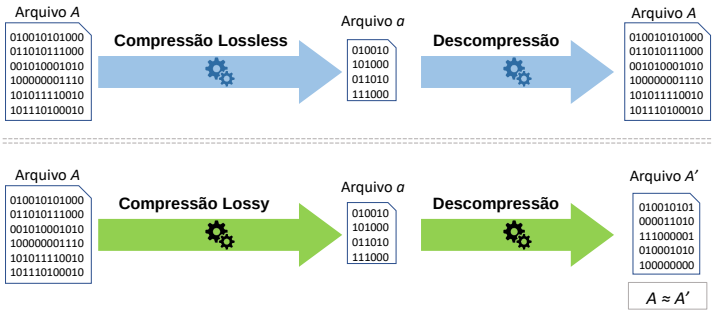


Figura 6: Dois principais métodos de compressão, Lossless e Lossy. O método Lossy resulta em maior taxa de compressão, contudo os dados restaurados a partir da descompressão não correspondem aos do arquivo original.

Nesta categoria do algoritmo de compressão com perdas, os métodos "lineares por partes", em inglês, *piecewise linear*, são frequentemente utilizados na indústria, pois demandam baixo poder de processamento e podem ser aplicados em tempo real no momento da aquisição dos dados [Roy and Nikolaidis, 2022]. Estes métodos consideram que o sinal segue uma linha reta enquanto os pontos amostrados estiverem contidos dentro de uma tolerância especificada. Os dados considerados redundantes ou mesmo próximos de uma certa tendência linear são descartados e, desse modo, tais dados podem ser recuperados futuramente com baixo erro, utilizando técnicas de interpolação. Dentre os algoritmos *lossy* que utilizam esse método e são adotados na I4.0, destacam-se o Boxcar/Backslope e o Swinging Door Trending [Khan et al., 2020] [Yin et al., 2021].

A união de dois algoritmos, Boxcar e BackSlope, resulta em uma técnica na qual se armazenam apenas os pontos que satisfazem as condições de ambos. Os parâmetros do algoritmo são a largura da janela do Boxcar e a largura da janela do Backslope. A partir da largura da janela informada, o algoritmo estabelece um limite de variação horizontal em relação ao último valor armazenado (BoxCar). Conforme apresentado na Figura 7, a cada novo valor reportado, são calculados limites de variação diagonal, utilizando a mesma largura de janela (BackSlope). Dessa

forma, assim que um valor violar duas áreas estabelecidas pelas linhas BoxCar e BackSlope, o valor deve ser armazenado.

O algoritmo Swinging Door Trending, apresentado na Figura 7b, utiliza como parâmetros o tempo máximo de compressão e Desvio de compressão. Considerando que, quanto maior o tempo máximo de compressão e o desvio de compressão, maior será a taxa de compressão. Uma área de cobertura é criada no formato de um paralelogramo, com a altura igual ao dobro do desvio de compressão entre o último valor armazenado e o último valor recebido. Se algum dos pontos coletados entre o último valor armazenado e o valor atual estiver fora da área, todos os pontos são descartados, com exceção do penúltimo ponto recebido, que é armazenado. Ainda, sempre que o intervalo entre o último valor armazenado for superior ao tempo máximo de compressão, o último valor deve ser armazenado.

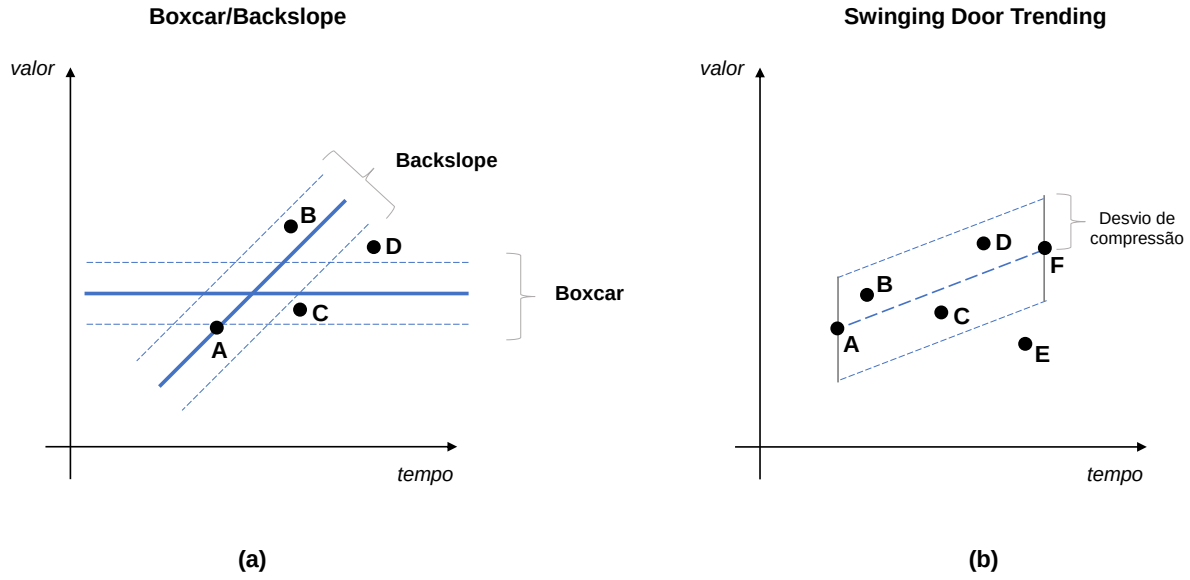


Figura 7: Em (a) está apresentado o algoritmo de compressão Boxcar/Backslope, onde o ponto A é o último ponto armazenado, o ponto B viola apenas o Backslope, o ponto C viola apenas o Boxcar e o ponto D viola os dois critérios e portanto é armazenado. Em (b) está o algoritmo Swinging Door Trending, com uma área de cobertura entre o último valor armazenado (A) e o último valor recebido (F), nesse caso os valores B, C e D serão descartados, enquanto o ponto E será armazenado.

2.3.2 Combinando técnicas de compressão de dados

Diferentes cenários e necessidades de sistemas direcionam à escolha de um algoritmo de compressão de dados com ou sem perda, conforme apresentado acima. A escolha recai sobre a qualidade dos dados que se deseja processar, bem como sobre o desempenho, o custo e a eficiência de armazenamento. Geralmente, a compactação com perdas é adequada para dados que não são sensíveis a pequenas alterações ou erros, como arquivos multimídia, conteúdo da Web ou dados analíticos. A compactação com perdas possibilita lidar melhor com fatores intrínsecos à transferência de dados, como latência e ocupação de largura de banda, além de

permitir economizar mais espaço de armazenamento e acelerar a transferência. No entanto, a compactação com perdas também introduz distorção e degradação dos dados, reduzindo a precisão da análise e a confiabilidade do backup. A compactação sem perdas é adequada para dados que requerem preservação e reprodução exatas, como arquivos de texto, arquivos binários ou dados transacionais. A compactação sem perdas pode ajudá-lo a reter todas as informações, a qualidade dos dados e garantir a consistência e a exatidão dos dados. Contudo, a compactação sem perdas também consome mais espaço de armazenamento e recursos, além de retardar a compactação e a descompactação dos dados.

Dadas as considerações da Seção 2.3.1 sobre as técnicas de compressão *lossy* e *lossless* e, retomando os cenários apresentados na Figura 5, podemos ponderar sobre quais técnicas são mais adequadas para cada cenário. Para a técnica de compressão *lossy*, os cenários 2 e 3 são mais adequados para serem aplicados, pois alguns tipos de arquivos não necessitam ser restaurados ao tamanho original após terem sido comprimidos. Por exemplo, na compressão de imagem padrão JPEG, a imagem original tem seu tamanho reduzido e não precisa mais ser restaurada. O mesmo se aplica a arquivos de áudio, nos quais o som original pode ser reduzido com baixo impacto no custo de fidelidade. A compressão do tipo *lossy* elimina apenas sons menos audíveis e menos significativos, reduzindo o espaço necessário para que sejam armazenados ou transmitidos, sem a necessidade de restaurar a qualidade original. Contudo, para a técnica de compressão *lossless*, apenas o cenário 3 contempla, na medida em que o arquivo comprimido é restaurado e se obtém os dados originais na sua integridade após o processo de descompressão.

2.4 Projeto MHD

O *framework* de gestão de dados de sinais vitais desenvolvido neste trabalho faz parte de um projeto denominado Minha História Digital (MHD), patrocinado pela FAPERGS e conduzido por meio de cooperação científica entre a Unisinos, o Instituto Colaborativo de Blockchain (ICOLAB) e uma aliança de pesquisa e inovação com hospitais do Rio Grande do Sul.

O projeto MHD visa ao desenvolvimento de um modelo computacional que compreende a arquitetura e os algoritmos que permitem a rastreabilidade e a captura de sinais vitais de pessoas em tempo real no escopo de uma determinada cidade. O projeto atua no combate a doenças respiratórias, tais como COVID-19,visto que possibilita obter a posição geográfica e os sinais vitais pontuais de uma determinada pessoa, bem como o histórico de dados ao longo do tempo. Com isso, é possível verificar o grau de dispersão de uma doença por meio do mapeamento dos lugares que a pessoa visitou, analisar a eficácia das ações de quarentena, bem como observar e predizer evoluções críticas em relação aos sinais vitais dos indivíduos.

Para atingir esse propósito, o projeto contempla a adoção de IoT, *Fog Computing* e *Cloud Computing* para a análise e armazenamento eficazes e escaláveis de informações. Nesse contexto, o projeto possui três principais objetivos:

- apresentar uma arquitetura que contemple diferentes níveis de instrumentação em IoT,

desde o uso de pulseiras e *wearables* até a instalação de leitores que atuam como portais em locais estratégicos da cidade, bem como o sensoriamento de ambientes hospitalares.

- organizar servidores e serviços em *Fog* e em *Cloud*, de modo que o número deles seja dinâmico de acordo com as necessidades computacionais e o número de usuários em um determinado momento. Dessa forma, o projeto possibilita a economia de energia e uma qualidade de serviço (QoS) inteligente para os usuários.

- definir serviços e algoritmos sensíveis à latência de rede, os quais serão executados em nós na *Fog*, e aqueles que realizarão computações mais pesadas, que serão executados na *Cloud*.

Uma visão geral da arquitetura do projeto MHD pode ser vista na Figura 8. A arquitetura é composta por três camadas: *Edge, Fog e Cloud*. Os indivíduos monitorados na camada *Edge* utilizam sensores de saúde disponíveis no mercado para monitorar seus sinais vitais durante as atividades diárias. Na camada *Fog*, os nós distribuídos geograficamente possibilitam uma infraestrutura de processamento de dados próxima da captura. Por fim, a camada *Cloud* fornece uma plataforma central que agrega dados de toda a implantação. Os serviços em nuvem empregam uma infraestrutura robusta para a análise aprofundada dos dados. Os administradores de saúde têm uma visão mais ampla de várias regiões, permitindo soluções capazes de identificar zonas de risco e surtos de doenças o mais cedo possível.

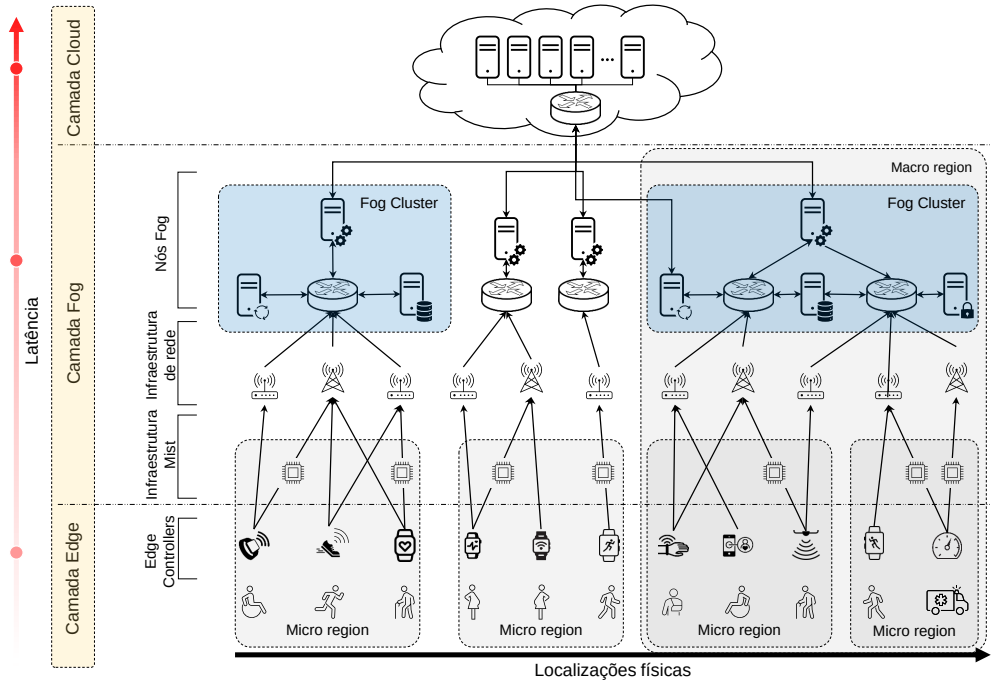


Figura 8: Visão arquitetural da solução através da combinação de Internet das Coisas, *Fog Computing* e *Cloud Computing*.

Cabe destacar na camada *Edge* a presença dos dispositivos *Edge Controllers*, situados próximos dos sensores, podendo ser um *smartphone* ou um computador de placa única instalado

na residência do usuário. Ao contrário dos *Edge Controllers*, que não se comunicam entre si, os nós da *Fog* podem trocar dados e descarregar o processamento de dados. Estes nós formam uma árvore hierárquica na qual os nós pais agregam informações dos nós filhos para produzir o conhecimento da região. Esse formato de arquitetura em árvore pode ser visto na Figura 9.

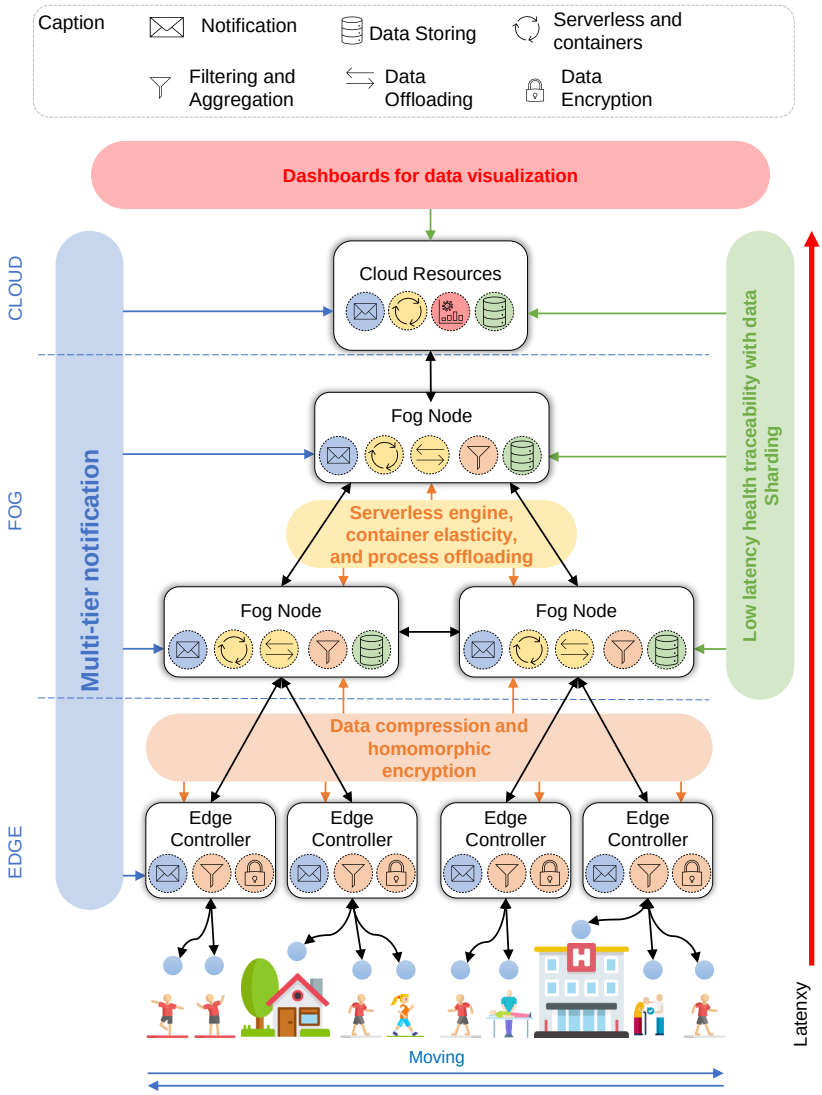


Figura 9: Arquitetura em três camadas com destaque para os nós da *Fog* adotando uma arquitetura em árvore. Fonte: [Rodrigues et al., 2023].

Nesse contexto do projeto MHD, o *framework* VSAC é responsável pelo processo de compressão de dados adaptativa, posicionado na camada de *Edge*. Ou seja, os nós da camada *Fog* recebem os dados dos sinais vitais da camada *Edge*, mais precisamente dos *Edge Controllers*, aplicando os métodos de compressão e tratando das notificações.

3 TRABALHOS RELACIONADOS

Este capítulo tem como objetivo apresentar os trabalhos relacionados ao *framework* proposto. Com base no tema da pesquisa, foram analisados alguns trabalhos relacionados que buscam endereçar problemas semelhantes aos citados na Seção 1.1, ou que podem apoiar possíveis tentativas de solução. Dessa forma, as próximas seções descrevem algumas abordagens que se concentram em técnicas de compressão de dados para obter eficiência e qualidade na transmissão e restauração dos dados no destino, a fim de proporcionar uma visão do atual estado da arte neste processo. Os artigos estão classificados em ordem crescente por ano de publicação e, em seguida, em ordem alfabética dos nomes dos autores. Por fim, as iniciativas serão comparadas e analisadas para identificar as lacunas atuais na literatura da área.

3.1 Metodologia de seleção dos trabalhos relacionados

A revisão da literatura utilizada neste estudo adota os princípios de revisão sistemática para obter reprodutibilidade e resultados de alta qualidade [Biolchini et al., 2005, Keele et al., 2007]. O objetivo é obter uma visão geral das estratégias de compressão de dados para sinais em sistemas de monitoramento de saúde. O escopo da pesquisa bibliográfica abrange a seleção de artigos de conferências e periódicos relevantes, indexados nas seguintes bases científicas: ACM Digital Library [1], Google Scholar [2], IEEE eXplore [3], ScienceDirect [4] e Scopus [5].

A pesquisa foi realizada nessas bases científicas em abril de 2025, procurando por artigos publicados a partir do ano de 2018, com o objetivo de identificar trabalhos que contivessem palavras-chave e alguns termos que deveriam aparecer no título ou no resumo dos artigos selecionados. Para esse processo, foram definidas as palavras-chave de pesquisa e a definição do escopo da pesquisa. Essas palavras foram definidas para obter resultados de pesquisa direcionados ao tema deste trabalho. Dessa forma, foi estabelecida a seguinte *string*, apresentada no quadro a seguir, para a busca e seleção dos artigos nas bibliotecas acadêmicas eletrônicas.

```
(IoT OR "Internet Of Things" OR sensors OR actuators)
                        AND
(latency OR bandwidth OR network)
                        AND
("data compress" OR compression)
                        AND
(healthcare OR medical OR "vital-signs")
```

Esta pesquisa resultou em 437 referências, e foram aplicados os três filtros sobre essa base para a seleção final dos artigos. Na lista abaixo, estão descritos os critérios de exclusão aplica-

[1] https://dl.acm.org/
[2] https://scholar.google.com.br/
[3] https://ieeexplore.ieee.org/
[4] https://www.sciencedirect.com/
[5] https://scopus.com/

dos em cada filtro, e na Figura 10 é apresentado o processo de seleção.

- **Filtro 1 - Remoção de duplicidades**: Todos os trabalhos foram organizados em uma lista, ordenados pelo título do trabalho, e as duplicidades foram removidas.

- **Filtro 2 - Revisão do título e resumo**: Foi realizada a leitura dos títulos e resumos dos trabalhos, e foram removidos todos aqueles que não estavam relacionados a aquisição e transmissão de sinais vitais para monitoramento, bem como à busca por eficiência na transmissão de dados.

- **Filtro 3 - Análise do texto completo**: Nesta etapa, foi realizada a leitura completa dos artigos. Foram selecionados apenas os trabalhos que apresentam soluções voltadas para a eficiência na transmissão de dados de sinais vitais através da rede, utilizando técnicas adaptativas ou combinando métodos para obter melhores taxas de compressão.

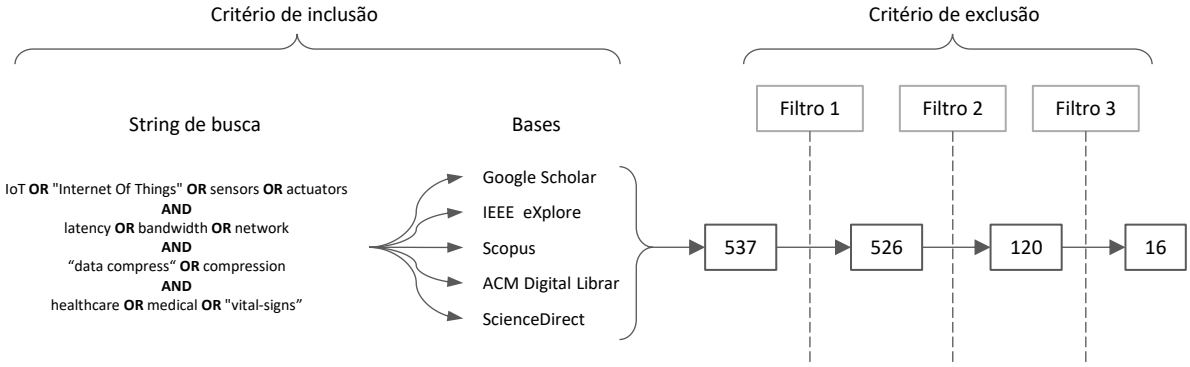


Figura 10: Sequência do processo de filtros para seleção dos artigos.

Tabela 3: Trabalhos relacionados

ÍD	Autor	Título	Ano
TR1	Xiao et al.	Adaptive Compressed Classification for Energy Efficient Activity Recognition in Wireless Body Sensor Networks	2018
TR2	Azar et al.	An energy efficient IoT data compression approach for edge machine learning	2019
TR3	Hanoune and Lysmos	Data compression mechanisms in an intelligent E-health gateway for medical monitoring applications	2020
TR4	Kalaivani et al.	Design and Implementation of Hybrid Compression Algorithm for Personal Health Care Big Data Applications	2020
TR5	Passos et al.	GROWN: Local Data Compression in Real-Time To Support Energy Efficiency in WBAN	2020
TR6	Lu et al.	Adaptively Compressing IoT Data on the Resource-constrained Edge	2020
TR7	Das and Rahman	A Compression Technique for Electronic Health Data through Encoding	2021
TR8	Idrees et al.	An Edge-Fog Computing-Enabled Lossless EEG Data Compression With Epileptic Seizure Detection in IoMT Networks	2022
TR9	Khlief and Idrees	Efficient EEG Data Compression Technique for Internet of Health Things Networks	2022
TR10	Hossein and Sheikh	A Novel Hybrid Medical Data Compression Using Huffman Coding and LZW in IoT	2022
TR11	Amiri et al.	SIC-EDGE: Semantic Iterative ECG Compression for Edge-Assisted Wearable Systems	2022
TR12	Idrees and Khlief	Efficient compression technique for reducing transmitted EEG data without loss in IoMT networks based on fog computing	2023
TR13	Chang and Sobelman	Lightweight Lossy Lossless ECG Compression for Medical IoT Systems	2023
TR14	Vakil and Shirmohammadi	LPRLC Linear predictive run length coding to improve energy consumption of wireless body area networks	2025
TR15	Kadhum Idreesees et al.	SZ4IoT an adaptive lightweight lossy compression algorithm for diverse IoT devices and data types	2025
TR16	Baliarsingh and Mohanty	Electrocardiogram Compression Using Adaptive Decomposition Techniques for Smart Healthcare System	2025

Fonte: Elaborado pelo autor

Ao final dessa etapa, restaram 16 artigos que exploravam o assunto da forma desejada e que estão listados na Tabela 3. Um resumo de cada um dos trabalhos é apresentado na Seção 3.2.

3.2 Descrição do Estado da Arte

A análise dos trabalhos relacionados (TR1 a TR16) na área de compressão de dados para sistemas de monitoramento de saúde em arquiteturas de Internet das Coisas (IoT) revela três eixos principais de pesquisa e desenvolvimento de soluções. Estes trabalhos buscam abordar problemas como a economia de energia, a redução do espaço de armazenamento e a otimização do desempenho da rede, mantendo a Qualidade de Serviço (QoS). A seguir, os trabalhos são agrupados e analisados conforme o foco na adaptação da compressão aos recursos computacionais, no uso de estratégias híbridas que combinam métodos com e sem perdas, e na integração da compressão com mecanismos analíticos e arquiteturas distribuídas Edge–Fog–Cloud.

3.2.1 Adaptação da Compressão ao Contexto e aos Recursos Computacionais Disponíveis

Diversos estudos concentram-se em adaptar a compressão para lidar com as severas restrições de recursos (energia, processamento, rede) inerentes aos dispositivos de borda e sensores vestíveis, ajustando dinamicamente os parâmetros de compressão em tempo de execução para equilibrar a eficiência e a precisão.

Xiao et al. (TR1) propõem um mecanismo de Classificação Comprimida Adaptativa (Adaptive CC) baseado em *Compressive Sensing* (CS) para eficiência energética em Redes de Sensores Corporais Sem Fio (WBSN). O método busca a taxa mínima de compressão para os sinais vitais que garanta a precisão desejada no reconhecimento de atividades. Esse ajuste é feito dinamicamente com base no *feedback* do resultado do reconhecimento da atividade, resultando em até 28% de economia de consumo de energia.

Similarmente, Azar et al. (TR2) focam na eficiência energética em dispositivos vestíveis, aplicando um compressor *error-bounded* (com limite de erro ajustável) com perdas. O método utiliza uma versão adaptada do algoritmo SZ para o contexto de IoT. O limite de erro é ajustável, podendo ser controlado pela equipe médica. Essa compressão é aplicada na camada do sensor, antes da transmissão para o Edge, com o objetivo de reduzir o número de bits transmitidos e proporcionar uma economia de bateria de até 27% após quatro horas.

No âmbito da adaptabilidade focada em recursos de infraestrutura, Lu et al. (TR6) introduzem o *middleware* ZipMate. Este modelo utiliza uma estratégia adaptável que emprega um modelo matemático para reconhecer a compactação de dados e os gargalos de transferência sob limitações de CPU, I/O para leitura de dados, largura de banda da rede e espaço de armazenamento. O ZipMate seleciona o mecanismo de compressão mais adequado em tempo real, com base no objetivo (seja para acelerar o tempo de transferência ou maximizar a taxa de compressão). Os resultados demonstraram que a largura de banda da rede tem um impacto mais significativo no tempo de transferência do que o poder de computação.

Amiri et al. (TR11) propõem o SIC-Edge, uma compressão adaptativa semântica e cola-

borativa para ECG. O método estabelece um *loop* de controle colaborativo entre o sensor e o servidor Edge para construir, iterativamente, uma representação semântica. Utilizando um mecanismo de Teste de Hipóteses Sequencial (SHT) sobre coeficientes *wavelet* (representação em camadas), o sistema ajusta a resolução da compressão (quantas camadas são solicitadas) dinamicamente, de acordo com os requisitos de um classificador CNN, minimizando a carga de computação no sensor e garantindo o nível de confiança desejado. O SIC-Edge aumenta em até 35% e 26% as métricas de *recall* e F1-score, respectivamente, em comparação com estratégias estáticas.

Por fim, Kadhum Idrees et al. (TR15) desenvolveram o SZ4IoT, um algoritmo *lossy* leve e adaptativo projetado para dispositivos IoT com recursos limitados. O algoritmo utiliza interpolação adaptativa para prever valores futuros, armazenando apenas aqueles que excedem um limiar de erro predefinido, garantindo, assim, compressão com controle sobre a perda. O SZ4IoT demonstrou tempos de compressão inferiores a 5 ms e economia de energia de até 31,4% em microcontroladores comuns, sendo adequado para compressão de séries temporais na borda.

3.2.2 Uso de Estratégias Híbridas que Combinam Métodos Com e Sem Perdas

Muitos trabalhos adotam a combinação de algoritmos *lossy* (com perdas) e *lossless* (sem perdas) em estágios sucessivos ou em cascata. Essa abordagem visa aproveitar a alta taxa de compressão dos métodos *lossy* na redução inicial do volume de dados, seguida por uma etapa *lossless* que preserva a integridade da informação remanescente, buscando o equilíbrio ideal entre Taxa de Compressão (TC) e fidelidade dos dados.

Kalaivani et al. (TR4) propõem um Algoritmo de Compressão Híbrida (HCA) para dados de ECG. O HCA combina o processo de decimação e interpolação (*lossy*) com a Codificação Rice-Golomb (*lossless*). O objetivo é alcançar um *trade-off* entre TC e *Percentage Root-mean-square Difference* (PRD), com o fator de decimação sendo determinado pela perda aceitável na compressão. O método resultou em uma redução de 70% no espaço de armazenamento.

Passos et al. (TR5), através do mecanismo GROWN, aplicam primeiramente compressão com perdas baseada em um *threshold* (componente CTC) para limitar as perdas de dados, seguida de compressão sem perdas por tabelas de codificação, reduzindo a quantidade de informações transmitidas.

Chang e Sobelman (TR13) propõem a técnica LPRLC (Compressão de ECG Leve com Perdas/Sem Perdas) para sensores WBAN. O LPRLC combina a Codificação Preditiva Linear (LPC), que é um método com perdas para prever o sinal, com a Codificação por Comprimento de Execução (RLE), que comprime o erro de forma sem perdas. Projetado para operar no sensor, alcançou uma compressão de até 71% e uma economia média de energia de 98%.

Vakil e Shirmohammadi (TR14) desenvolveram um modelo híbrido para ECG que utiliza um codificador com perdas baseado em redes neurais profundas (DNNs), seguido por uma

etapa de compressão sem perdas via Huffman. O foco é obter alta TC (superior a 5) e baixa distorção (PRD de até 0,2), garantindo que a acurácia diagnóstica do sinal, avaliada por DNNs, seja mantida após a compressão.

Baliarsingh e Mohanty (TR16) introduzem uma abordagem híbrida que combina técnicas de decomposição e transformação: *Adaptive Variational Mode Decomposition* (AVMD) e *Wavelet Packet Transform* (WPT), seguidas por codificação aritmética. O método busca reduzir os requisitos de armazenamento e transmissão de ECG sem comprometer a informação clínica, alcançando TC de 41,19% com PRD de 0,29.

Outros trabalhos combinam múltiplas técnicas de compressão *lossless* ou utilizam pré-codificação para otimizar a redundância:

Hossein e Sheikh (TR10) apresentam um método de compressão híbrida em cascata que combina LZW e Codificação Huffman, com uma etapa de rearranjo de informações binárias no meio. O uso de rearranjo de dados entre as duas etapas *lossless* busca aumentar a TC, preservando 100% da precisão dos dados. O algoritmo reduziu os dados em uma média de 37,85% para sinais, imagens e textos. Hanoune e Lysmos (TR3) utilizam, em um primeiro estágio, técnicas como compactação por previsão, amostragem e agendamento, seguidas de algoritmos sem perda (LZO, Bzip2 ou LZ4). Das e Rahman (TR7) propõem codificar os dados de sinais vitais utilizando um dicionário estruturado com base em padrões médicos, antes de aplicar a compressão *lossless*. Essa pré-codificação otimiza a compactação de sequências pouco repetitivas, como as de sinais vitais, e superou outros métodos (como LZO, LZ4, BZ2) em TC.

3.2.3 Integração da Compressão com Mecanismos Analíticos e Arquiteturas Distribuídas do Tipo Edge–Fog–Cloud

Esta frente de pesquisa aborda a colocação estratégica dos mecanismos de compressão nas camadas da arquitetura Edge–Fog–Cloud, frequentemente integrando-os a processos de análise (como *Machine Learning* ou algoritmos de agrupamento) para gerenciar o tráfego de dados e apoiar o diagnóstico rápido.

Idrees et al. (TR8) implementaram uma abordagem em uma arquitetura Edge–Fog–Cloud para compressão de dados de EEG e detecção de crises epilépticas. A compressão *lossless* (KCHE: K-means clustering + Codificação Huffman) é aplicada primeiro na Edge antes de ser enviada para o Fog. A detecção da doença é realizada na camada Fog usando *Machine Learning* (Naive Bayes). Em seguida, o algoritmo KCHE é aplicado novamente na Fog antes de ser transmitido para a Cloud para armazenamento e análise adicional. Essa abordagem resultou na redução do tempo de latência em até 88,2%.

Khlief e Idrees (TR9) desenvolveram a técnica EDaCoT (*Efficient EEG Data Compression Technique*) em um *gateway* na camada Fog para redes IoHT. O EDaCoT combina agrupamento DBSCAN, seguido de Codificação RLE e Codificação Huffman para compactar pacotes de EEG de forma *lossless*. O método permite que os pacotes sejam transmitidos com taxas de

compressão elevadas (entre 66% e 98,6%) e com tempo de processamento inferior ao de outros métodos.

Idrees e Khlief (TR12) propõem a técnica ECoT (*Efficient Compression Technique*) dentro de uma arquitetura Edge–Fog–Cloud. O método é aplicado em um *gateway* na Fog para agrupar e compactar os sinais de EEG antes do envio para a Cloud. O ECoT utiliza a combinação de agrupamento DBSCAN, Codificação Delta e Codificação Huffman (lossless). O foco é reduzir o volume de dados para otimizar o desempenho da rede IoMT e diminuir o tempo de latência na geração de diagnósticos médicos. Os testes demonstraram TC na faixa de 65% a 85%.

Embora Hanoune e Lysmos (TR3) utilizem uma abordagem híbrida (múltiplas técnicas lossy + lossless), o mecanismo é implementado em um *gateway* inteligente E-Health para otimizar o espaço de armazenamento e acelerar a transferência para a Cloud, o que se enquadra na arquitetura distribuída.

Além disso, o trabalho de Amiri et al. (TR11), embora classificado como adaptação devido ao seu controle dinâmico, também é relevante aqui por sua integração analítica. O SIC-Edge emprega um *loop* de controle colaborativo entre o sensor e o Edge *server* para garantir que a compressão se adapte às necessidades de um classificador CNN para detecção de arritmia, destacando a fusão de compressão e análise em tempo real em camadas distribuídas.

3.3 Análise e oportunidades

Esta seção fará uma análise dos trabalhos relacionados previamente selecionados, com foco nas tendências, abordagens recorrentes, lacunas e possíveis oportunidades de pesquisa na área de compressão de dados para transmissão eficiente de sinais vitais em sistemas de monitoramento de saúde. A Tabela 4 apresenta uma visão descritiva de cada trabalho, destacando o problema enfrentado, a abordagem adotada, a técnica empregada e os principais resultados obtidos.

Já a Tabela 5 organiza essas informações de forma categorizada em diferentes tópicos, permitindo observar padrões em termos de aplicação (como foco em eficiência energética, uso de espaço de armazenamento, desempenho de rede e qualidade da informação), método (modelos adaptativos ou parametrizados) e tipo de compressão utilizado (com perdas, sem perdas ou híbrido). Essas características foram elencadas com base no foco desta tese e nos principais temas abordados nos trabalhos selecionados, podendo, assim, oferecer uma visão mais clara das possíveis oportunidades e lacunas de atuação. Segue abaixo a descrição desses tópicos:

Tabela 4: Trabalhos relacionados analisados de acordo com o problema a ser resolvido, abordagem adotada técnica desenvolvida para obter o resultado.

Identificador	Autores	Ano	Problema abordado	Abordagem	Técnica	Resultados
TR1	Xiao et al.	2018	Alto consumo de energia de dispositivos IoT para transmitir dados de sinais vitais.	Reduzir o tamanho de pacotes com técnicas de compressão.	Modificar o método de Compressive Sensing para torná-lo adaptativo e reduzir o tamanho do pacote de dados.	Melhoria no reconhecimento de acurácia (90,5%) com redução de 28% no consumo de energia.
TR2	Azar et al.	2019	Alto consumo de energia em dispositivos vestíveis para dados de sinais vitais.	Aplicar dois estágios de processamento comprimindo dados antes da transmissão e depois restaurando na Edge usando técnicas de aprendizado de máquina.	Técnica de compressão com perdas na camada do sensor e aplicação de modelos de aprendizado profundo na borda para reconstruir os dados e enviá-los para a nuvem.	Redução da quantidade de dados em até 103 vezes sem perda de acurácia na classificação baseada em dados comprimidos.
TR3	Hanoune and Lysmos	2020	Espaço de armazenamento e tempo de transferência de dados em aplicações de monitoramento de saúde.	Comprimir dados no gateway antes de transmitir para Cloud.	Aplicar compressão por predição, amostragem e agendamento. Em seguida, aplicar compressão sem perdas e enviar à nuvem.	Para predição, Bzip2 teve melhor desempenho com TC de 8,43%; LZO e LZ4 para amostragem com TC entre 13,6% e 15,27%. Para agendamento, LZO com TC de 16,7%.
TR4	Kalaivani et al.	2020	Consumo de energia dos dispositivos IoT na transmissão e espaço de armazenamento	Comprimir os dados adquiridos antes de envio na camada de borda combinando técnicas de compressão.	Aplicar compressão sem perdas e outra híbrida, que inclui compressão com e sem perdas.	A análise do espaço de armazenamento utilizando o algoritmo proposto mostrou uma redução de 70% no espaço necessário.
TR5	Passos et al.	2020	Consumo de energia dos dispositivos IoT.	Gerenciar a compressão de dados em tempo real de acordo com o tipo de sinais vitais, evitando transmitir dados redundantes.	Aplicar compressão com perdas conforme um limiar para limitar a perda de dados dentro de limites aceitáveis e, a seguir, aplicar compressão sem perdas.	O método proposto alcançou eficiência energética de até 53,73% com atraso máximo de 55ms.
TR6	Lu et al.	2020	Alto consumo de largura de banda para transmitir grandes volumes de dados e uso de espaço de armazenamento.	Mecanismo de compressão adaptativo decide reduzir tempo de transmissão ou o espaço de armazenamento conforme tipo de dado.	Modelo matemático que considera CPU, I/O para leitura de dados, largura de banda da rede, e seleciona mecanismo de compressão mais adequado.	O método supera o melhor compressor unitário LZMA em 5% na taxa de compressão com 17% menos tempo de compressão.
TR7	Das and Rahman	2021	Espaço de armazenamento para dados e tempo de transmissão, em redes de baixa velocidade.	Criar dicionário com base na estrutura dos dados para codificá-los antes da compressão.	Classificar os dados com base em uma tabela padrão para cada tipo de sinal vital, depois comprimir com método padrão sem perdas e transmitir.	A técnica apresenta desempenho entre 18% a 55% melhor do que LZO, LZ4, BZ2, LZSS, LZMA.
TR8	Idrees et al.	2022	Latência e ocupação da largura de banda no envio de dados por redes IoMT.	Reduzir quantidade de dados na Edge, identificar possibilidade de crise epiléptica na camada Fog, e aplicar nova compressão antes do envio Cloud.	Combinar K-means e codificação de Huffman, depois enviar ao fog para detecção da doença com ML (Naive Bayes). Em seguida, aplicar novamente o método da primeira etapa.	A abordagem reduz o tamanho dos dados EEG após a compressão entre 85,6% e 89,2% comparado aos dados não comprimidos.
TR9	Khlief and Idrees	2022	Baixa eficiência no envio de dados em tempo real com largura de banda limitada, latência e consumo de energia.	Comprimir pacotes de dados oriundos de sensores em um gateway e transmitir com taxa de distorção configurável.	Agrupamento de dados com DBSCAN seguido de compressão sem perdas. Na etapa final, os dados são comprimidos com Huffman e transmitidos.	Dados EEG comprimidos entre 66% e 98,6%, enquanto LZW, HE, HCLZW e HCHE comprimem entre 40% e 54%, 65% e 70%, 14% e 56%, e 70% e 83%, respectivamente.
TR10	Hossein and Sheikh	2022	Envio de grandes volumes de dados para monitoramento médico através de canais com largura de banda limitada.	Empregar dois métodos de compressão em série, aumentando a TC e reduzir volume de dados alterando os bits de informação.	Iniciar a compressão dos dados usando o LZW na camada de aplicação, depois aplicar um processo de reorganização e, por fim, nova compressão com Huffman Coding.	O algoritmo proposto reduz os dados em 37,85% para sinais, imagens e textos.
TR11	Amiri et al.	2022	Monitoramento de pacientes que requerem computação e alto armazenamento em dispositivos vestíveis.	Método de compressão adaptativa para sinais vitais no qual ocorre processo colaborativo de interação entre sensor e servidor em loop.	Através de um ciclo entre sensor e servidor, obtém-se uma representação semântica dos dados. O servidor avalia o conjunto atual de coeficientes recebidos e determina se deve solicitar mais dados ou encerrar com base na precisão desejada.	Aumenta em até 37% e 26% os valores de recall e F1-score em comparação a uma estratégia de compressão otimizada porém estática.
TR12	Idrees and Khlief	2023	Atraso na geração de diagnósticos médicos devido ao grande volume de dados enviados Cloud via rede IoMT.	Utilizar arquitetura Edge-Fog-Cloud para comprimir sinais vitais em gateway localizado na Fog, agregando e comprimindo antes de envio para Cloud.	Os dados recebidos no gateway são agrupados via DBSCAN, codificados com Delta Encoding e, por fim, comprimidos com codificação de Huffman.	Os dados são comprimidos na faixa de 65–85% enquanto outros métodos variam de 40–54%, 65–70%, e 14–56% para LZW, HE e HCLZW, respectivamente.
TR13	Chang and Sobelman	2023	Espaço de armazenamento e o consumo de energia em sensores vestíveis.	Prever os sinais e eliminar redundâncias, comprimindo erro para explorar a repetição de dados.	Técnica híbrida baseada em LPC para predição seguida de compressão por RLE para reduzir o volume .	Ccompressão de até 71% e economia média de energia de 98%, superando Huffman, LZW.
TR14	Vakil and Shirmohammadi	2025	Reduzir tamanho de comprimir dados mantendo a fidelidade dos sinais originais.	Combinar diferentes técnicas de compressão, em arquitetura de codificador baseada em redes neurais.	Combinar codificador com perdas baseado em redes neurais e compressão sem perdas por Huffman, para alta taxa de compressão com baixa distorção.	TC superiores a 5 com baixa distorção (PRD até 0,2), mantendo variações inferiores a 2,5% na classificação dos sinais.
TR15	Kadhum Idrees et al.	2025	Consumo de energia e eficiência da transmissão em redes com recursos restritos	Utilizar algoritmo adaptativo para compressão com perdas para operar em dispositivos IoT.	Aplicar interpolação adaptativa para prever valores e armazenar apenas os que excedem um erro limite, descartando demais para reconstrução na descompressão.	TC até três vezes superiores a métodos comparativos com tempos de compressão inferiores a 5 ms e economia de energia de até 31,4%.
TR16	Baliarsingh and Mohanty	2025	Reduzir volume de dados dos sinais de ECG sem comprometer a integridade clínica.	Decompor dados de sinais vitais em conjunto de modos oscilatórios intrínsecos, com decomposição recursiva de um sinal para representação do espectro de frequência do sinal.	Decompor adaptativamente os sinais em modos intrínsecos por meio do AVMD e refinar resolução tempo-frequência com o uso do WPT. Após é aplicada codificação aritmética	TC de 41,19% com PRD de 0,29, superando métodos Huffman ou LZW em eficiência e qualidade de reconstrução.

TC - Taxa de Compressão, LZW - Lempel–Ziv–Welch, LZMA - Lempel–Ziv–Markov, HCLZW - Agrupamento Hierárquico com LZW, LPC - Codificação Preditiva Linear, RLE - Run Lenght Encode, HCHE - Agrupamento Hierárquico com Codificação Huffman [Idrees and Idrees, 2022].

- **Aplicação:** Nesse âmbito, verifica-se qual é o objetivo principal do trabalho analisado. Qual é o problema que os autores desejam solucionar por meio de técnicas de compressão de dados. Dentro desse contexto, os trabalhos analisados concentraram seus objetivos, basicamente, em quatro domínios.

  - **Energia:** os trabalhos que se concentraram nesse objetivo focam basicamente na transmissão de dados a partir dos dispositivos IoT que realizam a leitura dos sinais vitais. Conforme já citado, até 80% da potência da bateria de um sensor é consumida na transmissão de dados [Latif et al., 2020], ou seja, a compressão de dados constitui uma grande oportunidade para a eficiência energética.
  - **Espaço:** os trabalhos que estão nessa categoria buscam oferecer soluções que gerem eficiência na ocupação do espaço de armazenamento, em especial na *Cloud*. Uma boa gestão dos dados armazenados na *nuvem* é indispensável para não tornar um projeto financeiramente inviável. Essa necessidade é ainda mais relevante no contexto de dados para a saúde, coletados de diversos sensores e normalmente durante longos períodos, gerando grandes volumes de dados.
  - **Desempenho:** o objetivo dos trabalhos classificados nessa categoria está associado à velocidade de transmissão de dados, propondo soluções para lidar com a latência e a ocupação da banda de rede. Dado que o caminho a ser percorrido pelos dados de sinais vitais, desde a origem, nos sensores, até o sistema de destino para monitoração, pode apresentar gargalos e atrasos, degradando a qualidade do diagnóstico. Além disso, para diagnósticos preditivos e para antecipar situações de risco, o tempo necessário para que a informação esteja disponível para análise é de grande relevância.
  - **QoS:** nessa categoria estão os trabalhos que buscam garantir a qualidade das informações de saúde disponíveis para diagnóstico. Ao aplicar técnicas de compressão para dados da área da saúde, a reconstrução dos dados no sistema de destino representa um fator crítico. Desse modo, as propostas que estão nessa categoria buscam modos eficientes de tratar os dados desde a sua origem, para não degradar a informação final que irá embasar o diagnóstico.

- **Método:** neste tópico, os trabalhos relacionados foram classificados pelo modo como buscam alcançar os resultados para resolver o problema proposto. Mais especificamente, busca identificar se o modelo se adapta e ajusta conforme os resultados de *output* ou se é estático, alcançando os resultados conforme parâmetros pré-configurados.

  - **Adaptativo:** nessa categoria estão os trabalhos que tem como base modelos adaptativos, que se ajustam em tempo de execução para conseguir o resultado esperado, alterando dinamicamente suas configurações. Normalmente, são realimentados pela informação já restaurada no destino, conforme uma margem de erro tolerada.

  - **Parâmetros:** estes trabalhos apresentam modelos de compressão que não tem como característica monitorar o *output* dos dados em tempo de execução, ou seja, atuam de modo estático, de acordo com parâmetros pré-configurados.

- **Tipo de compressão:** esta categoria busca classificar os trabalhos quanto aos métodos de compressão adotados. Em grande parte, os trabalhos selecionados utilizam mais de um estágio de compressão; contudo, diferem na escolha das técnicas empregadas, combinando-as de acordo com o objetivo a ser alcançado e com o método adotado para o processo.

  - **Lossy:** os trabalhos que estão categorizados nesse tópico adotam como principal método de compressão o tipo de compressão com perdas. Esse tipo de compressão, conforme aprofundado na Seção 2.3, obtém maiores taxas de redução; contudo, é necessário adotar processos de controle para não degradar a qualidade da informação restaurada.
  - **Lossless:** nessa categoria estão os trabalhos que têm como principal método de compressão o tipo de compressão sem perdas. Esse tipo de compressão também foi aprofundado na Seção 2.3, na qual se descreve que os dados se mantém íntegros para serem restaurados. As técnicas de compressão dessa categoria obtém taxas menores de compressão.
  - **Híbrido:** neste tópico, estão relacionados os trabalhos que combinam as duas técnicas descritas acima. Dessa forma, trata-se de uma abordagem em que o processo de redução de dados ocorre em duas ou mais etapas, contemplando as técnicas de compressão com e sem perdas.

Tabela 5: Análise dos trabalhos relacionados quanto aos objetivos e técnicas utilizadas.

ID	Citação	Aplicação				Método		Tipo de compressão		
		Energia	Espaço	Desempenho	QoS	Adaptativo	Parâmetros	Lossy	Lossless	Híbrida
TR1	[Xiao et al., 2018]	✓				✓		✓		
TR2	[Azar et al., 2019]	✓					✓	✓		
TR3	[Hanoune and Lysmos, 2020]		✓	✓		✓				✓
TR4	[Kalaivani et al., 2020]	✓	✓		✓		✓			✓
TR5	[Passos et al., 2020]	✓				✓				✓
TR6	[Lu et al., 2020]		✓	✓		✓				✓
TR7	[Das and Rahman, 2021]			✓	✓	✓	✓		✓	
TR8	[Idrees et al., 2022]			✓	✓		✓		✓	
TR9	[Khlief and Idrees, 2022]			✓	✓		✓			✓
TR10	[Hossein and Sheikh, 2022]			✓	✓		✓		✓	
TR11	[Amiri et al., 2022]		✓	✓	✓	✓		✓		
TR12	[Idrees and Khlief, 2023]			✓			✓		✓	
TR13	[Chang and Sobelman, 2023]	✓	✓			✓				✓
TR14	[Vakil and Shirmohammadi, 2025]			✓	✓	✓				✓
TR15	[Kadhum Idreesees et al., 2025]	✓		✓	✓		✓	✓		
TR16	[Baliarsingh and Mohanty, 2025]		✓		✓	✓		✓		

Fonte: Elaborado pelo autor.

Legenda:
Energia: eficiência de consumo de energia.
Espaço: redução de espaço para armazenamento.
Desempenho: redução de latência e ocupação de banda.
QoS: foco na qualidade dos dados.
Adaptativo: adapta compressão em tempo de execução.
Parâmetros: parametrizável para obter o resultado esperado.
Lossy: aplica técnica *lossy*.
Lossless: aplica técnica *lossless*.
Híbrida: combina técnicas de compressão.

Os trabalhos selecionados focam basicamente em quatro fatores (Energia, Espaço, Desempenho e QoS), com predominância na busca por um melhor desempenho na transmissão de dados. Para atingir o objetivo, são aplicados diferentes métodos de compressão, com ênfase em métodos de compressão não adaptativos, ou seja, os modelos são previamente parametrizados para um dado desempenho e não em tempo de execução. Verificou-se também que, para os trabalhos voltados para a eficiência no consumo de energia, predomina a compressão aplicada na camada de borda, embarcada nos dispositivos, reduzindo o volume de dados antes da transmissão, a qual constitui a parte do processo que mais consome bateria. Contudo, são soluções desenvolvidas para *hardwares* específicos e, portanto, não são escaláveis.

Nesta base analisada, constatou-se ainda que os trabalhos que utilizam métodos de compressão reconfigurados em tempo de execução tratam de apenas um tipo de sinal vital. Ainda em relação às soluções adaptativas, os trabalhos exploram modos de obter melhores taxas de compressão, visando a um sinal restaurado com níveis de qualidade que estejam dentro de limites pré-definidos. Contudo, não se encontrou uma solução que utilize *thresholds* para diferentes tipos de sinais. Os limites *thresholds* poderiam ser utilizados para definir os intervalos da transmissão dos dados, mais longos ou mais curtos, de acordo com as variações estabelecidas para cada tipo de sinal.

Dado esse contexto, pode-se destacar as seguintes lacunas na área de compressão adaptativa de dados para monitoramento de sinais vitais:

- Embora as soluções propostas apliquem compressão dos dados capturados de sensores IoT, combinando diferentes algoritmos, elas não utilizam um processo adaptativo em tempo de execução para mais de um tipo de sinal vital;
- Os modelos desenvolvidos na camada de borda não são escaláveis e dependem de dispositivos não disponíveis comercialmente;
- Faltam a esses sistemas mecanismos que habilitem taxas de compressão adaptativas, que considerem as diferentes características dos sinais capturados;

Por fim, um *framework* de compressão de dados que seja adaptativo em tempo de execução, escalável com a aquisição de dados a partir de dispositivos comerciais e que trate simultaneamente diferentes tipos de sinais vitais constitui um fator importante para viabilizar a monitorização da saúde. Este *framework* pode ser aplicado à aquisição de dados de diferentes indivíduos distribuídos geograficamente em uma determinada região, considerando o contexto de Cidades Inteligentes. Dentre os trabalhos selecionados, não se verificou nenhum que contemplasse essas características, havendo uma lacuna nessa área.

4 *FRAMEWORK* VSAC

Este capítulo apresenta e detalha o *framework* VSAC — Vital Sign Adaptive Compressor — que representa a principal contribuição desta pesquisa. O *framework* foi concebido como um *framework* adaptativo para a gestão da compressão de sinais vitais em ambientes de saúde conectada, sendo especialmente projetado para operar em *gateways* localizados na camada *Edge* da arquitetura de rede. Nessas condições, os dados são coletados a partir de dispositivos com recursos computacionais limitados e submetidos a técnicas de compressão que otimizam a transmissão e o armazenamento, sem comprometer a integridade clínica dos sinais. O foco central do *framework* é possibilitar um envio inteligente e ajustável dos dados ao sistema de monitoramento, respeitando tanto as restrições da infraestrutura quanto os requisitos de precisão diagnóstica.

Para facilitar a compreensão da proposta, este capítulo está organizado em cinco seções. A Seção 4.1 introduz um caso de uso representativo que ilustra o contexto de aplicação e motiva o desenvolvimento do *framework*. Em seguida, a Seção 4.2 discute as decisões de projeto que nortearam o *framework*, destacando os critérios técnicos adotados. A Seção 4.3 apresenta a arquitetura do VSAC, começando por uma visão conceitual geral e aprofundando-se na descrição das camadas e módulos que compõem o sistema. Por fim, a Seção 4.4 descreve os algoritmos que compõem o *framework*, explicando sua lógica de operação, os mecanismos de compressão utilizados e sua integração com o fluxo de dados no ambiente monitorado.

4.1 Caso de Uso

Para ilustrar melhor o funcionamento do *framework* VSAC, conforme apresentado na Figura 11, imaginemos uma pessoa caminhando por uma avenida movimentada enquanto utiliza um dispositivo vestível (*wearable*) que monitora, em tempo real, seus sinais vitais — como frequência cardíaca, saturação de oxigênio no sangue e pressão arterial. Esses dados são continuamente coletados e transmitidos de forma segura para coletores urbanos estrategicamente posicionados pela cidade, como postes inteligentes, paradas de ônibus ou centros comunitários conectados. Ao alcançar os dispositivos da camada *Edge*, os dados são processados pelo *framework* VSAC, que aplica técnicas de compressão sensíveis ao conteúdo e à criticidade clínica das informações. O *framework* ajusta dinamicamente o nível de descarte de dados com base na relevância do sinal para o diagnóstico, de modo a reduzir o volume transmitido sem comprometer a qualidade das informações essenciais para a análise. A saída resultante é, então, encaminhada para as camadas *Fog–Cloud*, onde seguirá para etapas posteriores de processamento e avaliação.

Na nuvem, o sistema de monitoramento avalia continuamente os sinais vitais recebidos, utilizando critérios clínicos como o score NEWS2 [Smith et al., 2019], que permite identificar alterações fisiológicas indicativas de risco iminente à saúde. O cidadão monitorado pelo dispo-

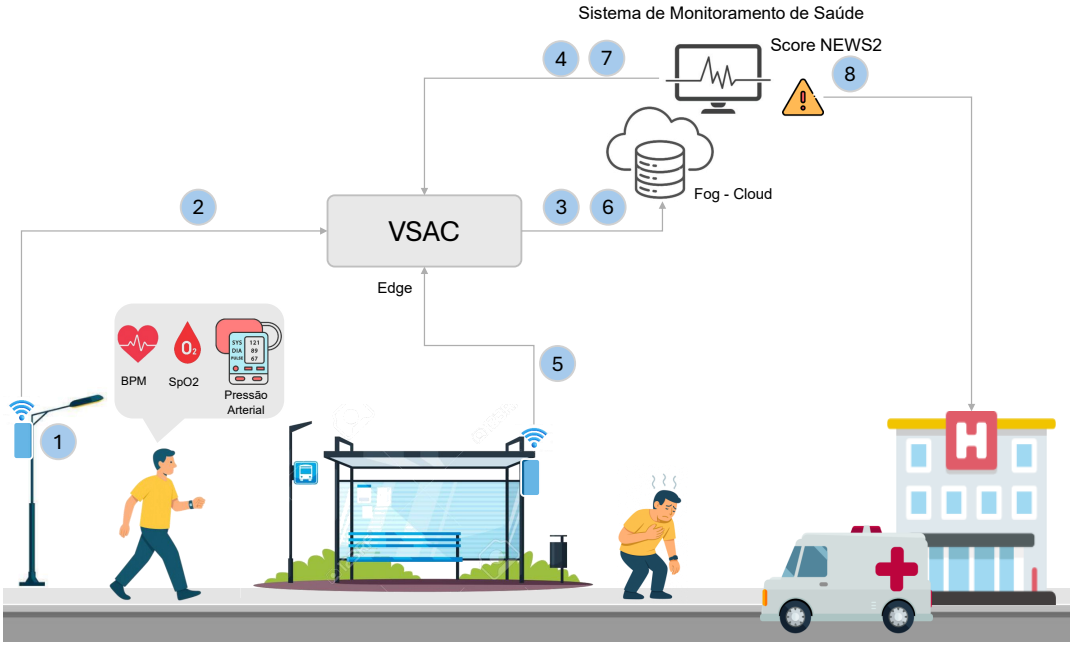


Figura 11: Ilustração do caso de uso do *framework* VSAC no qual um cidadão caminha utilizando um *smartwatch* que monitora continuamente seus sinais vitais. (1) Os dados coletados são transmitidos em uma estação de base próxima. (2) Ao chegar à camada *Edge*, os sinais são processados pelo *framework* VSAC, que aplica compressão adaptativa com base na criticidade clínica da informação. (3) Os dados comprimidos são então encaminhados à camada *Fog–Cloud*, onde são armazenados e analisados por um Sistema de Monitoramento de Saúde. (4) Esse sistema aplica um score de alerta clínico (como o NEWS2) e envia uma notificação de retorno ao *framework* VSAC para ajustar dinamicamente os parâmetros de compressão. (5) Em situações de emergência, os dados são enviados (6) para VSAC e, ao verificar a prioridade, notifica o *framework* (7) e pode também direcionar alertas (8) em tempo real para centros médicos ou unidades de atendimento.

Fonte: Elaborado pelo autor

sitivo *wearable* pode, ao longo do dia, apresentar leves oscilações na frequência cardíaca e na saturação de oxigênio. Inicialmente, o *framework* VSAC mantém uma compressão mais agressiva, priorizando a eficiência da transmissão. No entanto, à medida que os sinais vitais começam a se deteriorar — com quedas progressivas na saturação e aumento da frequência cardíaca — o sistema de monitoramento, ao detectar padrões anormais, envia notificações para o VSAC, que adapta automaticamente o nível de compressão, reduzindo o descarte de dados e preservando mais informações relevantes. Com a piora do quadro, os sinais passam a ser transmitidos com maior fidelidade, permitindo que o sistema, ao aplicar o score NEWS2, identifique rapidamente um estado crítico. Um alerta é, então, gerado, e a central de saúde aciona uma equipe médica que localiza o cidadão com base no ponto de coleta mais recente. O atendimento ocorre ainda em via pública, evitando que a condição evolua para uma insuficiência respiratória grave.

Esse cenário exemplifica a aplicabilidade prática de um *framework* de compressão em um contexto urbano inteligente, onde a infraestrutura distribuída e a conectividade contínua são essenciais para o monitoramento preventivo da saúde populacional. A compressão adaptativa em dois estágios permite não apenas otimizar o uso de recursos computacionais e de rede, mas

também garantir a entrega de informações clínicas relevantes em tempo hábil para a tomada de decisão. Ao integrar a compressão sensível ao contexto com mecanismos de priorização baseados em criticidade, o *framework* contribui significativamente para a construção de sistemas de saúde mais responsivos, eficientes e centrados no bem-estar do cidadão.

4.2 Decisões de Projeto

Esta seção apresenta as principais decisões de projeto que nortearam o desenvolvimento do *framework* VSAC no contexto do projeto Minha História Digital. O foco está na compressão adaptativa e eficiente dos dados de sinais vitais coletados por dispositivos vestíveis em cidades inteligentes, priorizando a integridade clínica da informação e a viabilidade técnica de transmissão em ambientes distribuídos. Conforme destacado na Seção 2.4, o *framework* é implementado na camada *Edge*, sendo responsável pela gestão, compressão e encaminhamento dos dados para as camadas superiores de processamento — *Fog* e *Cloud* —, onde ocorre a análise final e o monitoramento contínuo da saúde dos usuários.

4.2.1 Métodos de compressão

Esse *framework* faz uso de técnicas de compressão de dados como estratégia de otimização da transmissão de arquivos na rede, tendo como base as propriedades intrínsecas desses sinais monitorados e as sinalizações de prioridade recebidas a partir do sistema de monitoramento. Estas sinalizações estão associadas ao estado de saúde de um determinado paciente e indicam ao *framework* tanto a priorização do envio quanto a periodicidade com que os sinais devem ser medidos.

O *framework* VSAC combina diferentes tipos de algoritmos de compressão em uma arquitetura em camadas. Em um primeiro estágio, os dados dos sinais vitais são recebidos em um módulo que aplica compressão adaptativa, viabilizada por meio de um algoritmo *lossy*, estabelecendo tempos de aquisição dos dados dos sinais vitais. Um *loop* entre o sistema de monitoramento e este módulo permite estabelecer prioridades e padrões de tempo para a aquisição dos dados. O estágio seguinte aplica o segundo algoritmo do tipo *lossless* para comprimir os dados antes de enviá-los para a nuvem. Com este *framework*, busca-se uma solução na qual as informações são coletadas em um contexto de distribuição geográfica, mobilidade, grande número de nós na rede, processamento em tempo real e heterogeneidade de dispositivos. Isso permite economizar largura de banda da rede, à medida que muitos nós estão distribuídos em diferentes regiões da cidade, com a população enviando dados para esta camada.

A decisão de integrar estratégias de compressão com perdas (*lossy*) e sem perdas (*lossless*) no *framework* VSAC está fundamentada na necessidade de equilibrar a eficiência de transmissão com a preservação da qualidade da informação em cenários críticos. No domínio da saúde, embora a integridade dos sinais vitais seja essencial para a análise clínica, existe margem para

a aplicação de técnicas *lossy* controladas, especialmente na pré-filtragem de dados redundantes ou de baixa variabilidade. Nesse sentido, o uso de um algoritmo adaptativo de compressão com perdas na primeira etapa permite uma redução significativa do volume de dados, atendendo às restrições de banda típicas de redes de IoT em ambientes urbanos. A etapa seguinte de compressão *lossless* garante que a informação remanescente seja armazenada ou transmitida sem alterações, assegurando rastreabilidade e precisão nos momentos críticos. Essa abordagem híbrida mostra-se eficaz nos testes realizados, com ganhos expressivos na taxa de compressão sem comprometer a utilidade clínica dos dados, validando sua adequação ao cenário de transmissão contínua de sinais biomédicos em infraestruturas distribuídas e sensíveis à latência.

4.2.2 Aquisição de dados

Para o funcionamento do *framework* VSAC, parte-se da premissa de que os indivíduos monitorados utilizam dispositivos vestíveis capazes de coletar sinais vitais e transmitir dados continuamente enquanto se deslocam pela cidade inteligente. Esses dispositivos devem fornecer informações sobre o estado de saúde dos usuários, possibilitando o acompanhamento permanente, a identificação de situações críticas e o acionamento de alertas pelo sistema de monitoramento. Os sinais vitais monitorados incluem frequência cardíaca, temperatura corporal, saturação de oxigênio (SpO2) e frequência respiratória, conforme destacado na Subseção 2.1.1.

O *framework* VSAC é projetado de forma agnóstica quanto ao meio de aquisição e transmissão dos dados, permitindo sua adaptação a diferentes cenários operacionais, como demonstrado na Tabela 6. No entanto, a escolha dessa infraestrutura impacta diretamente fatores como a frequência de envio, latência, compressão necessária e capacidade de adaptação em tempo real. Essa variabilidade reforça a importância de um *framework* flexível e configurável, como o VSAC.

Neste projeto, adota-se como estratégia principal a utilização de estações de coleta distribuídas pela cidade, às quais os dispositivos vestíveis podem conectar-se diretamente via Wi-Fi ou por meio de *smartphones* que atuam como *gateways*. *Smartwatches* com SDKs abertos, como Amazfit, Fitbit Versa e Samsung Galaxy Watch, oferecem suporte a APIs e conectividade Wi-Fi, viabilizando a transmissão direta dos dados. Para garantir a eficiência do monitoramento, os leitores instalados nas estações devem ser capazes de adquirir e transmitir os dados em tempo real à camada *Edge* mais próxima, além de identificar a geolocalização dos indivíduos, permitindo que o sistema detecte situações emergenciais com maior precisão e agilidade.

4.2.3 Arquitetura e funcionamento do *framework*

Quanto à arquitetura adotada para o projeto, ela é composta por três camadas: *Edge, Fog e Cloud*. A aquisição de dados de sinais vitais ocorre na camada *Edge*, que contempla os módulos do *framework* de gestão e compressão de dados, enviando os pacotes compactados

Tabela 6: Possibilidades de aquisição de dados de sinais vitais.

Modo de Aquisição	Alcance	Latência	Consumo Energia	Custo	Adequação ao modelo
Rede celular (4G/5G)	Alta (cobertura nacional)	Baixa (especialmente com 5G)	Alto	Alto (dados móveis)	Alta (ideal para tempo real e adaptação)
Smartphone como gateway	Alta (depende do smartphone)	Baixa a média	Baixo no wearable	Baixo (uso de infraestrutura existente)	Alta (boa para uso pessoal e mobilidade)
Rede LPWAN (LoRaWAN, NB-IoT)	Muito alta (vários km)	Alta	Muito baixo	Baixo a médio	Média (ideal para dados periódicos)
Armazenamento local com sincronização	Limitado ao armazenamento	Alta (não em tempo real)	Muito baixo	Muito baixo	Baixa (sem tempo real)
Wi-Fi público ou comunitário	Média (rede local)	Média	Médio	Baixo	Média (depende da rede)
Drones ou robôs móveis	Variável (conforme missão)	Alta	Baixo (no wearable)	Alto	Baixa (uso pontual)

para a camada *Fog* , enquanto o armazenamento de informações ocorre na camada *Cloud*. Nesse sentido, o foco deste trabalho está na camada *Edge*, onde o *framework* será desenvolvido. Os dados originados dos sensores serão simulados por meio de *datasets*, abstraindo os diferentes tipos de dispositivos de IoT e os protocolos de comunicação. Do mesmo modo, o tratamento dos dados na camada *Cloud* não será desenvolvido, bem como os *dashboards* e o processamento dos dados para gerar diagnósticos de saúde. Quanto aos sinais de alerta e notificações que devem ser originados da *Cloud* para testes de priorização no módulo de gestão de dados, serão feitos por meio de simulação. Tendo essas premissas assumidas, a Figura 12 apresenta o diagrama do *framework* VSAC desenvolvido na camada *Edge*.

Para este *framework*, foi desenvolvido um processo de compressão de dados dividido em duas etapas. No primeiro estágio, um algoritmo é responsável por executar um processo de seleção de dados com base em *thresholds* configurados a partir de um módulo de gestão de notificações. Isso permite descartar dados redundantes ou irrelevantes, de acordo com o que é sinalizado pelo sistema de monitoramento de saúde. Os dados resultantes do primeiro estágio de compressão são armazenados em um módulo denominado *Assemble*, que agrupa os pacotes e os direciona para o segundo estágio de compressão. O segundo estágio do módulo coleta os dados do *Assemble* e, a partir de arquivos no formato *JavaScript Object Notation* (JSON), aplica novamente um algoritmo de compressão antes de enviá-los para o sistema de monitoramento na *Cloud*. Este arquivo comprimido é restaurado na *Cloud*. Entretanto, neste trabalho, não se aborda do processo de descompressão de dados para monitoramento.

No primeiro estágio de compressão de dados, será adotado o método com perdas Swinging Door Trending (SDT). Este algoritmo de compressão tem como característica permitir ajustes na taxa de compressão em tempo de execução, possui baixa complexidade computacional e utiliza uma tendência linear para representar uma quantidade de dados [Correa et al., 2019]. Essas características do SDT permitem que se tenha um processo adaptativo na primeira etapa de compressão, na medida em que se pode ajustar o parâmetro de $desvio$ conforme as sinalizações recebidas do sistema de monitoramento. O parâmetro $desvio$ representa a diferença máxima

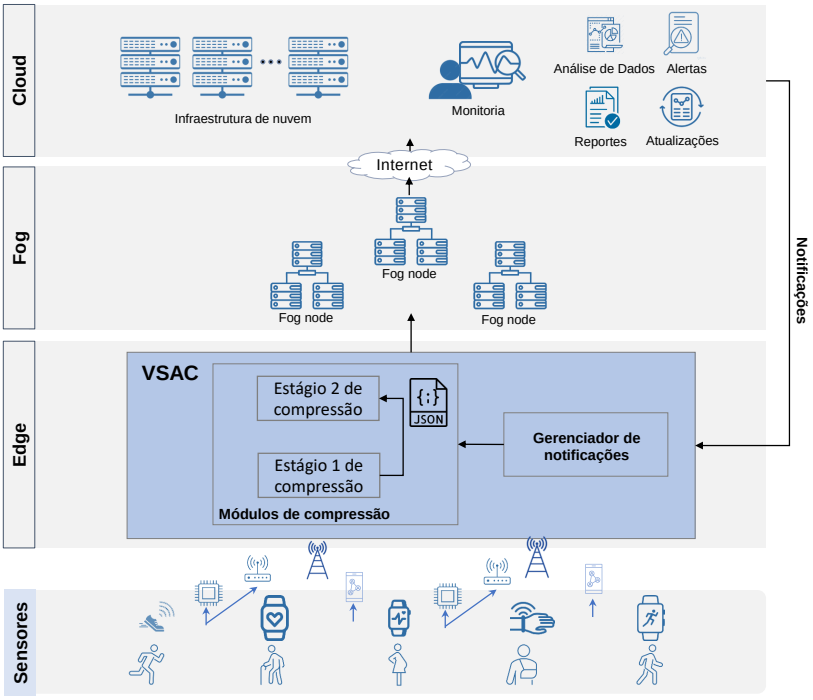


Figura 12: *framework* de compressão de dados VSAC desenvolvido na camada *Edge*, recebe dados dos sensores distribuídos, processa os dados conforme notificações recebidas do sistema de monitoração desenvolvido na camada *Cloud*.

Fonte: Elaborado pelo autor

entre a amostra atual e a tendência linear atual utilizada para representar os dados previamente coletados, conforme detalhado na Subseção 2.3.1.

4.2.4 Formatação dos dados

O formato de arquivo JSON foi escolhido pois permite estruturar dados em formato texto para serem utilizados em diferentes tipos de sistemas. É um formato simples e leve que oferece mais agilidade no processamento Por ser no formato de texto, existem vários algoritmos disponíveis que oferecem altas taxas de compressão com baixo consumo de processamento. Dentre esses algoritmos, destacam-se o Huffman Coding e o LZW, ambos amplamente conhecidos e mencionados na literatura por seu bom desempenho em arquivos com formato de texto. Contudo, eles possuem características diferentes, conforme o tamanho do arquivo a ser comprimido. Também são de fácil implementação e demandam pouco processamento [Sharma, 2020].

Ainda de acordo com os testes realizados em [Sharma, 2020], para arquivos de texto com tamanho a partir de 2,2 kB até 33,2 kB, a taxa de compressão do Huffman Coding é melhor do que a obtida com o LZW. Porém, nos testes com arquivos de tamanho a partir de 198,8 kB, o desempenho do LZW passa a ser superior, não apenas em taxa de compressão, mas também em tempo de compressão e descompressão. Na Tabela 7 estão apresentados os dados dos ensaios para os dois códigos de compressão, onde se percebe a diferença entre ambos à medida que se

Tabela 7: Comparativo de desempenho dos métodos Huffman Coding e LZW para diferentes tamanhos de arquivo texto, sendo TC= Taxa de Compressão, T= Tempo de Compressão e TD= Tempo de Descompressão.

	Huffman Coding			LZW		
Tamanho arquivo de entrada	**TC (%)**	**T (seg)**	**TD (seg)**	**TC (%)**	**T (seg)**	**TD (seg)**
2.2 kB	54,20	0,0	0,0156	105,56	0,0	0,0
6.7 kB	54,20	0,0	0,069	84,46	0,0	0,0
33.2 kB	**54,205**	**0,015**	**0,231**	49,94	0,0156	0,0156
198.8 kB	54,204	0,0625	1,402	24,219	0,052	0,0159
994.1 kB	54,203	0,601	7,076	**12,458**	**0,284**	**0,115**
3.9 MB	54,203	8,574	30,029	6,319	1,622	1,624
7.8 MB	54,203	34,266	55,879	5,062	3,224	6,852
15.5 MB	54,203	123,664	115,858	3,604	8,560	29,734
31.1 MB	54,203	548,473	237,528	2,566	17,352	103,317
62.1 MB	54,203	1.945,389	470,933	1,827	53,198	469,634

TC (Taxa de Compressão) = (C2/C1) *100%, onde C1= Tamanho do arquivo de entrada antes da compactação e C2= Tamanho do arquivo após a compressão.

aumenta o tamanho do arquivo e da entrada.

Tais características, próprias de cada algoritmo, podem ser utilizadas no *framework* VSAC, permitindo que se opere com diferentes tamanhos de arquivos. Isso constituiu uma vantagem, na medida em que se podem gerar arquivos menores a partir de uma pessoa que precisa ser tratada com prioridade, acelerando o envio ao sistema de monitoramento. E, para os demais pacotes de dados de sinais vitais, estes podem ser agrupados em arquivos maiores para serem compactados e transmitidos utilizando o algoritmo de melhor desempenho. Dado que o período recomendável para coleta de dados para análise de saúde deve ser de no mínimo 18 horas [Cygankiewicz and Zareba, 2013], se adotarmos para este modelo um padrão de 24 horas de coleta, com cadência de 5 minutos — que consiste em um intervalo curto entre as leituras [Sammito and Böckelmann, 2016] — isso corresponde a 288 leituras e resulta em um arquivo com tamanho de 7 kB. Ou seja, para os dados de apenas um indivíduo, o arquivo aponta para a escolha do algoritmo Huffman Coding. Em resumo, o *framework* utiliza os dois algoritmos de compressão: LZW para arquivos com tamanho próximo a 1 MB e Huffman Coding para arquivos em ordem de prioridade e de menor tamanho.

O modelo concentra-se em computação de alto desempenho, e o fator segurança é extremamente importante; contudo, não foi explorado em profundidade. No entanto, na versão atual do modelo VSAC, já estão planejados controles de privacidade. Para tanto, o modelo VSAC incorpora técnicas de pseudonimização de dados. Ao remover ou mascarar informações de identificação pessoal, o sistema protege dados sensíveis, ao mesmo tempo em que permite que as principais necessidades de monitoramento sejam atendidas. Essa abordagem está alinhada aos princípios descritos no *General Data Protection Regulation* (RGPD). Especificamente, Artigo 6(1), Artigo 25 e Artigo 5(1)(c). Quanto à Lei Geral de Proteção de Dados (LGPD), no Art. 5º, inciso XI.

4.2.5 Adoção de Compressão Adaptativa em Cenários Reais

Em ambientes como hospitais, residências assistidas, unidades móveis de saúde ou eventos públicos, centenas ou milhares de indivíduos podem estar sendo monitorados simultaneamente. Ainda que a quantidade de dados gerada por cada paciente seja relativamente pequena, o volume agregado tende a crescer exponencialmente, sobrecarregando a infraestrutura de rede, os dispositivos intermediários (*Edge/Fog*) e os servidores em nuvem. O *framework* VSAC enfrenta diretamente esse desafio ao aplicar técnicas de compressão adaptativa e seletiva, que permitem uma redução significativa do volume de dados transmitidos sem comprometer a integridade clínica dos sinais mais relevantes.

Além disso, é importante reconhecer que a avaliação do estado de saúde de um indivíduo não depende apenas dos sinais vitais, mas também de uma série de fatores contextuais que podem interferir nas medições e no risco clínico. Entre esses fatores estão a temperatura ambiente, a umidade do ar, a qualidade do ar (níveis de CO_2, partículas e alérgenos), a altitude local, o nível de atividade física, o tempo de exposição ao calor ou frio, a presença de ruído ambiental e até mesmo o horário do dia — todos os elementos que podem modular a resposta fisiológica do organismo.

A incorporação desses parâmetros adicionais tende a gerar ainda mais dados e reforça a necessidade de um *framework* como o VSAC, capaz de gerenciar, adaptar e priorizar eficientemente o fluxo de dados em tempo real, assegurando que a informação crítica chegue de forma oportuna ao sistema de saúde. Esse contexto evidencia a necessidade de uma solução escalável, adaptativa e sensível ao contexto, preparada para lidar com cenários complexos compostos por múltiplos fatores da medicina moderna.

4.2.6 Síntese das decisões de projeto e aplicabilidade

As decisões descritas ao longo deste capítulo demonstram a preocupação em alinhar o *framework* VSAC às demandas de sistemas de monitoramento de saúde distribuídos, com requisitos de baixa latência, escalabilidade e preservação da qualidade da informação. A integração entre compressão com perdas e sem perdas, combinada com uma arquitetura modular e adaptável, permite que o *framework* opere eficientemente em ambientes urbanos dinâmicos e sob diferentes restrições de conectividade.

Além disso, o *framework* considera diversos modos de aquisição e transmissão de dados, sendo agnóstico quanto à tecnologia empregada, o que o torna aplicável tanto em contextos domiciliares quanto hospitalares, em eventos públicos ou em regiões de alta densidade populacional. Com base nesses fundamentos, as decisões consolidadas a seguir orientam a implementação prática e a validação do protótipo do *framework*.

Em resumo, as seguintes decisões de projeto compõem o *framework* VSAC:

- Para a avaliação dos resultados do protótipo do *framework* VSAC, os dados dos sinais

vitais provenientes dos dispositivos IoT serão considerados por meio de *datasets*;

- O *framework* será desenvolvido na camada *Edge*, abstraindo as redes de comunicação, o formato dos arquivos e os protocolos de transferência entre os sensores e a camada *Fog*;

- Os dados dos sinais vitais processados pelo *framework* serão comprimidos em duas etapas, sendo que o formato do arquivo de saída do primeiro estágio de compressão será JSON. Este formato será, então, comprimido através da codificação LZW ou Huffman no segundo estágio e enviado ao sistema de monitoramento.

- As estratégias adotadas para a gestão de dados são transparentes para os usuários; ou seja, as decisões de priorização do tipo de sinal vital e respectivas cadências de medidas não demandam nenhuma ação dos usuários ou mesmo alertas informativos.

4.3 Arquitetura VSAC

Esta seção detalha a arquitetura do *framework* VSAC, descrevendo suas camadas, componentes e o processo de fluxo de dados. Primeiramente, será apresentada uma visão geral do *framework*, conceituando seus módulos e descrevendo suas respectivas entradas e saídas de dados, bem como sua função no *framework*, tornando mais claro o entendimento dos processos. A seguir, será detalhado o funcionamento de cada módulo, o processamento dos dados e o modo como eles se comunicam entre si.

4.3.1 Visão geral

O *framework* de VSAC tem como objetivo realizar a gestão de dados de sinais vitais em um sistema de monitoramento de saúde, a fim de gerar eficiência no consumo de recursos de rede e no espaço de armazenamento, mantendo a qualidade da informação disponível para a tomada de decisão em diagnósticos e para gerar alertas de saúde. Para isso, a arquitetura apresentada na Figura 13 foi projetada com componentes que fazem parte de um processo de tratamento dos dados, desde a origem nos sensores até serem disponibilizados para as aplicações de inteligência e diagnóstico. Estes componentes serão descritos a seguir, de acordo com o caminho que os dados seguem ao longo do módulo.

- **Adaptive Compress**: é o componente responsável pela primeira etapa de redução do volume de dados. Neste ponto, são inseridos os sinais vitais recebidos dos sensores, bem como as sinalizações recebidas do módulo **SetUp**. Essas sinalizações irão definir os *thresholds* e a cadência de seleção de dados. A compressão desenvolvida nesse componente é do tipo *Lossy*, ou seja, valores repetitivos ou com pouca variação, conforme limites estabelecidos a partir do **SetUp** , serão descartados. Esse fator torna o módulo

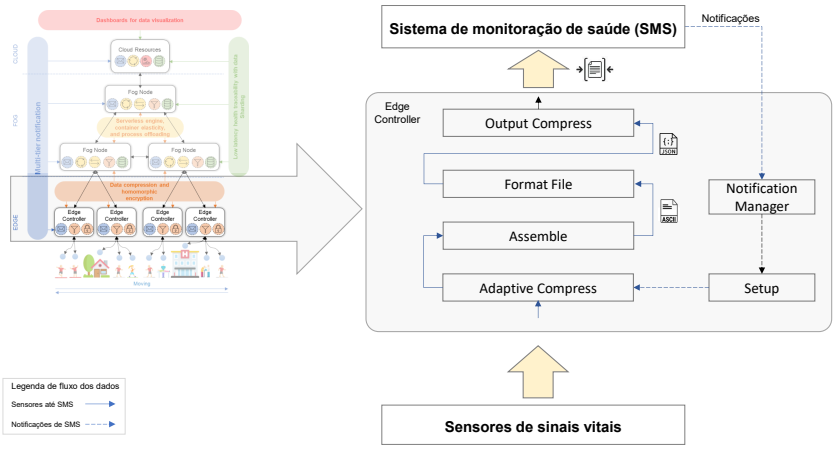


Figura 13: O *framework* VSAC em destaque a partir da camada *Edge*, onde estão posicionados os *Edge Controllers* no Projeto MHD. Os componentes que formam o *framework* integram um processo em que a informação dos sinais vitais é recebida em **Adaptive Compress**, responsável pelo primeiro estágio de compressão adaptativo, que irá executar a redução de dados conforme parâmetros de **SetUp**. Este componente é ajustado conforme as notificações recebidas a partir do sistema de monitoramento em **Notifiction Manager**. Após a compactação dos dados em **Adaptive Compress**, são enviados para *Assemble*, que irá agrupá-los em pacotes e disponibilizar em formato de texto ASCII para **Format File**, responsável pela formatação dos arquivos. Os arquivos agora em formato JSON seguem para o último estágio de compressão em **Output Compress**, para então serem transmitidos enviados para o sistema de saúde, onde serão restaurados, armazenados e utilizados para diagnóstico, sinalizações e monitoração.
Fonte: Elaborado pelo autor

adaptativo, na medida em que ajusta os níveis de seleção e descarte de dados de acordo com as sinalizações originadas do **sistema de monitoramento**;

- **Notification Manager**: este componente tem a função de receber as notificações do Sistema de Monitoramento de Saúde desenvolvido na *Cloud* e encaminhá-las para o módulo **SetUp**. A partir deste módulo, é possível individualizar o tipo de sinal vital a ser priorizado e a respectiva cadência de leituras para um indivíduo específico;

- **SetUp**: este componente é responsável por traduzir as notificações recebidas do **Notification Manager** em comandos que servem para o ajuste dos *thresholds* utilizados no componente **Adaptive Compress**;

- **Assemble**: este componente realiza o agrupamento dos dados que são recebidos do módulo de **Adaptive Compress**. Este componente, por meio de técnicas de concatenação de dados, tem como objetivo estruturar um arquivo no formato de texto padrão ASCII. Este arquivo de saída contém os dados já comprimidos, formatados de acordo com o padrão definido pelo projeto, a fim de serem interpretados no sistema de monitoramento. Neste componente, também será inserido *header* no arquivo com informações de priorização que servirão para as decisões de compressão nos próximos estágios do *framework*;

- **Format file**: este componente tem como função formatar os dados recebidos do compo-

nente **Assemble** - ASCII para o padrão JSON, que corresponde ao formato do arquivo de saída deste módulo;

- **Output compress**: este é o último estágio do *framework* de gestão de dados; nele será aplicado o método de compressão *Lossless*. Nessa etapa, os dados são compactados sem perda de informações, uma vez que a primeira etapa de compressão do *framework* já realizou as reduções de tamanho com base no descarte de dados. O método de compressão aplicado nesta etapa é o LZW ou Huffman Coding, dependendo do tamanho do arquivo a ser comprimido, o qual é identificado por meio de um *header*. Os arquivos de menor tamanho são encaminhados para o algoritmo de Huffman e consistem em dados que estão sendo tratados com prioridade, não passando pelo processo de agrupamento de dados em **Assemble**, que demanda tempo de processamento. Os demais arquivos, sem prioridade, são direcionados para LZW, que apresenta melhor desempenho, conforme visto na Seção 4.2. Ambos os algoritmos são do tipo sem perdas, com alta taxa de compressão para esses tipos de arquivo e para o tamanho estimado dos pacotes [Sharma, 2020]. Os arquivos comprimidos que resultam deste módulo são transmitidos para a *Cloud* para serem armazenados e utilizados pelo sistema de monitoramento.

Tabela 8: Módulos que compõem o *framework* com as respectivas entradas e saídas.

Módulo	**Entrada**	**Tipo de computação**	**Tipo de processamento de dados**	**Latência**		**Saída**
				Nível	**Requerido**	
Adaptive Compress	Sinais vitais e parâmetros	Tempo de execução	Compressão de dados Lossy, processamento de sinais e filtro adaptativo.	Baixo	Millisegundos	Sinais vitais comprimidos
SetUp	Notificações de prioridade	Tempo de execução	Configuração de parâmetros e refinamento.	Baixo	Milisegundos	Parâmetros para ajustes de compressão
Notification Manager	Sistema de monitoramento	Tempo de execução	Orientado a eventos e filtro de dados.	Médio	Segundos	Prioridade das notificações
Assemble	Dados de sinais vitais comprimidos	Batch	Agrupamento de dados, formatação e inclusão de metadados.	Médio	Segundos	Pacotes de dados agrupados
Format File	Pacotes de dados agrupados	Batch	Serialização de dados e formatação.	Médio	Segundos	Arquivos no formato JSON
Output Compress	Arquivos de sinais vitais no formato JSON	Batch	Compressão de dados Lossless (LZW ou Huffman Coding)	Médio	Segundos	Arquivos com sinais vitais comprimidos

Fonte: Elaborado pelo autor

O *framework* VSAC está posicionado na camada mais próxima dos sensores, descartando dados redundantes de acordo com cada tipo de sinal vital e as sinalizações do módulo de decisão. Os módulos seguintes ao da compressão adaptativa irão agrupar e formatar os dados de modo que estejam estruturados para serem novamente comprimidos e transmitidos para armazenamento e uso no sistema de monitoramento. Uma visão geral mais abrangente das propriedades computacionais de cada módulo é fornecida na Tabela 8. A tabela apresenta os dados de entrada e saída de cada módulo, juntamente com seu tipo de computação (tempo de execução ou *batch*), tipo de processamento de dados, nível de latência (baixo, médio ou alto) e requisitos específicos de latência (milissegundos ou segundos).

4.3.2 Visão detalhada

Seguindo o entendimento dos módulos que compõem o *framework* VSAC, partindo do ponto de entrada dos dados, os pacotes oriundos dos leitores distribuídos na cidade inteligente seguem uma estrutura que contempla as necessidades do sistema de monitoramento da saúde. Essa estrutura está apresentada na Tabela 9, detalhando as informações que compõem cada *string* de dados coletados. Na Figura 14 é apresentado o formato dos pacotes recebidos no *framework*, seguindo o padrão de texto em ASCII. eles são originados das estações de base de coleta de dados, contendo as *strings* com os sinais vitais de um grupo de pessoas monitoradas.

Conforme visto na Seção anterior, os pacotes de dados contendo as *strings* de cada indivíduo são recebidos no módulo **Adaptive Compress**. Na Figura 15, detalhando os componentes desse módulo, verifica-se que a entrada dos dados ocorre no componente denominado ***Extractor***, responsável por extrair de cada ID apenas os sinais vitais, mantendo as informações de localização e horários de coleta na *string* original. O respectivo ID (ID_i) e o período de leituras dos sinais (P_i) da pessoa que está sendo monitorada também são encaminhados juntamente com os sinais vitais, formando assim o pacote Pck_i. Este pacote é, então, transmitido para o módulo de compressão *Swinging Door Trending* - **SDT** - que aplicará o algoritmo adaptativo nos sinais vitais, conforme a parametrização recebida de **SetUp**. Este módulo tem uma comunicação de duas vias com **SDT**, recebendo o ID_i de cada pacote e retornando os parâmetros de compressão ajustados para aplicar no algoritmo. A seguir, em ***Integrator***, os dados compactados de cada tipo de sinal vital serão integrados ao respectivo ID de origem, de modo que FC_i, TC_i, SO_i e FR_i serão substituídos pelos novos pacotes de dados FC_i', TC_i', SO_i' e FR_i'.

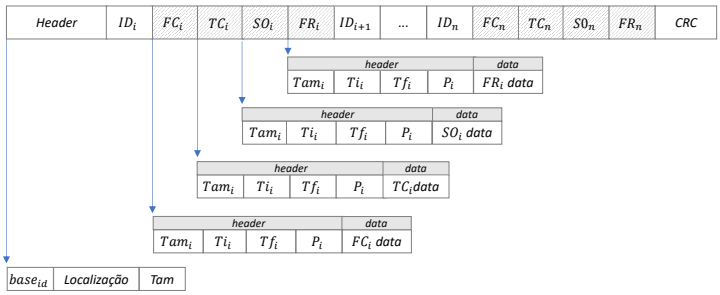


Figura 14: Formato de um pacote recebido de uma estação de coleta de dados com as *strings* contendo os sinais vitais de um grupo de pessoas monitoradas. Sendo STX o caractere indicador de início do arquivo, $Base$ o identificador da estação de coleta de dados e em ID_i estão as *strings* com as informações de saúde das pessoas monitoradas. Cada *string* contendo os sinais vitais de uma pessoa monitorada tem como finalizador o caractere ETX

Fonte: Elaborado pelo autor

Os parâmetros do módulo **SDT**, conforme apresentados na Seção 2.3.1, são o tempo máximo de compressão (T_m) e o desvio de compressão ($desvio$). Sendo que, quanto maior o T_m e $desvio$, maior será a taxa de compressão. Estes parâmetros partem de um valor *default* e são ajustados através do componente **SetUp**, conforme as notificações recebidas e a frequência de

Tabela 9: Campos que formam a *string* de dados com as informações de sinais vitais coletados de uma pessoa monitorada.

Nome do campo			Formato	Tamanho (bytes)	Descrição
Header	Base ID		nnnnnnnn	8	Identificação da base de aquisição
	Localização		+/-dd.nnnnnn;+/-dd.nnnnnn	21	Localização geográfica da estação base
	Tam		nnnnnnnn	8	Tamanho do pacote de dados
Dados paciente	ID		nnnnnnnn	8	Identificação da pessoa monitorada
	FC	Tam	nnnnnnnn	8	Tamanho do pacote de frequência cardíaca
		Ti	ddmmaaaahhmmss	14	Data/hora de início de leitura dos dados
		Tf	ddmmaaaahhmmss	14	Data/hora de fim de leitura dos dados
		P	ssssss	6	Período/Cadencia de leitura dos dados (seg)
		Dados	nnn...nnn	var	Dados de frequência cardíaca
	TC	Tam	nnnnnnnn	8	Tamanho do pacote de temperatura corporal
		Ti	ddmmaaaahhmmss	14	Data/hora de início de leitura dos dados
		Tf	ddmmaaaahhmmss	14	Data/hora de fim de leitura dos dados
		P	ssssss	6	Período/Cadencia de leitura dos dados (seg)
		Dados	nnn...nnn	var	Dados de temperatura corporal
	SO	Tam	nnnnnnnn	8	Tamanho do pacote de saturação de oxigênio
		Ti	ddmmaaaahhmmss	14	Data/hora de início de leitura dos dados
		Tf	ddmmaaaahhmmss	14	Data/hora de fim de leitura dos dados
		P	ssssss	6	Período/Cadencia de leitura dos dados (seg)
		Dados	nnn...nnn	var	Dados de saturação de oxigênio no sangue
	FR	Tam	nnnnnnnn	8	Tamanho do pacote de frequência respiratória
		Ti	ddmmaaaahhmmss	14	Data/hora de início de leitura dos dados
		Tf	ddmmaaaahhmmss	14	Data/hora de fim de leitura dos dados
		P	ssssss	6	Período/Cadencia de leitura dos dados (seg)
		Dados	nnn...nnn	var	Dados de frequência respiratória
CRC	Tam		nnn.nnn	16	Checksum calculado para cada pacote

Fonte: Elaborado pelo autor

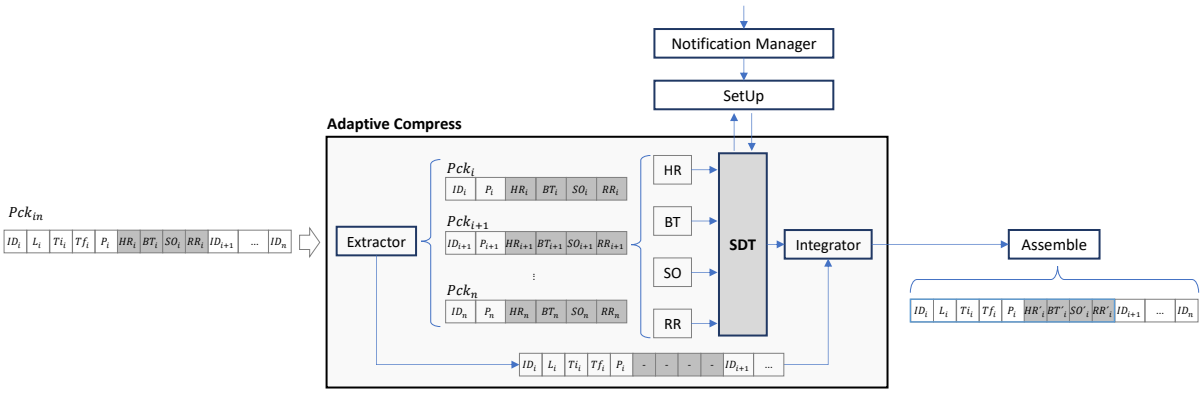


Figura 15: Módulo de compressão adaptativa, *Adaptive Compress* recebe pacotes Pck_{in} originados dos sensores e são processados no módulo *Extractor*. Aqui, são separados de cada *ID* apenas os dados de sinais vitais e enviados para serem compactados no *SDT* de acordo com a parametrização de *SetUp*. Os dados compactados resultantes desta etapa são então reagrupados em *Integrator* no pacote original, que havia sido previamente separado em *Extractor*.

Fonte: Elaborado pelo autor

monitoramento dos sinais descrita na Subseção 2.1.1.

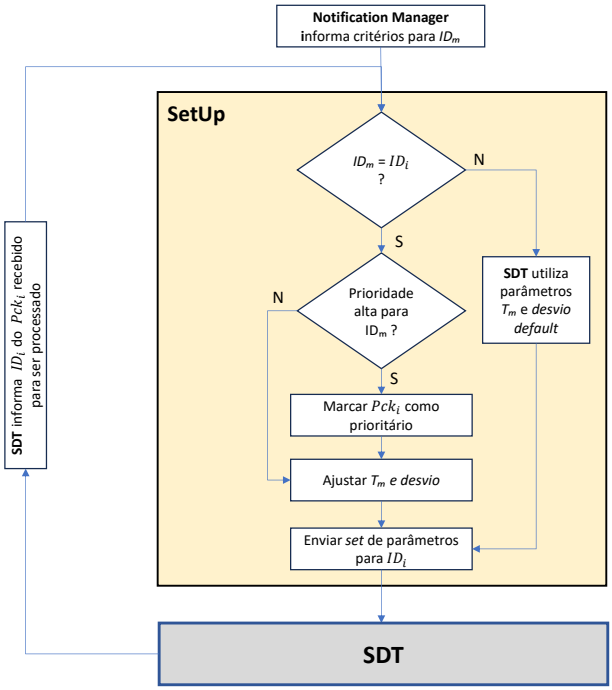


Figura 16: Fluxograma para o processo de decisão de ajuste dos parâmetros do módulo **SDT**.
Fonte: Elaborado pelo autor

Na Figura 16 é apresentado o algoritmo para o processo de decisão de ajuste dos parâmetros do **SDT**. Este processo ocorre no módulo **SetUp** e inicia a partir de parâmetros recebidos do **Notification Manager** e **do SDT**. A partir de **SDT** , é informado o ID_i, extraído de Pck_i e, a partir do **Notification Manager** , recebe ID_m. Com estas duas informações, verifica-se se o pacote a ser processado precisa ser tratado com prioridade e, se o retorno dessa comparação for negativo, então são mantidos os valores *default* do módulo de compressão - T_m e $desvio$. Se o resultado da $ID_i = ID_m$ for positivo, o próximo passo é verificar se há necessidade de priorizar o processo de compressão de Pck_i. O processo de priorização consiste em atribuir alta prioridade ao pacote no processo que consome a fila para compressão e, com isso, promover o envio antecipado para o módulo **Assemble**, acelerando assim o processo de transmissão para a *Cloud*.

O passo seguinte, após a priorização, é efetuar ajustes nos parâmetros de compressão. Para isso, os parâmetros T_m *e* $desvio$ de **SDT** são modificados de acordo com o que foi recebido do **Notification Manager**. Esse novo *set* de parâmetros, relativo ao Pck_i, será enviado para o módulo de compressão.

Os pacotes resultantes do **Adaptive Compress** são recebidos no **Assemble** e, então, podem seguir dois caminhos diferentes, de acordo com seu nível de prioridade. Se os dados estiverem identificados com prioridade alta, precisam ser enviados para o sistema de monitoramento de saúde o mais rápido possível e, portanto, não passarão pelo processo de agrupamento de arquivos que ocorre no **Assemble**. Os demais arquivos, que não têm um nível elevado de prioridade,

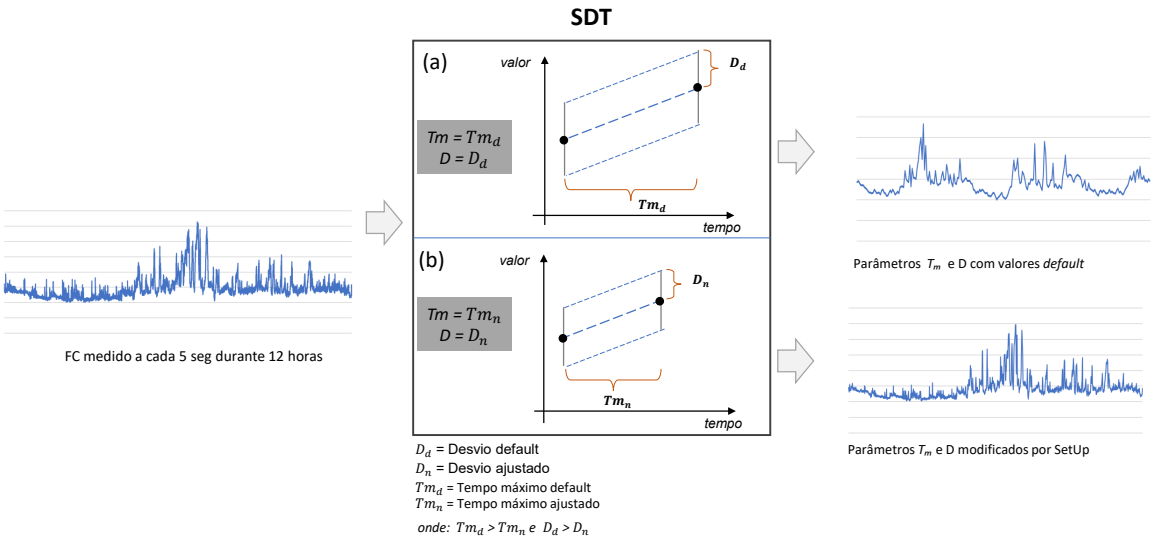


Figura 17: Módulo **SDT** atuando em tempo de execução com diferentes valores de parâmetros $desvio$ e Tm. Em **(a)** os parâmetros estão com valor *default* e **(b)** os valores foram ajustados conforme notificação. No exemplo acima, para o cenário **(b)**, a notificação recebida em **SDT** faz com que os valores dos parâmetros sejam ajustados para uma nova janela de compressão, com área menor que em **(a)**, levando a uma taxa de compressão maior.

Fonte: Elaborado pelo autor

seguem o processo planejado para serem agrupados.

O componente **Assemble**, à medida que recebe os pacotes, tem a função de agrupá-los até que o arquivo atinja um tamanho pré-definido. Esse tamanho do arquivo resultante corresponde ao valor que permite a maior eficiência de compressão para o método *Lossless* adotado no segundo e último estágio de compressão do *framework*. Conforme visto na Seção 4.2, os valores da **Taxa de Compressão**, **do Tempo de Compressão** e **do Tempo de Descompressão** para o algoritmo LZW, ao processar arquivos de texto, indicam que a Taxa de Compressão aumenta à medida que aumenta o tamanho do arquivo; contudo, é necessário estabelecer uma linha de corte, ou seja, um tamanho limite para garantir um processo contínuo entre todos os componentes do *framework*. Sem esse limite de tamanho, o processo ficaria concentrado entre **STD** e **Assemble**, gerando atraso no envio de pacotes para o sistema de monitoramento na *Cloud*.

Quanto aos fatores de tempo de compressão e tempo de descompressão, conforme Sharma [2020] e apresentado na Tabela 7, para tamanhos de arquivo próximos a 1 Megabyte (MB), os tempos para compressão e descompressão são de 0,284 segundos e 0,115 segundos, respectivamente. Para tamanhos de arquivo maiores, verifica-se que os tempos se elevam em uma ordem não proporcional ao ganho obtido com a taxa de compressão; ou seja, o ganho que se obtém ao reduzir ainda mais o arquivo acaba sendo penalizado com uma elevação desproporcional no tempo de compressão e descompressão. Com isso, o tamanho dos arquivos adotado neste trabalho como saída do módulo **Assemble** será de 1 MB. Esse valor será dado por um parâmetro estático utilizado pelo módulo, denominado $SzFileCluster_{max}$. Com isso, o valor pode ser

Tabela 10: Formato do *header* do arquivo resultante da saída de **Assemble**.

Campo	Formato	Tamanho	Descrição
FileSz	nnnnnnnn	8	Tamanho do arquivo em bytes
Date	aammddhhmmss	12	Data de criação do arquivo
FileRate	nnn	3	Nível de prioridade do arquivo
IDnum	nnnnnn	8	Quantidade de IDs presentes no arquivo

Tabela 11: Nomenclaturas utilizadas para os atributos do arquivo de saída do módulo **Format File**.

Atributo	Obrigatório	Tipo	Descrição
base_id	Sim	texto	Identificador da estação base de coleta de dados
location	Sim	numérico	Coordenadas geográficas da estação base
user_id	Sim	texto	Identificador do usuário
time_start	Sim	texto	Data e hora de início da coleta de dados
time_end	Sim	texto	Data e hora de fim da coleta de dados
time_sample	Sim	numérico	Cadência de leitura dos dados
heart_rate	Sim	texto	String com dados da frequência cardíaca
temperature	Sim	texto	String com dados temperatura corporal
spo2	Sim	texto	String com dados da saturação de oxigênio
breath_rate	Sim	texto	String com dados da frequência respiratória

Fonte: Elaborado pelo autor

alterado e ajustado para extrair os melhores resultados do algoritmo de compressão.

Os arquivos de saída do módulo **Assemble** possuem um *header* contendo informações que serão necessárias para a decisão do tipo de compressão no segundo estágio do *framework* VSAC. A Tabela 10 apresenta quais são os campos deste *header* e suas respectivas descrições. Em particular, o campo *prioridade* tem como origem a informação trazida do módulo **SetUp**, com a sinalização recebida do **Notification Manager** , conforme o fluxograma da Figura 16.

Dando seguimento ao processo de compressão do *framework* VSAC, os arquivos de saída do módulo **Assemble** são recebidos em **Format File**. Neste componente, o arquivo em formato de texto será convertido para o formato JSON, conforme descrito na Subseção 4.2. Conforme exemplificado na Figura 18, o arquivo de saída consiste no formato padrão JSON, contendo os atributos descritos na Tabela 11. Neste arquivo, os dados dos usuários estão agrupados em *arrays* de acordo com a estação base em que foram coletados, conforme o atributo de identificador "base_id". Cada usuário é identificado pelo atributo "user_id"seguido pelos respectivos sinais vitais dispostos em *strings*.

No último módulo do *framework*, **Output Compress**, ocorre a segunda etapa de compressão, agora sem perdas, dado que o primeiro estágio de compressão já efetuou o descarte de dados redundantes, ou seja, uma compressão com perdas. Para essa etapa, conforme descrito na Subseção 4.2, os arquivos sinalizados como prioritários e com tamanho de até 33 kB serão comprimidos através do algoritmo Huffman. Os demais arquivos serão comprimidos através do algoritmo LZW.

A Figura 19 apresenta o fluxo de decisão quanto ao método de compressão a ser aplicado.Na entrada do módulo, será realizada uma leitura do *header* do arquivo e será verificado o nível de prioridade do arquivo recebido, informado através do parâmetro $FileRate$. Esse parâmetro é comparado com o valor estabelecido no parâmetro P_H, que indica o limite do nível de prioridade adotado pelo *framework*. Se a prioridade não atender ao primeiro critério, o arquivo é

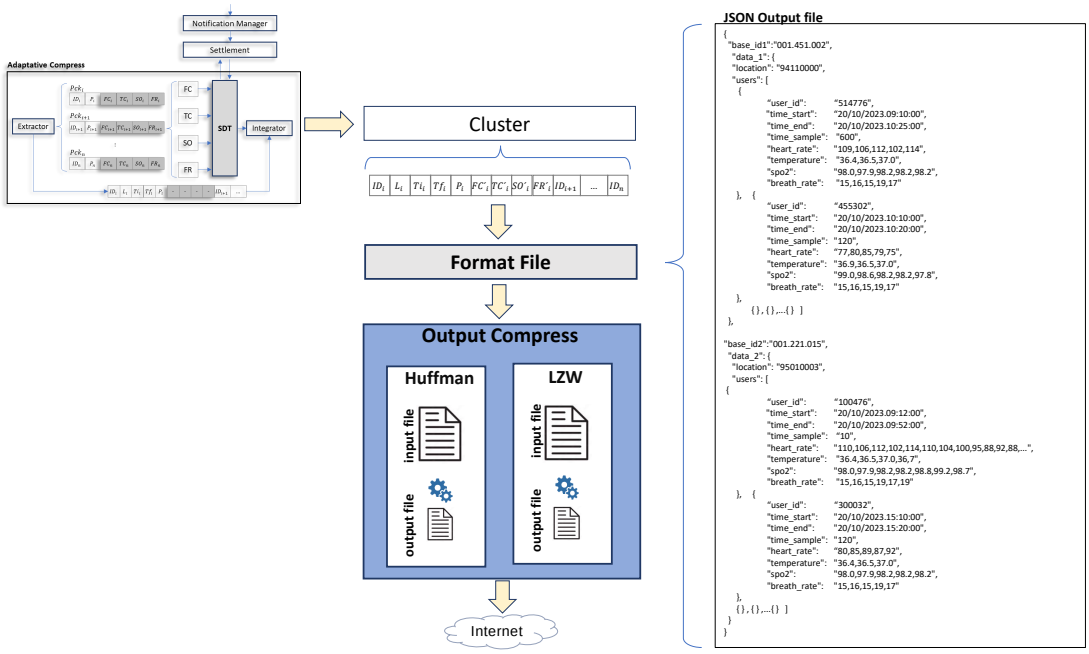


Figura 18: Os pacotes de dados originados de **Assemble** são recebidos no módulo **Format File** para serem formatados para o padrão JSON. Na figura está exemplificado um arquivo de saída contendo coletas de dados de duas estações base, identificadas através de "base_id". O arquivo é então enviado para o módulo **Output Compress**, que consiste na última etapa do *framework*. Nesse módulo os dados serão compactados através dos algoritmos *lossless* LZW ou Huffman Coding, para então serem enviados para a *Cloud* ao sistema de monitoramento de saúde.

Fonte: Elaborado pelo autor

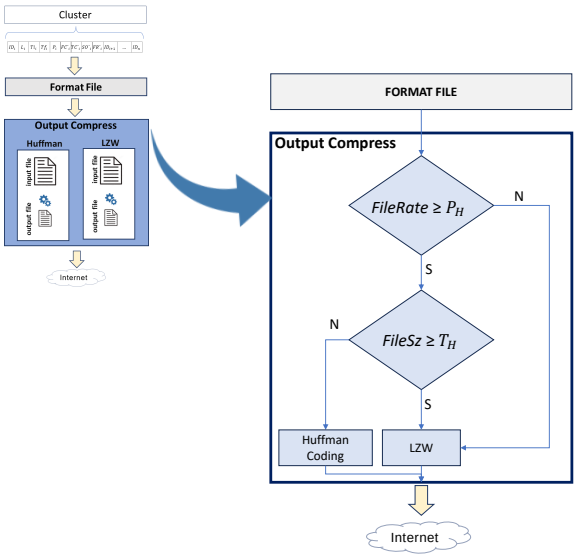


Figura 19: Fluxograma para o processo de decisão na entrada do módulo **Output Compress**. Primeiramente verifica prioridade do arquivo, comparando $FileRate$ com o valor estabelecido para o *framework* no parâmetro P_H. Prioridade baixa é direcionado para o método LZW de compressão. Para prioridade alta, mais uma verificação é feita, o tamanho $FileSz$ é comparado com o tamanho limite estabelecido pelo *framework*, T_H, se estiver no limite inferior será direcionado para a compressão Huffman, porém, se o tamanho exceder o limite definido, será utilizado LZW.

Fonte: Elaborado pelo autor

Tabela 12: Principais parâmetros utilizados no *framework*, classificados quanto ao tipo. Os parâmetros podem ser estáticos ou se modificar em tempo de execução, permitindo o comportamento adaptativo do *framework*.

Tipo	Parâmetro	Descrição
Estático	P_H	Limite para nível de prioridade alta
	T_L	Limite inferior para tamanho do arquivo
	$SzFileCluster_{max}$	Tamanho máximo para o arquivo de saída do módulo **Assemble**
Modifica em tempo de execução	$desvio$	Amplitude da janela do algoritmo SDT
	T_m	Largura da janela do algoritmo SDT

Fonte: Elaborado pelo autor

direcionado para o método de compressão LZW. Se a prioridade for alta, mais uma verificação é feita, agora com o tamanho do arquivo. O tamanho $FileSz$ é comparado ao tamanho limite estabelecido pelo *framework*, dado por T_H. Se estiver no limite inferior, será direcionado para a compressão Huffman; porém, se o tamanho exceder o limite definido, será utilizado o LZW. A saída deste módulo consiste no arquivo pronto para ser enviado para a **Cloud**, até o sistema de monitoramento de saúde, onde será descompactado utilizando o processo de restauração do algoritmo correspondente.

Por fim, após a descrição de cada um dos módulos, cabe apresentar e classificar os principais parâmetros utilizados no processo. Esses parâmetros, atuando conjuntamente, possibilitam que o *framework* tenha um comportamento adaptativo. A Tabela 12 apresenta esses parâmetros, compilados e classificados conforme seu tipo, indicando se são estáticos ou modificados em tempo de execução.

4.4 Algoritmos e operação

A partir da visão detalhada do *framework* apresentada na subseção anterior, é possível acompanhar cada etapa do processo, desde a coleta de sinais vitais dos usuários nas estações base distribuídas geograficamente na Cidade Inteligente até o estágio final de compressão, com os dados prontos para serem transmitidos para o sistema de monitoramento de saúde. A seguir, serão apresentados os principais algoritmos do *framework* VSAC, detalhando as etapas do processo.

O processo de compressão tem início com o recebimento dos dados no módulo **Adaptive Compress**, tendo como entrada o componente *Extractor*. O Algoritmo 1 apresenta este processo no qual os dados brutos são tratados no *Extractor*, onde serão formatados de modo que cada indivíduo tenha seus sinais vitais armazenados no respectivo Pck_i. Para isso, em um *loop* que passa por todos os dados, serão retirados aqueles que estiverem entre os identificadores de início - *STX* e fim - *ETX* de dados para serem armazenados em cada Pck_i. A partir dessa identificação, a monitoração e o tratamento individualizado das informações de cada pessoa podem ser feitos em tempo de execução ao longo do processo de compressão do *framework*.

Algorithm 1 Algoritmo de aquisição de dados brutos recebidos da estação de coleta de dados no módulo **Extractor** e preparação dos pacotes Pck_i para o primeiro estágio de compressão.

```
1: Entrada: Dados brutos data a partir das estações base de coleta.
2: Saída: Pacotes segregados por indivíduo monitorado em Pck_i.
3: Início
4: Definir array de dados recebidos como input_data.
5: Inicializar booleana in_buffer para indicar cópia de dado de array para Pck_i.
6: Inicializar buffer Pck_i para armazenar os dados.
7: Inicializar N com tamanho array.
8: j ← 0
9: i ← 0
10: for N > j do
11:     for cada item em input_data do
12:         if item = STX then
13:             Set in_buffer para True.
14:             Clear Pck_i
15:             i ← i + 1
16:         end if
17:         if item = ETX then
18:             Set in_buffer para False.
19:         end if
20:         if in_buffer is True then
21:             Acrescentar item em Pck_i
22:         end if
23:         j ← j + 1
24:     end for
25: end for
26: Fim
```

Como saída dessa primeira etapa de tratamento dos dados brutos, teremos os dados dos sinais vitais de cada pessoa monitorada armazenados em pacotes Pck_i. A seguir, cada um desses pacotes terá seus sinais separados em arquivos para que possam ser tratados individualmente no módulo de compressão adaptativa **STD**. Este processo está demonstrado no Algoritmo 2, onde um *loop* garante que cada pacote Pck_i passe pelo processo de separação de seus sinais vitais. Dessa forma, os dados de frequência cardíaca, temperatura corporal, saturação de oxigênio

e frequência respiratória estarão disponíveis para serem transmitidos ao primeiro módulo de compressão. Após essa fase de preparação dos dados, deixando cada sinal vital segregado por pessoa monitorada, é possível individualizar a taxa de compressão não apenas por indivíduos, mas também por tipo de sinal vital.

Ainda na etapa adaptativa, a taxa de compressão do algoritmo com perdas **SDT** será dada pelos parâmetros $desvio$ e T_m, partindo de um valor *default*, a ser definido conforme o consumo de largura de banda disponível e o tempo de transmissão dos dados para o Sistema de Monitoramento. Esses parâmetros serão adaptados através do **SetUp** para valores que podem atenuar a taxa de compressão para um dado sinal vital ou mesmo não aplicar compressão nos dados, mantendo-os íntegros conforme recebidos na origem.

O algoritmo **SDT** funciona basicamente "abrindo" ou "fechando" a "porta" com base na tendência dos dados. Esta lógica está refletida no Algoritmo 3 na *Função* $SwingingDoorAlgoritmo$, comparando os valores entre dois pontos consecutivos. Se a diferença entre esses pontos for maior que o desvio especificado, o ponto de dados é considerado significativo e é adicionado ao $resultado$. O índice $esquerda$ é atualizado para o novo ponto significativo, e o processo continua até que todos os dados sejam percorridos.

Seguindo neste componente, a variável $desvio$ - que representa o parâmetro D na Figura 17 - corresponde à amplitude da porta, a qual está diretamente relacionada com a taxa de compressão do algoritmo. Para valores maiores da variável $desvio$, maior será a taxa de compressão; e para valores menores, menor será a taxa de compressão. Esses ajustes no parâmetro serão feitos de acordo com as sinalizações recebidas do módulo **Notification Manager**.

Após os dados terem sido processados em **SDT**, eles são recebidos no componente **Integrator**, que irá restaurar os dados agora comprimidos nos respectivos Pck_i , mantendo o formato original estabelecido no componente **Extractor**. Após esse processo, os pacotes de dados estarão prontos para serem enviados para a próxima etapa do *framework*, no módulo **Assemble**.

O módulo **Assemble**, além de agrupar os pacotes com os dados já processados, desempenha um papel importante no processo de compressão, que é garantir que o tamanho do arquivo final seja adequado para a próxima etapa de compressão sem perdas. Conforme apresentado na Seção anterior, na Tabela 7, para que o algoritmo LZW seja mais eficiente, tanto em taxa de compressão quanto em tempo de processamento, os arquivos devem estar situados em uma determinada faixa de tamanho, e esses limites para a construção do arquivo serão estabelecidos em **Assemble**.

Durante o processo de compressão, a atuação conjunta entre **SetUp** e **SDT** é o habilitador do modo adaptativo do *framework*. Conforme apresentado no fluxograma da Figura 16 e no Algoritmo 4, para cada pacote com os sinais vitais Pck_i recebidos em **SDT**, o respectivo ID da pessoa monitorada é comparado com aqueles que precisam ser tratados de modo diferenciado. Esse tratamento diferenciado consiste em dar prioridade na fila de envio dos dados de um indivíduo particular, assim como em fazer ajustes nos parâmetros do algoritmo de compressão *Swinging Door Trending* , de modo que os dados sejam preservados para um acompanhamento

mais detalhado do estado de saúde do paciente.

Algorithm 2 Algoritmo para retirar de cada Pck_i apenas os dados de cada sinal vital e enviar para o módulo **STD**.

Entrada: Pacotes de dados com sinais vitais das pessoas monitoradas.
Saída: Sinais vitais segregados para serem transmitidos para módulo de compressão.
Início
Inicializar buffer FC_i para armazenar os dados de frequência cardíaca.
Inicializar buffer TC_i para armazenar os dados de temperatura corporal.
Inicializar buffer SO_i para armazenar os dados de SpO2.
Inicializar buffer FR_i para armazenar os dados frequência respiratória.
Inicializar `Num_IDs` com número de pessoas monitoradas.
$i \leftarrow 0$
while $Num_IDs \neq 0$ **do**
    for cada item em Pck_i **do**
        if item = `FC` **then**
            Clear FC_i
            Recortar dados de Pck_i e mover para FR_i
        end if
        if item = `TC` **then**
            Clear TC_i
            Recortar dados de Pck_i e mover para TC_i
        end if
        if item = `SP` **then**
            Clear SP_i
            Recortar dados de Pck_i e mover para SP_i
        end if
        if item = `FR` **then**
            Clear FR_i
            Recortar dados de Pck_i e mover para FR_i
        end if
    end for
    Enviar FC_i, TC_i, SO_i e FR_i para módulo `STD`.
    $i \leftarrow i + 1$
    $Num_IDs \leftarrow i + 1$
end while
Fim

O Algoritmo 5 descreve o funcionamento do módulo *Assemble*, responsável por agrupar os pacotes de dados Pck_i já comprimidos no estágio com perdas, de forma que o tamanho final de cada bloco esteja dentro de um intervalo ideal para a etapa de compressão sem perdas. O processo tem início com a definição dos limites mínimo e máximo de tamanho dos arquivos, conforme estabelecido na Tabela 7, e com a inicialização de um bloco temporário para acumular os pacotes. A cada iteração, o algoritmo verifica se o próximo pacote pode ser adicionado ao bloco corrente sem ultrapassar o tamanho máximo permitido. Caso contrário, o bloco é fechado e armazenado na estrutura *File_Blocks*, iniciando um novo agrupamento. O algoritmo também trata situações em que o bloco corrente não atinge o tamanho mínimo, realizando o preenchimento com pacotes adicionais até atender ao critério estabelecido. Essa lógica garante que os arquivos produzidos ao final do processo estejam otimizados para a compressão sem perdas via Huffman ou LZW, contribuindo para ganhos de eficiência tanto na taxa de compressão quanto no tempo de processamento.

Os arquivos, agora com tamanho reduzido, estão prontos para seguir para a próxima etapa de compressão. Contudo, serão primeiramente formatados para o padrão JSON no módulo **Format File** e, com isso, o arquivo pode ser enviado para a última etapa do processo de compressão, no módulo **Output File**. Nessa última etapa, será então aplicado o algoritmo de compressão sem perdas LZW ou Huffman Coding, conforme estabelecido na Seção 4.2. Neste último módulo,

Algorithm 3 Algoritmo do primeiro módulo do protótipo, emulando **Adaptive Compress** através do método de compressão Swinging Door Trending.

Entrada: Pacotes de dados com sinais vitais das pessoas monitoradas.
Saída: Sinais vitais compactados.
Início
Função SwingingDoorAlgoritmo(dados, desvio):
$n \leftarrow Comprimento(dados)$
$resultado \leftarrow [dados[0]]$
$esquerda \leftarrow 0$
Para cada direita de 1 até n - 1:
if Absoluto(dados[direita][1] - dados[esquerda][1]) > desvio **then**
    $dados[direita] \leftarrow resultado$
    $esquerda \leftarrow direita$
end if
Retornar $resultado$
Fim

a decisão sobre o tipo de compressão será feita de acordo com as informações contidas no *header* do arquivo a ser comprimido, conforme Tabela 10. Os campos referentes ao tamanho do arquivo e ao nível de prioridade servem como base para direcionar para Huffman ou LZW, sendo o tamanho do arquivo o principal fator de decisão.

Na entrada do módulo **Output File** , será utilizado um parâmetro para determinar qual algoritmo de compressão deve ser utilizado. Se o tamanho do arquivo for inferior ou igual ao valor de Tam_{huff}, o método será Huffman; caso contrário, será utilizado LZW. Assim, como saída, teremos o arquivo contendo todos os pacotes Pck_i das pessoas monitoradas comprimidos. Essa compressão é a última etapa do *framework* VSAC, deixando os dados prontos para serem enviados para a *Cloud* através da rede de internet.

Os arquivos, então, recebidos no Sistema de Monitoramento, estão prontos para serem armazenados e, através de um processo de descompressão do algoritmo LZW, os dados restaurados podem ser utilizados para análise, monitoramento e para alimentar *dashboards* com informações das pessoas monitoradas.

Algorithm 4 Algoritmo do módulo **SetUp** que atua sobre o primeiro estágio de compressão no módulo **STD** para priorizações em tempo de execução.

Entrada: `SetUp` recebe parâmetros de `STD` e `Notification Manager`
Saída: Parâmetros de configuração de prioridade para primeiro nível de compressão.
Início
if $ID_m = ID_i$ **then**
    if Prioridade = TRUE **then**
        Identificar Pck_i como alta prioridade em STD
    end if
    Ajustar parâmetros de T_m e `desvio` reduzindo valores para menos taxa de compressão.
end if
Enviar conjunto de parâmetros para módulo `STD`.
Fim

A Figura 20 apresenta a distribuição dos algoritmos ao longo do fluxo de dados do *framework* VSAC, destacando o papel específico de cada um nos diferentes módulos do sistema. Os algoritmos 1, 2 e 3 compõem o módulo **Adaptive Compress**, responsável pelas etapas de extração, separação e compressão com perda dos sinais vitais. O algoritmo 4 está vinculado ao módulo **SetUp**, que realiza ajustes adaptativos dos parâmetros de compressão em tempo de execução com base na criticidade clínica. Já o algoritmo 5 pertence ao módulo **Assemble**,

Algorithm 5 Algoritmo do módulo **Assemble** responsável por agrupar os pacotes Pck_i já comprimidos no primeiro estágio, organizando-os para a próxima etapa de compressão sem perdas.

Entrada: Lista de pacotes Pck_i já processados por `STD`.
Saída: Arquivos organizados com tamanho adequado para compressão sem perdas.
Início
Inicializar lista vazia `File_Blocks`.
Definir `block_size_min` e `block_size_max` com base na Tabela 7.
Inicializar `current_block` como vazio.
Inicializar variável `current_size` como 0.
for cada Pck_i em lista de pacotes **do**
    if `current_size` + tamanho(Pck_i) ≤ `block_size_max` **then**
        Adicionar Pck_i a `current_block`
        Atualizar `current_size`
    else
        if `current_size` ≥ `block_size_min` **then**
            Adicionar `current_block` à lista `File_Blocks`
            Inicializar novo `current_block` e resetar `current_size`
            Reprocessar Pck_i na nova iteração
        else
            Preencher `current_block` com Pck_i até atingir tamanho mínimo
        end if
    end if
end for
if `current_block` não estiver vazio **then**
    Adicionar `current_block` à lista `File_Blocks`
end if
Enviar `File_Blocks` ao módulo `Format File`
Fim

cuja função é agrupar os pacotes comprimidos em blocos compatíveis com a compressão sem perdas. A Tabela 4.4 complementa as informações apresentadas, oferecendo uma visão estruturada da correspondência entre cada algoritmo, o módulo em que estão inseridos, sua função principal e o tipo de operação realizada, para fins de compreensão sequencial e modularidade do *framework*.

Tabela 13: Correspondência entre algoritmos conforme módulo em que está inserido, sua função principal e o tipo de operação.
Fonte: Elaborado pelo autor

Módulo	Algoritmo	Função	Tipo de operação
Adaptive Compress	1	Separar pacotes Pck_i por indivíduo	Pré-processamento
Adaptive Compress	2	Segregar sinais vitais por tipo (FC, TC, FR, SpO2)	Extração
Adaptive Compress	3	Aplicar compressão com perdas (Swinging Door Trending)	Compressão com perdas
SetUp	4	Ajustar parâmetro desvio	Adaptação
Assemble	5	Agrupar pacotes para compressão sem perdas, respeitando limites	Pré-processamento

Em síntese, os algoritmos apresentados nesta seção constituem uma arquitetura sólida e modular; contudo, seu desempenho pode ser impactado por fatores críticos, como a configuração dos parâmetros adaptativos, a escolha do algoritmo de compressão sem perdas, o custo da serialização intermediária e a integração entre os módulos. A identificação desses pontos reforça a importância de avaliar cuidadosamente o comportamento do *framework* em diferentes condições operacionais. Assim, os aspectos discutidos aqui fundamentam a necessidade de uma metodologia de avaliação abrangente, apresentada no próximo capítulo, capaz de validar a eficiência, a escalabilidade e a adaptabilidade do VSAC diante dos desafios propostos.

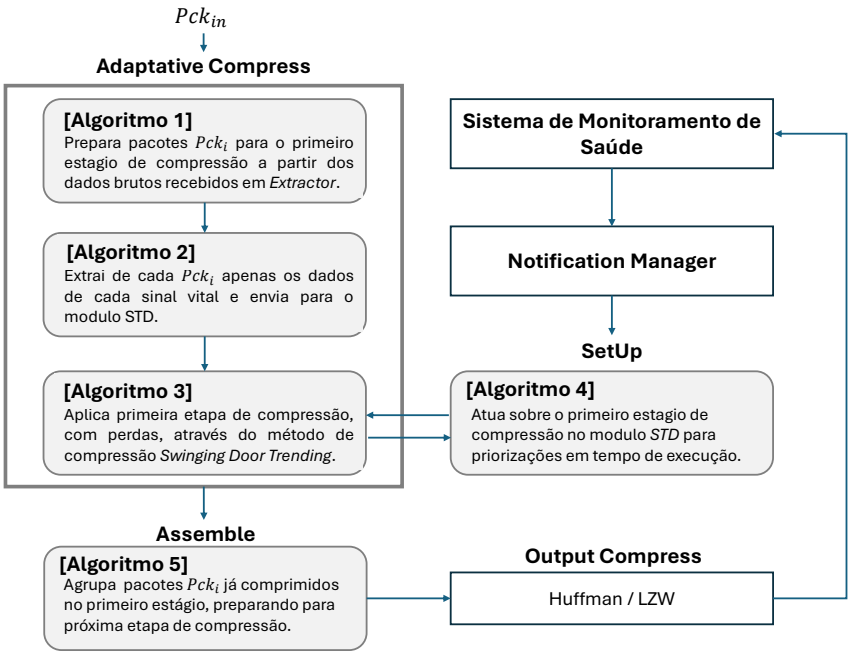


Figura 20: Representação dos algoritmos ao longo do fluxo de dados, destacando sua distribuição nos módulos *Adaptive Compress*, *SetUp* e *Assemble*.

Fonte: Elaborado pelo autor

5 METODOLOGIA E AVALIAÇÃO

Este capítulo apresenta a metodologia desenvolvida para avaliar o desempenho do *framework* VSAC, bem como os procedimentos experimentais aplicados para validar sua eficiência em contextos realistas de monitoramento de sinais vitais em sistemas de saúde conectados. A definição desta metodologia foi fundamentada na literatura científica associada e, principalmente, nas análises realizadas no Capítulo 3 - Trabalhos Relacionados - em que foram considerados trabalhos científicos relevantes, os quais serviram como base para estruturar os critérios de avaliação e os cenários operacionais utilizados neste estudo.

Embora a literatura ofereça diversas contribuições sobre compressão de dados biomédicos e transmissão em redes IoT, identificou-se a ausência de uma abordagem consolidada que una, de forma adaptativa, técnicas híbridas de compressão com e sem perdas, especialmente considerando a criticidade dos dados clínicos e as restrições das arquiteturas do tipo Edge–Fog–Cloud. Diante dessa lacuna, foi necessário desenvolver uma metodologia própria, combinando boas práticas extraídas de diferentes trabalhos. Em particular, os estudos TR3 e TR6 ofereceram fundamentos para a compressão orientada ao contexto clínico; TR4, TR10 e TR12 forneceram diretrizes para a construção de arquiteturas híbridas de compressão; e os trabalhos TR8 e TR9 contribuíram para a definição de requisitos, escalabilidade e integridade dos dados em infraestruturas distribuídas. Esses fundamentos permitiram definir uma metodologia que considera, de forma integrada, os requisitos técnicos e clínicos envolvidos na compressão de sinais vitais em cidades inteligentes.

Assim, a metodologia adotada neste trabalho contempla não apenas os aspectos técnicos da compressão, mas também a adaptabilidade do *framework*. Foram definidas métricas de avaliação amplamente reconhecidas na literatura — como a taxa de compressão (TC), o tempo de compressão (T) e o nível de distorção (PRD) — que permitem não apenas comparar o VSAC a soluções existentes, mas também verificar seu comportamento diante de diferentes condições de operação e prioridades clínicas.

Do mesmo modo que a metodologia adotada teve como base os trabalhos relacionados, a escolha das métricas de avaliação também foi fundamentada nessa análise. Esta literatura orientou a construção de uma abordagem associada aos desafios da transmissão de dados para aplicações na área da saúde. As métricas selecionadas — TC, T e PRD — são amplamente reconhecidas na literatura e foram escolhidas por sua capacidade de avaliar não apenas a eficiência algorítmica do *framework*, mas também seu desempenho sob diferentes condições de operação e graus de criticidade clínica. A Seção 5.3 deste capítulo aprofunda a descrição de cada uma dessas métricas, justificando sua escolha e o papel delas na análise comparativa do VSAC em relação às abordagens existentes.

Este capítulo está estruturado para apresentar, inicialmente, as características do protótipo funcional desenvolvido em Python (Seção 5.1), seguido pela definição das cargas de teste utilizadas nos experimentos (Seção 5.2). Na Seção 5.3, detalham-se as métricas empregadas para

mensurar o desempenho do *framework*. Em seguida, na Seção 5.4, são descritos os três cenários de compressão avaliados — sem perdas, com perdas e híbrido — os quais permitem analisar a eficiência do VSAC em diferentes níveis de compactação, mantendo a fidelidade dos sinais conforme as necessidades do sistema de saúde.

5.1 Protótipo

Para a validação do *framework* VSAC, será desenvolvido um protótipo funcional, implementado na linguagem Python. A escolha dessa linguagem deve-se ao fato de que existem muitos algoritmos, classes e bibliotecas desenvolvidos nessa linguagem que estão disponíveis como *open source*. A linguagem Python também é amplamente utilizada, gratuita e de código aberto, podendo ser usada em diversos sistemas operacionais. O protótipo contém os principais componentes que influenciam o resultado do *framework*, ou seja, os dois estágios de compressão: o primeiro é com perdas e o segundo, sem perdas. Os componentes intermediários, responsáveis por formatar os arquivos e agrupar os pacotes, serão abstraídos através de arquivos prontos para simulação.

Para melhor entendimento do protótipo, a Figura 21 apresenta os módulos do *framework* que serão emulados. Ao todo, três módulos integrados irão emular o comportamento do *framework*, tendo como ponto de entrada os dados referentes aos sinais vitais obtidos por meio de *datasets* de séries temporais com leituras em diferentes tipos de cadências.

Os códigos em linguagem Python para os algoritmos de compressão foram obtidos a partir da plataforma Github, sendo os seguintes *links* para os respectivos códigos:

- Swinging Door Trending: https://github.com/chelaxe/SwingingDoor/blob/main/swinging_door.py
- LZW: https://github.com/adityagupta3006/LZW-Compressor-in-Python
- Huffman: https://github.com/bhrigu123/huffman-coding

O protótipo foi executado em um PC com as seguintes características:

- Processador: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz
- Sistema Operacional: Windows 11 Pro (64 bits)
- Memória RAM: 8 GB
- Memória SSD: 256 GB

Os dados obtidos dos *datasets* consistem em séries temporais com medidas reais de sinais vitais de pessoas monitoradas por meio de sensores biométricos. Os ensaios realizados no protótipo considerarão apenas a frequência cardíaca, na medida em que apresenta, ao longo da série temporal, um comportamento dos dados que varia em relação a uma medida central, característica que é comum também aos demais sinais vitais que serão objeto de estudo deste trabalho.

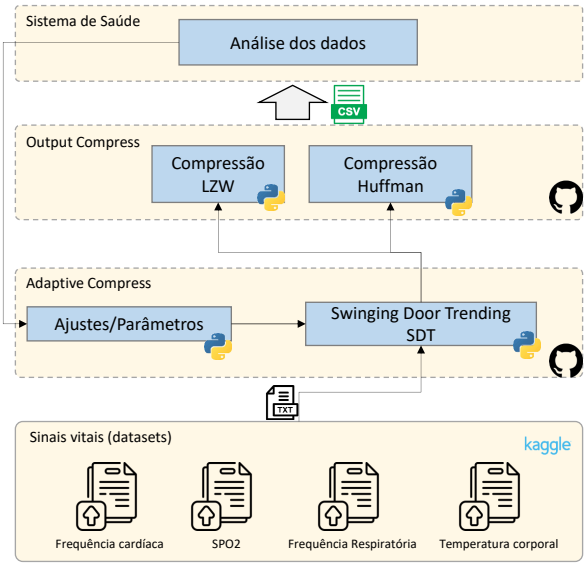


Figura 21: Infraestrutura do protótipo para validação do *framework* com duas etapas formadas por um módulo adaptativo com algoritmo de compressão com perdas e o segundo módulo composto por dois algoritmos de compressão sem perdas.

Fonte: Elaborado pelo autor

Para processar os dados de entrada, foi desenvolvido um módulo para emular **Adaptive Compress**. A parte central deste módulo é dada pelo algoritmo Swinging Door Trending (SDT), conforme descrito na apresentação do algoritmo na Seção 4.4. Neste componente, a variável $desvio$ será ajustada manualmente conforme as leituras de saída e a simulação de diferentes cenários de sinalizações recebidas do módulo **Notification Manager**.

O módulo de compressão **Output Compress** é formado por dois algoritmos de compressão sem perdas: Huffman Coding e LZW. Conforme apresentado na Subseção 4.3.2, o último estágio de compressão aplicará o método de compressão mais adequado ao arquivo de sinais vitais, obtendo a melhor eficiência de acordo com seu tamanho. Aqueles arquivos com tamanho menor, marcados como prioritários, serão direcionados para o método Huffman Coding. Quanto aos arquivos maiores, que contêm vários pacotes agrupados no módulo **Assemble**, eles serão encaminhados para o método LZW. Para efetuar estes testes, o direcionamento dos arquivos para o algoritmo de compressão mais adequado será feito manualmente, sem o processo automatizado de leitura do *header* do arquivo para a tomada de decisão.

5.2 Cargas de teste

A carga utilizada nos testes consiste em *datasets* que permitem simular diferentes cenários com distintos tamanhos de pacotes. Os *datasets* foram obtidos através da base de dados Kaggle [1] e, para a validação deste protótipo, serão realizados ensaios com dados de pessoas monitoradas para frequência cardíaca.

[1] https://www.kaggle.com/datasets

Esse conjunto de dados foi extraído de dispositivos vestíveis, de modelos disponíveis comercialmente, para medir a atividade física em nível populacional. Para isso, uma amostra de 46 participantes (26 mulheres e 20 homens) utilizou três dispositivos: GENEActiv, Apple Watch Series 2 e Fitbit Charge HR2. Os testes consideram o sinal vital da frequência cardíaca, pois ele apresenta um comportamento dos dados que varia em relação a uma medida central ao longo da série temporal. Cada conjunto de dados utilizado corresponde a uma sequência de leituras periódicas de diferentes pessoas. Foram utilizados três conjuntos de dados diferentes nos ensaios, com intervalos de medição distintos ao longo de 24 horas, os quais correspondem a intervalos de leitura de 15 segundos, 1 minuto e 15 minutos. Para testar o *framework* para diferentes densidades de dados, três novos conjuntos de dados foram criados, combinando os três anteriores, conforme mostrado na Tabela 15 e também representados por meio da Figura 22. Essa estratégia permite o monitoramento de indivíduos por um dia, abrangendo o período mínimo recomendado de 18 horas, o que é essencial para capturar arritmias episódicas e variações cardíacas que não seriam detectadas em janelas temporais mais curtas [Cygankiewicz and Zareba, 2013].

O menor conjunto de dados, com tamanho de 2,6 kB e cadência de 15 min, é utilizado para avaliar o desempenho do *framework* em arquivos pequenos e de alta prioridade. Esse conjunto de dados ajuda a avaliar a capacidade do *framework* de lidar com volumes mínimos de dados de forma eficiente. À medida que o tamanho do conjunto de dados aumenta, o desempenho do *framework* é testado com densidades de dados mais altas e números variados de dispositivos. O maior conjunto de dados, com tamanho de 100 MB e alta densidade, desafia sua capacidade de lidar com fluxos de dados heterogêneos e em larga escala. Para atingir os tamanhos e as características de arquivo desejados, os conjuntos de dados foram compostos por diferentes números de dispositivos com variadas cadências de dados. Por exemplo, o conjunto de dados de 1 MB foi gerado combinando dados de 34 dispositivos com cadência de 15 segundos, 34 dispositivos com cadência de 1 minuto e 40 dispositivos com cadência de 15 minutos, totalizando 108 dispositivos. Essa gama diversificada de conjuntos de dados permite uma avaliação abrangente da escalabilidade, eficiência e precisão de VSAC sob configurações operacionais realistas.

5.3 Métricas

As métricas de avaliação utilizadas neste trabalho foram selecionadas com base nos trabalhos relacionados, conforme a Tabela 14, com foco em sua relevância e adequação para mensurar e garantir consistência na avaliação do desempenho do *framework*. As métricas adotadas, Taxa de Compressão (TC), Tempo de Compressão (T) e Percentual de Distorção Relativa (PRD), foram escolhidas com base em uma análise dos Trabalhos Relacionados, que aplicam diferentes abordagens de compressão de sinais vitais no contexto de IoT para a saúde. Nesse sentido, observou-se que a TC é amplamente empregada nos estudos como uma métrica fundamental para avaliar a eficiência da redução do volume de dados.

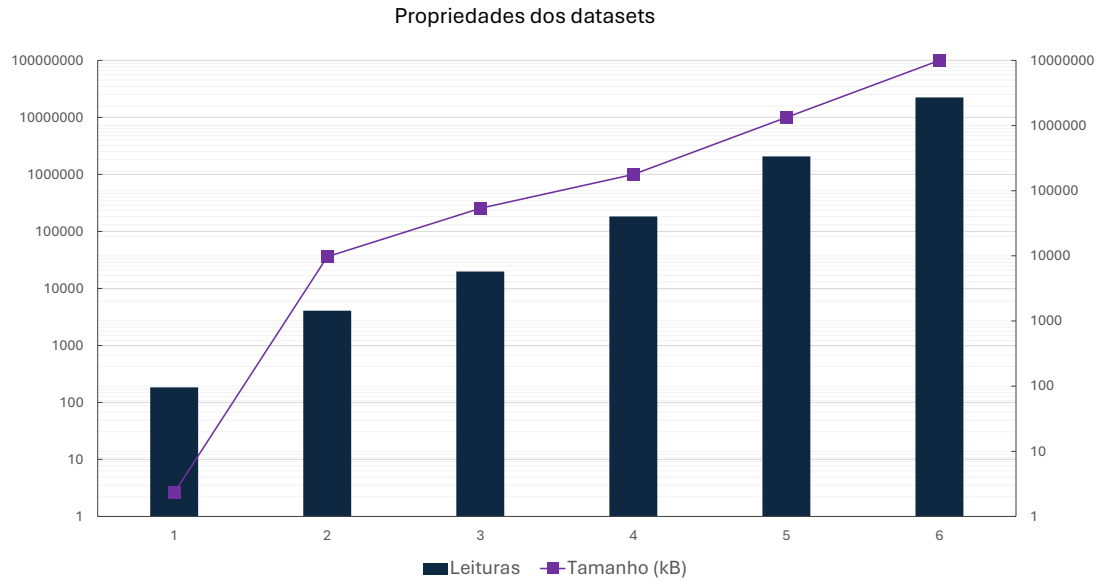


Figura 22: Gráfico em escala logarítmica que apresenta as propriedades dos seis *datasests* utilizados para validação do protótipo de acordo com a quantidade de leituras (aquisição de dados) e respectivo tamanho.

Fonte: Elaborado pelo autor

Dessa forma, com base nessas verificações, primeiramente será medida a TC para avaliar a eficiência com que o VSAC reduz o tamanho dos pacotes de dados em seus dois estágios de compressão (Equação 5.1). Na equação de TC, Tf é o tamanho final do arquivo com os dados compactados e Ti é o tamanho inicial do arquivo com os dados originais. Essa métrica é essencial para entender quão bem o *framework* conserva a largura de banda, retendo informações importantes para o monitoramento do estado de saúde. Além disso, também será medido o Tempo de Compressão (T), indicando a velocidade de processamento para os estágios de compressão, com e sem perdas. Essa métrica fornece *insights* sobre a capacidade do sistema de lidar com altos volumes de dados de forma eficiente, um recurso necessário para aplicações de monitoramento em tempo real. Esta métrica é adotada em TR2, TR3, TR5, TR6, TR7, TR9, TR12 e TR15, principalmente em cenários de *Edge/Fog Computing*, nos quais a latência é um fator crítico para aplicações em tempo real.

$$TC = \frac{Tf}{Ti} \times 100 \tag{5.1}$$

Outra métrica significativa adotada na literatura relacionada é a medida da Diferença Quadrática Média Percentual, ou Percentage Root-Mean-Square Difference (PRD), a qual se aplica para avaliar o nível de distorção introduzido pela etapa de compressão com perdas. Esta métrica foi adotada por trabalhos que lidam com compressão com perdas e preocupação com a qualidade dos sinais restaurados, como TR2, TR3, TR4 e TR8 a TR16, especialmente quando

os dados envolvem sinais sensíveis, como ECG ou EEG. A PRD mede a precisão dos dados compactados em relação ao original, tornando-se uma métrica necessária para garantir a confiabilidade das informações sobre o estado de saúde [Hassan and Mohsen, 2024, Rajankar et al., 2024]. Esta métrica é comumente utilizada no processamento de sinais biomédicos; a PRD reflete o equilíbrio entre a redução do tamanho dos dados e a fidelidade dos dados [Moein et al., 2024]. É importante avaliar a capacidade de VSAC de preservar a qualidade dos dados dos sinais vitais, otimizando o armazenamento e a transmissão.

Para medir o efeito da distorção dos dados nos dados originais após a aplicação da compressão com perdas, a fórmula PRD envolve a comparação dos dados originais (entrada) com os dados reconstruídos (saída) para quantificar as diferenças introduzidas pela compressão com perdas (Equação 5.2). Aqui, N é o número total de pontos de dados em cada arquivo, X_i representa cada ponto de dados no arquivo de dados de frequência cardíaca original e Y_i representa cada ponto de dados correspondente no arquivo de dados de frequência cardíaca compactado.

Para cada cenário de teste avaliado no protótipo, foram realizadas 10 repetições por medida, incluindo a taxa de compressão e o tempo de compressão. A escolha desse número foi fundamentada na estabilidade observada nos resultados experimentais, que apresentaram baixa variação entre execuções sucessivas. Essa consistência permitiu reduzir o número de repetições em relação à recomendação da literatura tradicional [Montgomery, 2017, Devore, 2011, Jain, 1990] de 30 execuções, mantendo, ainda assim, a confiabilidade nos valores médios obtidos e a viabilidade prática na condução dos experimentos.

Com isso, esse conjunto de métricas fornece uma visão abrangente do desempenho do protótipo, permitindo avaliar simultaneamente aspectos importantes para sua aplicação em ambientes de monitoramento de saúde. Ou seja, será possível quantificar a eficiência na redução do volume de dados através da TC, essencial para economizar largura de banda e otimizar o armazenamento; por meio de T, irá se analisar a viabilidade de uso do *framework* em sistemas que exigem respostas rápidas, como aqueles baseados em arquiteturas Edge-Fog-Cloud; e o PRD avalia o quanto a integridade do sinal é preservada após a compressão, o que é relevante para a confiabilidade clínica das informações transmitidas. Em conjunto, essas métricas possibilitam uma análise equilibrada entre desempenho computacional, fidelidade dos dados e sua capacidade de adaptação, especialmente em cenários com restrições operacionais típicas de aplicações em tempo real e em cidades inteligentes.

$$PRD = \frac{\sqrt{\sum_{i=1}^{N}(X_i - Y_i)^2}}{\sqrt{\sum_{i=1}^{N} X_i^2}} \times 100 \tag{5.2}$$

5.4 Cenários de teste

Apesar da ampla presença das métricas de avaliação na literatura, não se verifica, da mesma forma, uma metodologia integrada e adaptativa, como a proposta nesta pesquisa. Os trabalhos

Tabela 14: Principais métricas adotadas nos trabalhos relacionados.

Métrica	Trabalhos Relacionados																Total
	TR1	TR2	TR3	TR4	TR5	TR6	TR7	TR8	TR9	TR10	TR11	TR12	TR13	TR14	TR15	TR16	
Taxa de compressão	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	**16**
Taxa de distorção		✓	✓	✓				✓	✓	✓	✓	✓	✓	✓	✓	✓	**12**
Tempo de compressão		✓	✓		✓	✓	✓		✓			✓			✓		**8**
Consumo de energia	✓				✓							✓	✓		✓		**5**
Tempo de descompressão		✓			✓				✓			✓					4

analisados não contemplam, de forma simultânea, os seguintes elementos: (i) compressão híbrida com dois estágios, (ii) parametrização adaptativa sensível ao estado clínico do paciente e (iii) avaliação empírica com dados reais e múltiplos tamanhos de arquivos, permitindo observar o comportamento do *framework* em condições variadas que simulem desafios técnicos presentes em aplicações urbanas baseadas em IoT. Assim, foi necessário desenvolver uma metodologia própria de avaliação, construída de forma modular, com base em aspectos parciais de alguns trabalhos da literatura.

O trabalho utilizou TR3 e TR6 como referências para compressão adaptativa orientada ao contexto clínico, em TR4, TR10 e TR12, pela estrutura híbrida de compressão, e em TR8 e TR9, pelas preocupações com latência, escalabilidade e integridade da informação em arquiteturas *Edge-Fog-Cloud*. Além disso, a ênfase na distorção clínica controlada e no balanceamento entre a qualidade do sinal e a eficiência de transmissão diferencia esta pesquisa dos estudos anteriores. A partir dessa combinação de fundamentos, este trabalho propõe um modelo de avaliação diretamente alinhado às exigências operacionais dos sistemas de saúde conectados em cidades inteligentes.

Os ensaios elaborados consistem em validar o comportamento e a eficiência de VSAC através de dados reais de um paciente monitorado ao longo de um período de 24 horas. Os principais fatores que permitem analisar o desempenho estão diretamente relacionados à taxa de compressão final e ao tempo despendido para executar a compressão dos dados. Para isso, estes testes precisam simular cenários reais com base em dados obtidos de pessoas monitoradas em diferentes características, conforme descrito na Seção 5.2. Considerando que, para estes testes, serão utilizados dados de frequência cardíaca, os *datasets* aplicados contemplam pessoas com variações de frequência normais e situações de arritmia cardíaca.

Para esses diferentes tipos de pacotes de sinais vitais, são analisados três cenários de teste. Estes cenários foram elaborados de modo que seja possível analisar o *framework* proposto sob a perspectiva de diferentes tipos de dados de entrada e diferentes combinações de métodos de compressão. Desse modo, teremos os seguintes ensaios para análise:

- **Cenário 1: Sem perdas** - neste cenário, os dados serão processados apenas com algoritmos de compressão sem perdas. Neste teste, será analisado o resultado com apenas um componente de compressão, com base nos dois algoritmos projetados para o *framework*;
- **Cenário 2: Com perdas** - nesse cenário, os dados serão processados apenas com o método de compressão com perdas O teste consiste em analisar o resultado ao aplicar apenas o algoritmo com perdas planejado no *framework*;

Tabela 15: Tipos de arquivos utilizados para o teste do *framework*, com medidas de cada indivíduo durante um período de 24 horas.

Tamanho (kB)	Cadência	# Leituras	# Dispositivos	Propriedade	Propósito
2,6	15 min	96	1	Baixa densidade de dados e arquivo de tamanho muito pequeno.	Validar desempenho com arquivos pequenos e com prioridade.
36	1 min	1.440	1	Alta densidade de dados e arquivos de tamanho pequeno.	Analisar desempenho com dados homogêneos em poucos pacotes.
255	15 seg	5.760	1	Muito alta densidade de dados e arquivo de tamanho médio.	Analisar o *framework* com poucos pacotes e alta densidade.
1.000	1 min	40.320	28	Alta densidade de dados e arquivo de tamanho médio.	Analisar *framework* com pacotes de tamanho padrão.
10.000	15 seg / 1 min / 15 min	335.222	108	Alta densidade de dados e arquivo de tamanho grande.	Analisar *framework* com dados heterogêneos e grande número de pacotes.
100.000	15 seg / 1 min / 15 min	2.707.584	661	Alta densidade de dados e arquivo de tamanho muito grande.	Analisar o *framework* com dados heterogêneos de número de pacotes muito grande.

- **Cenário 3: Híbrido** - neste cenário, os dados serão processados com base no que foi projetado para o *framework*, com duas etapas de compressão. Sendo a primeira etapa, aplicou-se o método com perdas e, na segunda etapa, o método de compressão sem perdas. Na primeira etapa, também serão realizados ensaios com diferentes configurações do algoritmo SDT.

Com este quadro definido, o desempenho do *framework* será avaliado com base em sua capacidade de atingir uma alta taxa de compressão e um baixo tempo de compressão. Também é importante analisar a relação entre a taxa de compressão alcançada na etapa de compressão com perdas e seu impacto na taxa de distorção, medida pela Diferença Quadrática Média Percentual (PRD). O nível de PRD considerado aceitável será definido remotamente pelo sistema de monitoramento de saúde, com base em critérios médicos adaptados às necessidades de cada paciente. Essa adaptabilidade permite uma abordagem personalizada, na qual os parâmetros de compressão são ajustados para manter a qualidade dos dados que atendem aos padrões clínicos, garantindo a preservação de informações essenciais para um monitoramento preciso da saúde.

Aprofundar a análise dessa relação ajuda a determinar as configurações ideais de compressão para cada paciente, equilibrando a redução de dados com a preservação da integridade dos dados críticos de saúde. Ao monitorar os níveis de PRD, podemos quantificar a extensão da retenção de informações na saída compactada, permitindo que o VSAC alcance uma gestão de dados eficiente sem comprometer a relevância clínica dos sinais vitais. Essa adaptabilidade dos limites de PRD garante que o *framework* VSAC possa responder a diferentes condições do paciente, proporcionando compressão eficaz e, ao mesmo tempo, fornecendo mais evidências de saúde específicas do paciente em tempo real.

6 RESULTADOS E ANÁLISE

Este capítulo apresenta os resultados obtidos a partir da execução dos testes experimentais descritos na metodologia, com foco na avaliação do desempenho do VSAC em relação a diferentes estratégias de compressão de dados. Os experimentos foram conduzidos por meio do protótipo funcional implementado em Python, utilizando conjuntos de dados com diferentes cadências de amostragem e variações controladas no parâmetro de desvio da compressão com perdas.

A estrutura do capítulo está organizada, iniciando com a Seção 6.1, que descreve os ensaios realizados para cada um dos três cenários de teste — compressão sem perdas, compressão com perdas e compressão híbrida com estratégia adaptativa. Os testes contemplam arquivos de entrada de diferentes tamanhos e densidades, permitindo uma avaliação abrangente da escalabilidade e eficiência do *framework*.

Na Seção 6.2, os dados coletados são analisados com base nas métricas de Taxa de Compressão (TC), Tempo de Compressão (T) e Diferença Percentual Quadrática Média (PRD). Essa análise busca identificar padrões de comportamento entre os cenários testados, destacando os ganhos de desempenho proporcionados pela estratégia híbrida do VSAC. Além disso, discute-se a relação entre o nível de compressão alcançado e o impacto na qualidade dos dados, de modo a verificar o equilíbrio entre eficiência e preservação da informação clínica. Por fim, na Seção 6.3, serão discutidos os resultados obtidos em relação aos trabalhos relacionados e às possibilidades de utilização em cenários de monitoramento de sinais vitais.

6.1 Ensaios

Nesta seção, são apresentados os resultados obtidos a partir dos cenários planejados para avaliar o *framework* proposto com base no protótipo. Os testes cobrem os cenários que permitem comparar a eficiência de um processo que combina dois métodos de compressão com abordagens distintas, aplicados em dois estágios de compressão, utilizando os métodos separadamente. Com base em diferentes tamanhos de arquivo de entrada, os valores da taxa de compressão e do tempo de processamento foram coletados. No caso do módulo adaptativo, foram testados diferentes valores do parâmetro $desvio$, aumentando seu valor em cada ciclo de teste. Esses valores foram sendo incrementados a cada ensaio, partindo de um desvio de 2 e chegando a 5. Os valores foram escolhidos considerando a média dos sinais monitorados, sem incorrer em perda de informação importante para o diagnóstico médico.

6.1.1 Cenário 1: Sem perdas

Para o primeiro cenário de testes, foram aplicados os *datasets* com os dados de sinais vitais na entrada dos módulos de compressão sem perdas, conforme apresentado na Figura 23, que

descreve a forma como o teste foi executado. Nesses testes, foram aplicados os mesmos *datasets* nos dois algoritmos de compressão sem perdas. Para cada arquivo, foram medidos os tempos desde a entrada dos dados até o final do processo de compressão.

Na Tabela 16 estão os resultados dos testes do cenário 1, com os algoritmos de compressão Huffman Coding e LZW utilizados no processo de redução dos arquivos com pacotes de sinais vitais. Mesmo sendo conhecido que estes dois algoritmos apresentam desempenhos diferentes conforme o tamanho do arquivo de entrada, ambos foram submetidos ao mesmo tipo de ensaio. Cabe destacar que no *framework* final proposto neste trabalho, no módulo de **Output compress** o arquivo será direcionado para o método de compressão adequado, de acordo com os critérios de tamanho e priorização.

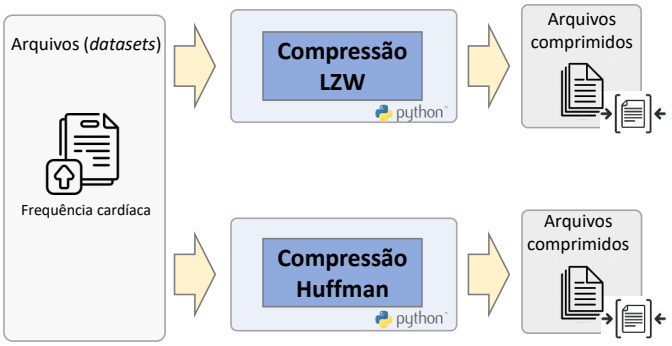


Figura 23: Estrutura para o cenário de testes contemplando apenas algoritmos sem perdas, Huffman Coding e LZW.

Fonte: Elaborado pelo autor

Tabela 16: Resultado do cenário 1 de testes, aplicando apenas algoritmos de compressão sem perdas, tendo como entrada os arquivos de diferentes tamanhos. Os resultados deste teste colocam lado a lado os resultados de LZW e Huffman Coding.

	Huffman Coding			**LZW**		
Entrada (kB)	Saída (kB)	TC	T (seg)	Saída (kB)	TC	T (seg)
2,38	1,39	58,40	0,005	1,58	66,38	0,0149
36	17,5	48,61	0,052	12,5	34,72	0,0214
255	128	50,19	0,428	83,4	32,70	0,082
1.113	550	49,41	1,709	293	26,32	0,32
10.000	4.830	48,30	15,5	2560	25,60	3,26
100.000	49.552	49,55	154,64	20.100	20,10	43,3

6.1.2 Cenário 2: Com perdas

Nestes testes, estão sendo aplicados os seis diferentes tipos de arquivos de entrada e, para cada um, será aplicado um ciclo com desvio diferente. Ao todo, são quatro ensaios para cada arquivo, sendo medidos os respectivos tempos e o tamanho do arquivo, resultando na taxa de compressão associada.

A Figura 24 apresenta a estrutura dos módulos do protótipo para a execução do teste. Cada um dos seis *datasets* é lido pelo módulo SDT, e o parâmetro $desvio$ é incrementado em cada

ciclo, totalizando 24 ciclos de teste para este cenário. O tempo total transcorrido desde a entrada do arquivo até a saída do arquivo reduzido será computado como o tempo total.

Os resultados das medidas desses 24 ciclos, juntamente com os cálculos de TC e o tempo total despendido em cada etapa, estão apresentados na Tabela 17.

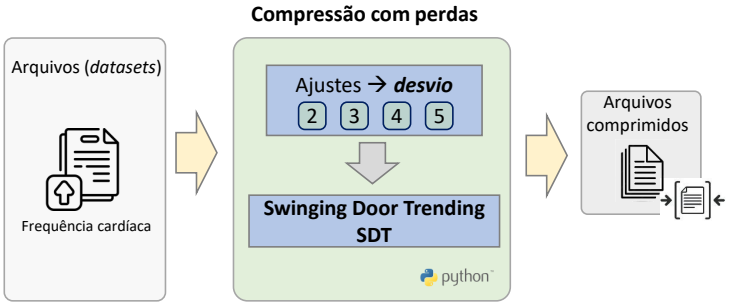


Figura 24: Estrutura para o cenário de testes de um processo de compressão apenas algoritmos com perdas, no qual cada arquivo de entrada será testado com os desvios de 2, 3, 4 e 5.

Fonte: Elaborado pelo autor

Tabela 17: Resultado do cenário 2 de testes, aplicando apenas algoritmo de compressão com perdas (Swinging Door Trending), foram testadas diferentes configurações de taxa de compressão, através do parâmetro *desvio*.

Entrada (kB)	Desvio = 2			Desvio = 3			Desvio = 4			Desvio =5		
	Saída (kB)	TC	T (seg)	Saída (kB)	TC	T (seg)	Saída (kB)	TC	T (seg)	Saída (kB)	TC	T (seg)
2,38	1,21	50,84	0,002	1,18	50,84	0,003	0,886	37,23	0,001	0,774	32,52	0,001
36	12,63	35,08	0,015	9,32	25,89	0,005	7,59	21,08	0,007	6,06	16,83	0,007
255	44,34	17,39	0,023	30,69	12,04	0,021	21,92	8,60	0,023	16,26	6,38	0,022
1.113	404,93	36,38	0,336	303	27,22	0,208	237	21,29	0,299	188	16,89	0,278
10.000	3.560	35,60	1,806	2.670	26,70	2,280	2.090	20,90	1,862	1.660	16,60	1,823
100.000	36.445	36,45	18,840	26.650	26,65	16,320	20.910	20,91	15,988	16.560	16,56	14,252

Fonte: Elaborado pelo autor

6.1.3 Cenário 3: Híbrido

O terceiro cenário de testes é aquele que emula o *framework* que é o objeto deste trabalho. Com dois estágios de compressão, conforme apresentado na Figura 25, considera-se que os dois cenários anteriores estão integrados. Nesse cenário, cada um dos seis *datasets* de entrada será processado no módulo de compressão com perdas, sendo cada um submetido a quatro ciclos de testes para gerar uma análise com os resultados de cada valor do parâmetro *desvio*. A seguir, os arquivos resultantes deste primeiro estágio serão a entrada dos módulos de compressão sem perdas e, conforme projetado, os arquivos de tamanhos 2,38 kB, 26 kB e 255 kB serão direcionados para o algoritmo Huffman Coding, e os demais para o algoritmo LZW.

Na Tabela 18 estão descritos os resultados das medidas de cada etapa, que compreende os dois estágios de compressão, com e sem perdas, bem como para cada valor do parâmetro

$desvio$. Os valores finais de TC e $Ttotal$ consideram o caminho percorrido pelo arquivo desde a origem até a saída do módulo de compressão sem perdas.

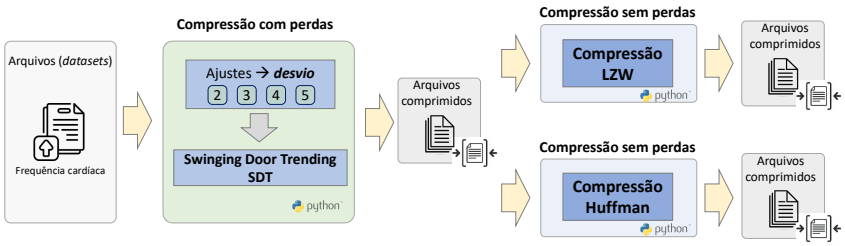


Figura 25: Estrutura para o cenário de testes contemplando dois métodos de compressão em sequencia, sendo que no segundo estágio estão os algoritmos sem perdas, Huffman Coding e LZW.

6.2 Análise dos Resultados

Esta seção apresenta os resultados obtidos a partir dos ensaios realizados com o VSAC nos cenários de avaliação descritos anteriormente. Dado o contexto de compressão em duas etapas, integrando algoritmos com perdas (lossy) e sem perdas (lossless), os seis conjuntos de dados com diferentes parâmetros de desvio são analisados para explicitar a relação entre as diferentes configurações e como isso impacta a taxa de compressão (TC), o tempo de compressão (T) e a integridade dos dados, conforme indicado pelo PRD. Os valores de TC e T foram coletados com base em diferentes tamanhos de arquivos de entrada para avaliar o desempenho do *framework*. Para o módulo adaptativo, o parâmetro $desvio$ foi incrementado gradualmente de 2 a 5 em cada ciclo de teste, escolhido de forma a equilibrar a redução de dados enquanto preserva o nível de informação necessário para o diagnóstico médico, com base nas características médias dos sinais monitorados.

6.2.1 Avaliação da Taxa de Compressão e Tempo de Compressão

A análise dos resultados tem início com o Cenário 1, no qual os dados foram submetidos exclusivamente a algoritmos de compressão sem perdas. Essa abordagem permite uma comparação direta entre os métodos Huffman Coding e LZW. Conforme apresentado na Tabela 16, observa-se que o desempenho de ambos os algoritmos é fortemente influenciado pelo tamanho dos arquivos de entrada, impactando significativamente tanto a TC quanto o T.

Quanto à TC, observa-se que o algoritmo Huffman Coding apresenta um melhor desempenho de compressão para arquivos pequenos. Para o menor arquivo testado (2,38 kB), Huffman atinge uma TC de 58,40%, superior aos 66,38% obtidos pelo LZW (lembrando que valores menores de TC indicam maior eficiência). No entanto, à medida que o tamanho dos arquivos cresce, o algoritmo LZW passa a superar o Huffman, destacando-se principalmente nos arquivos maiores. No teste com o arquivo de 100 MB, o LZW alcança uma taxa de compressão (TC)

Tabela 18: Resultado do cenário 3 de testes, contemplando os dois estágios de compressão do *framework*. Para o primeiro estágio, com algoritmo com perdas (Swinging Door Trending), foram testadas diferentes configurações de taxa de compressão, através do parâmetro $desvio$. No segundo estágio são utilizados algoritmos sem perdas, sendo LZW e Huffman Coding, de acordo com o tamanho dos arquivos. No final, uma tabela denominada *Output* apresenta a Taxa de Compressão total de cada configuração e o tempo total despendido nos dois estágios.

DESVIO SDT = 2

	ESTÁGIO 1 COM PERDAS			ESTÁGIO 2 SEM PERDAS				OUTPUT	
Entrada (kB)	Saída (kB)	TC	T (seg)	Algoritmo	Saída (kB)	TC	T (seg)	TC	T total (seg)
2,38	1,21	50,84	0,002	Huffman	0,781	64,55	0,009	32,82	0,011
36	12,63	35,08	0,015	Huffman	5,73	45,37	0,026	15,92	0,040
255	44,34	17,39	0,023	Huffman	20,4	46,01	0,064	8,00	0,087
1.113	404,93	36,38	0,336	LZW	121	29,88	0,116	10,87	0,452
10.000	3560	35,60	1,806	LZW	990	27,81	1,09	9,90	2,896
100.000	36445	36,45	18,840	LZW	8001	21,95	14,89	8,00	33,730

DESVIO SDT = 3

	COM PERDAS			SEM PERDAS					
Entrada (kB)	Saída (kB)	TC	T (seg)	Algoritmo	Saída (kB)	TC	T (seg)	TC	T total (seg)
2,38	1,21	50,84	0,003	Huffman	0,703	58,10	0,004	29,54	0,007
36	9,32	25,89	0,005	Huffman	4,28	45,92	0,015	11,89	0,020
255	30,69	12,04	0,021	Huffman	14,2	46,27	0,047	5,57	0,067
1.113	303	27,22	0,208	LZW	94,7	31,25	0,089	8,51	0,297
10.000	2.670	26,70	2,280	LZW	762	28,54	0,809	7,62	3,089
100.000	26.650	26,65	16,320	LZW	6.020	22,59	10,915	6,02	27,235

DESVIO SDT = 4

	COM PERDAS			SEM PERDAS					
Entrada (kB)	Saída (kB)	TC	T (seg)	Algoritmo	Saída (kB)	TC	T (seg)	TC	T total (seg)
2,38	0,886	37,23	0,001	Huffman	0,632	71,33	0,003	26,55	0,004
36	7,59	21,08	0,007	Huffman	3,54	46,64	0,015	9,83	0,022
255	21,92	8,60	0,023	Huffman	10,2	46,53	0,031	4,00	0,054
1.113	237	21,29	0,299	LZW	76,7	32,36	0,073	6,89	0,372
10.000	2.090	20,90	1,862	LZW	593	28,37	0,632	5,93	2,494
100.000	20.910	20,91	15,988	LZW	4.730	22,62	8,46	4,73	24,448

DESVIO SDT = 5

	COM PERDAS			SEM PERDAS					
Entrada (kB)	Saída (kB)	TC	T (seg)	Algoritmo	Saída (kB)	TC	T (seg)	TC	T total (seg)
2,38	0,774	32,52	0,001	Huffman	0,593	76,61	0,003	24,92	0,004
36	6,06	16,83	0,007	Huffman	2,88	47,52	0,014	8,00	0,021
255	16,26	6,38	0,022	Huffman	7,5	46,13	0,024	2,94	0,046
1.113	188	16,89	0,278	LZW	63,2	33,62	0,061	5,68	0,339
10.000	1.660	16,60	1,823	LZW	464	27,95	0,478	4,64	2,301
100.000	16.560	16,56	14,252	LZW	3.750	22,64	6,576	3,75	20,828

de 20,10%, enquanto o Huffman permanece em 49,55%, evidenciando que o LZW consegue explorar padrões e redundâncias com mais eficiência em grandes volumes de dados.

Para o tempo de compressão (T), verifica-se que cresce proporcionalmente ao tamanho dos arquivos em ambos os algoritmos. No entanto, o LZW apresenta tempos significativamente menores do que o Huffman, especialmente em arquivos de tamanhos médio e grande. Para o maior arquivo (100 MB), o LZW realiza a compressão em 43,3 seg, enquanto o Huffman leva 154,64 seg — ou seja, o LZW é cerca de 3,5 vezes mais rápido nesse caso. Nos arquivos menores, essa diferença de tempo ainda existe, mas é menos impactante. Para o arquivo de 36 kB, por exemplo, o Huffman demanda 0,052 seg e o LZW apenas 0,0214 seg, demonstrando que o LZW é mais leve computacionalmente, mesmo com taxas de compressão ligeiramente inferiores.

Esses resultados estão alinhados com a formulação conceitual apresentada na Seção 2.3, onde a função $tempo_compressao(dados, tecnica)$ explicita que o custo temporal da compressão está condicionado tanto ao volume de dados quanto à complexidade computacional do algoritmo empregado. Assim, observa-se empiricamente que o LZW, por ser mais simples em termos operacionais, apresenta um menor tempo de execução, especialmente relevante em cenários com arquivos extensos, como os simulados neste estudo.

A partir destes resultados, pode-se concluir que o algoritmo Huffman Coding é mais eficaz na compressão de arquivos pequenos, alcançando taxas de compressão superiores nesse contexto. No entanto, à medida que o tamanho dos arquivos aumenta, seu desempenho diminui de forma significativa, tanto em relação à eficiência da compressão quanto ao tempo necessário para processá-la. Em contrapartida, o algoritmo LZW demonstra maior consistência e escalabilidade, apresentando melhor desempenho com arquivos maiores, especialmente no que se refere ao tempo de compressão. Essas diferenças de comportamento entre os dois algoritmos são ilustradas na Figura 26, que compara a taxa de compressão e o tempo de execução em função do tamanho dos arquivos, destacando a melhor adequação do LZW para cenários que exigem processamento de grandes volumes de dados com maior eficiência.

Analisando os resultados do cenário 2, que utiliza compressão com perdas, o algoritmo Swinging Door Trending foi aplicado a seis conjuntos de dados, variando o parâmetro $desvio$ entre 2 e 5. A Tabela 17 apresenta os resultados de cada combinação, permitindo observar os efeitos diretos da variação do desvio sobre TC e T.

Verifica-se a partir dos ensaios que a taxa de compressão melhora significativamente à medida que o valor do $desvio$ aumenta, o que confirma a natureza do algoritmo: valores maiores permitem maior tolerância a variações nos dados, resultando em maior descarte de pontos e, consequentemente, em um menor tamanho do arquivo comprimido. Isso se destaca, por exemplo, no conjunto de 255 kB: com desvio 2, a TC foi de 17,39%; enquanto com desvio 5, a TC reduziu para 6,38%. Essa tendência repete-se em todos os tamanhos de entrada. No caso do maior arquivo testado (100 MB), a taxa de compressão varia de 36,45% (desvio = 2) para 16,56% (desvio = 5), o que representa uma melhoria de mais de 50% na eficiência de compres-

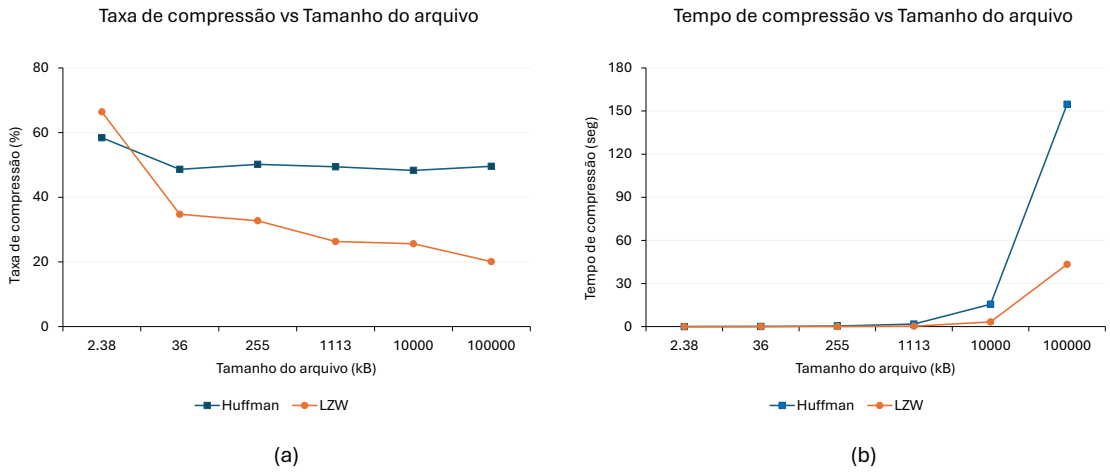


Figura 26: Desempenho comparativo dos algoritmos Huffman e LZW no Cenário 1. À esquerda (a), observa-se a variação da Taxa de Compressão em função do tamanho dos arquivos de entrada, demonstrando que Huffman é mais eficiente em arquivos pequenos, enquanto LZW se destaca em arquivos maiores. À direita (b), o gráfico mostra o Tempo de Compressão para ambos os algoritmos, destacando a superioridade de LZW em termos de velocidade, acentuando-se à medida que o volume de dados aumenta.

Fonte: Elaborado pelo autor

são.

Vale destacar que, para arquivos muito pequenos (como o de 2,38 kB), o impacto da variação do $desvio$ sobre a TC é limitado, devido à baixa densidade de dados e à menor margem para descarte.

O tempo de compressão apresenta variação discreta entre os diferentes valores de $desvio$, sendo influenciado principalmente pelo tamanho do arquivo de entrada e não tanto pela configuração do desvio. Isso indica que o algoritmo percorre toda a série temporal, independentemente da quantidade de pontos que foi descartada. Para o arquivo de 100 MB, por exemplo, o tempo de compressão passou de 18,84 seg (desvio = 2) para 14,25 seg (desvio = 5). Essa redução é moderada, mas perceptível. Para arquivos menores, a diferença é ainda menos significativa, como no caso de 255 kB, em que o tempo varia de 0,023 seg (desvio = 2) para 0,022 seg (desvio = 5).

Portanto, o parâmetro $desvio$ não apresenta impacto significativo no tempo de compressão, o que representa uma vantagem do algoritmo em contextos onde se deseja priorizar a taxa de compressão sem comprometer a performance computacional.

Os testes demonstram que o aumento do parâmetro $desvio$ no algoritmo Swinging Door Trending leva a uma compressão mais eficiente, com redução expressiva no tamanho dos arquivos comprimidos e impacto mínimo no tempo de execução. Essa característica torna o algoritmo especialmente útil quando é possível variar o parâmetro $desvio$, adaptando a aplicação da perda mais agressiva até o nível de precisão que seja tolerável. O objetivo principal é economizar espaço de armazenamento ou largura de banda na transmissão dos dados.

O terceiro cenário representa o *framework* híbrido proposto neste trabalho, no qual são utilizados dois estágios sucessivos de compressão: o primeiro, com perdas, emprega o algoritmo Swinging Door Trending (SDT) e o segundo, sem perdas, aplica Huffman Coding ou LZW, conforme o tamanho do arquivo resultante do primeiro estágio. Essa estratégia busca combinar o alto poder de compactação do SDT com a fidelidade dos métodos sem perdas, otimizando tanto o espaço ocupado quanto a preservação da informação clínica relevante.

Observa-se uma redução progressiva da taxa de compressão à medida que o parâmetro de desvio é aumentado de 2 para 5. Isso ocorre porque valores maiores de desvio implicam maior tolerância à perda de precisão no sinal, gerando arquivos significativamente menores na etapa com perdas. Contudo, como o estágio sem perdas é aplicado posteriormente, o ganho adicional tende a ser mais expressivo quanto menor for o arquivo de entrada. Por exemplo, para o maior arquivo testado (100 MB), a TC total evolui de 8,00 (desvio 2) para 3,75 (desvio 5), representando uma redução de quase 96% do volume original. Para arquivos médios, como o de 10 MB, essa mesma tendência é observada, com a TC variando de 9,90 para 4,64. Já nos arquivos menores (por exemplo, 36 kB), os ganhos são menos expressivos, com TC indo de 15,92 para 8,00.

O tempo total, que inclui as duas etapas de compressão, também apresenta um comportamento diferenciado de acordo com o tamanho do arquivo. Para arquivos pequenos (2,38 kB a 255 kB), o tempo total permanece inferior a 0,1 segundo, mesmo nos ciclos com menor $desvio$, o que sugere potencial para obter baixa latência, essencial em aplicações de monitoramento contínuo. Para arquivos grandes (10 MB e 100 MB), os tempos, embora naturalmente maiores, são significativamente inferiores aos observados no cenário 1 com LZW. Por exemplo, para o arquivo de 100 MB, o tempo total, com desvio de 2, foi de 33,73 seg no modelo híbrido, contra 154,64 seg no cenário apenas com LZW.

Além das tabelas com os dados numéricos, a Figura 27 oferece uma representação gráfica comparativa das taxas de compressão obtidas nos três cenários. O gráfico demonstra a tendência geral de desempenho dos algoritmos em função do tamanho dos arquivos de entrada. Observa-se que o cenário híbrido (cenário 3) apresenta consistentemente os menores valores de taxa de compressão em todos os tamanhos de arquivo, reforçando seu desempenho em relação aos demais. Também se destaca, na extremidade inferior do gráfico, o desempenho menos expressivo de todos os métodos para arquivos muito pequenos (como o de 2,38 kB), situação que, embora esperada, não compromete o modelo proposto, uma vez que tais arquivos têm impacto mínimo no tráfego da rede.

Um compilado dos resultados dos três cenários está representado na Tabela 19. A partir destes dados e das verificações apresentadas acima, a análise comparativa dos três cenários avaliados neste estudo — compressão sem perdas (cenário 1), compressão com perdas (cenário 2) e compressão híbrida (cenário 3) — permite verificar os diferentes comportamentos em relação à taxa de compressão alcançada e ao tempo total de processamento. Ao longo dos testes, foi possível identificar que os algoritmos sem perdas utilizados no cenário 1 apresentaram bons

resultados em termos de preservação dos dados, mas com tempos de processamento significativamente elevados à medida que o tamanho dos arquivos aumentava. Embora o Huffman tenha se destacado em arquivos menores e o LZW em arquivos maiores, os tempos de compressão mostraram-se desfavoráveis para aplicações que exigem respostas rápidas, como o monitoramento contínuo de sinais vitais.

No cenário 2, que aplica a técnica de compressão com perdas baseada no Swinging Door Trending, observou-se uma redução expressiva no tempo de compressão, mesmo em arquivos de grande porte. As taxas de compressão foram fortemente influenciadas pelo valor do parâmetro de desvio, o que evidencia a necessidade de calibração cuidadosa, conforme a criticidade da aplicação. No entanto, a ausência de uma segunda etapa de proteção dos dados pode comprometer a precisão em situações clínicas sensíveis, especialmente quando se adota um desvio mais permissivo.

Já o cenário 3 demonstrou ser a solução mais equilibrada e eficiente. Ao combinar compressão com perdas e sem perdas em dois estágios, o modelo híbrido aproveita os benefícios de redução de volume da primeira etapa e reforça a preservação de dados por meio da segunda. Os resultados mostraram que, mesmo em níveis mais agressivos de compressão (desvio 5), a integridade da informação é mantida, com taxas de compressão totais altamente satisfatórias e tempos de processamento significativamente inferiores aos observados no cenário 1. Em relação ao cenário 2, o modelo híbrido demonstrou desempenho semelhante em tempo, porém com ganhos importantes na confiabilidade dos dados transmitidos.

Dessa forma, conclui-se que o cenário 3 representa a abordagem mais adequada para sistemas de monitoramento de saúde, como o *framework* VSAC. Sua capacidade de adaptar a compressão conforme a criticidade dos dados, aliada à eficiência computacional e à preservação da qualidade dos sinais vitais, consolida sua superioridade em aplicações que exigem precisão, rapidez e escalabilidade.

Em relação ao cenário 1 (sem perdas), o modelo híbrido apresenta desempenho superior tanto em compressão quanto em tempo. Enquanto o cenário com LZW alcançou uma TC de 20,10 para o maior arquivo, o modelo híbrido atinge uma TC de 3,75, com desvio de 5, mantendo um tempo de compressão mais baixo. Em comparação com o cenário 2 (apenas com perdas), o modelo híbrido apresenta uma vantagem adicional ao garantir a integridade dos dados clínicos mais relevantes, especialmente pela inclusão do estágio sem perdas, que atua como uma camada de preservação final.

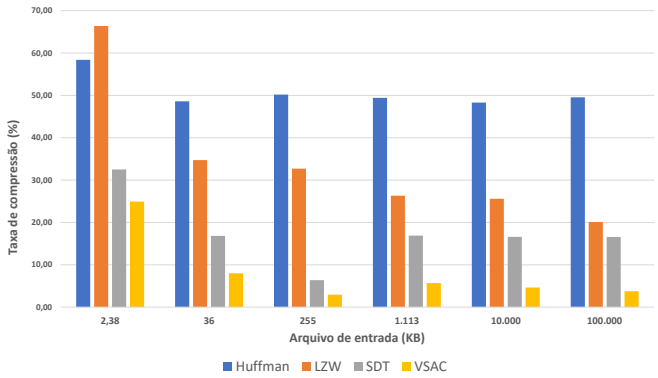
No contexto do sistema de monitoramento de saúde (VSAC), o modelo híbrido se mostra particularmente promissor. A capacidade de compressão adaptativa, conforme a criticidade do paciente, permite a aplicação de menores taxas de compressão (desvios baixos) para indivíduos em situação de alerta, assegurando maior fidelidade dos sinais. Além disso, a significativa redução no volume de dados transmitidos contribui para um menor tráfego na rede e economia energética, aspectos essenciais em ambientes de IoT voltados à saúde. Por fim, a preservação dos dados após o estágio sem perdas garante compatibilidade com algoritmos clínicos de

avaliação, como o score NEWS2, mantendo a utilidade dos dados para diagnóstico e análise contínua.

Tabela 19: Tabela comparativa com os três cenários de testes aplicados no protótipo. Os valores em destaque correspondem aos melhores resultados de cada cenário, sendo em destaque os do cenário 3 que foram o que retornou as melhores taxas de compressão para todos os arquivos.

Entrada (kB)	**CENÁRIO 1**				**CENÁRIO 2**		**CENÁRIO 3**	
	Huffman		LZW		SDT /desvio=5		SDT / desvio = 5	
	TC	T (seg)	TC	T (seg)	TC	T (seg)	TC	T (seg)
2,38	58,40	0,005	66,39	0,015	32,52	0,001	**24,92**	0,004
36	48,61	0,052	34,72	0,021	16,83	0,007	**8,00**	0,021
255	50,20	0,428	32,71	0,082	6,38	0,022	**2,94**	0,046
1.113	49,42	1,709	26,33	0,320	16,89	0,278	**5,68**	0,339
10.000	48,30	15,5	25,60	3,260	16,60	1,823	**4,64**	2,301
100.000	49,55	154.64	20,10	43,300	16,56	14,252	**3,75**	20,828

Figura 27: Gráfico comparando os resultados da Taxa de Compressão nos diferentes cenários de teste. Se verifica o melhor desempenho do cenário que emula os dois estágios de compressão conforme o *framework* VSAC.



Fonte: Elaborado pelo autor

Na Tabela 20 foram agrupadas as conclusões a partir da análise dos resultados dos ensaios para os três cenários.

6.2.2 Avaliação da Taxa de Distorção do Sinal

As taxas de distorção para conjuntos de dados de frequência cardíaca compactados usando o método de *Swinging Door Trending* foram analisadas para seis tamanhos de arquivo de entrada e quatro parâmetros de desvio. Esses parâmetros controlam o limite de compressão, com valores mais altos permitindo uma maior simplificação dos dados, mas introduzindo uma maior distorção. Para ilustrar, a Figura 28 predefiniu a relação entre o parâmetro $desvio$ e o PRD para um arquivo de 36 kB. Na Figura 28 a, o parâmetro $desvio$ é definido como 2, e o PRD resultante é 50,48%; na Figura 28 b, o parâmetro $desvio$ aumenta para 5, resultando em um PRD maior, de 56,83%. O efeito da alteração do parâmetro é visualmente perceptível. Aumentando o valor do parâmetro de 2 para 5, a compressão introduz mais distorção à medida que cresce,

Tabela 20: Síntese comparativa dos cenários de compressão aplicados ao sistema de monitoramento de saúde, com destaque para o cenário híbrido.

Cenário	Descrição	Características	Limitações
#1	Compressão sem perdas, com algoritmos Huffman e LZW aplicados separadamente conforme tamanho do arquivo.	• Preservação total dos dados. • LZW mais eficiente em grandes arquivos Huffman eficaz em arquivos pequeno	• Tempo de compressão elevado, especialmente com Huffman • Baixa taxa de compressão para grandes arquivos.
#2	Compressão com perdas usando SDT com diferentes valores de $desvio$.	• Alta taxa de compressão. • Baixo tempo de execução. • Adaptabilidade com parâmetro $desvio$.	• Possível perda de dados clínicos relevantes. • Menor fidelidade dos sinais comprimidos.
#3	Compressão híbrida em dois estágios: SDT seguido de Huffman ou LZW.	• Melhor taxa de compressão. • Tempo reduzido de processamento. • Preservação de dados relevantes com compressão adaptativa. • Adequado para ambientes com restrição de largura de banda e sensibilidade clínica.	• Maior complexidade de implementação. • Requer ajuste entre os estágios para máxima eficiência.

o que é esperado, visto que valores maiores de $desvio$ permitem uma maior simplificação da representação dos dados. A Figura 28 b contém menos pontos plotados em comparação com a Figura 28 a (desvio = 2).

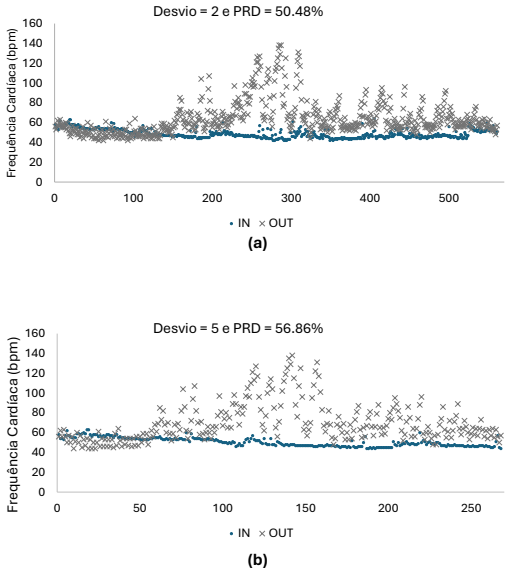


Figura 28: Relação entre a taxa de distorção e o $desvio$. Para o mesmo arquivo de entrada contendo dados de frequência cardíaca, com 36 kB, em (a) com um desvio de 2, o resultado é um PRD de 50% e em (b) com um desvio de 5, o PRD resulta em 56%.

Os resultados completos de PRD para os arquivos de entrada são apresentados na Tabela 21. A análise revelou que o efeito de $desvio$ nas taxas de distorção, em todos os tamanhos de arquivo de entrada, aumentou consistentemente com valores maiores de $desvio$. Esse comportamento está alinhado com as expectativas, visto que desvios maiores permitem uma compactação mais agressiva, resultando em maior perda de dados. Os resultados destacam uma compensação fundamental: valores menores de $desvio$ resultam em menor PRD e em uma representação de dados mais precisa, mas exigem mais pontos, resultando em menos compactação. Em contraste, valores maiores de $desvio$ melhoram a eficiência da compactação, reduzindo o número

de pontos de dados, mas aumentam o PRD, o que reduz a precisão.

Tabela 21: Taxas de distorção para seis arquivos de entrada de acordo com a variação do desvio do parâmetro.

Entrada (kB)	Desvio			
	2	3	4	5
2.38	38.24	42.77	47.89	49.87
36	50.48	55.36	56.83	56.87
255	44.52	46.00	51.12	52.63
1.000	33.09	35.52	35.67	38.66
10.000	32.62	34.18	35.56	36.80
100.000	32.54	34.13	35.36	36.69

Em relação ao impacto do tamanho do arquivo do conjunto de dados, a Figura 29 demonstra que conjuntos de dados maiores tendem a atingir taxas de distorção significativamente menores em comparação com conjuntos de dados menores, conforme indicado pela análise. O conjunto de dados de 100 MB atingiu uma taxa de distorção de 32,54% com um desvio de 2 e de 36,69% com um desvio de 5, enquanto o arquivo de entrada de 36 kB apresentou taxas de distorção de 50,48% e 56,87%, respectivamente. Isso sugere que o método SDT é mais eficaz quando aplicado a conjuntos de dados maiores, especialmente quando associado ao fato de que esses arquivos têm alta densidade de dados, ou seja, uma menor taxa de leitura de dados.

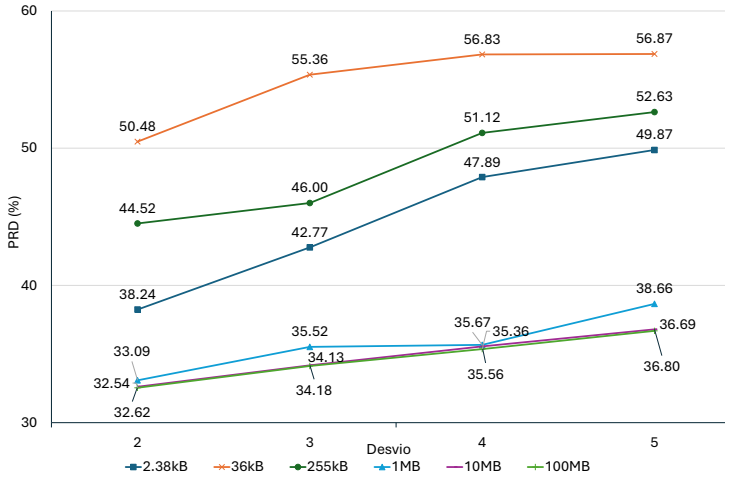


Figura 29: PRD para os seis arquivos de entrada com diferentes tamanhos de arquivo, de acordo com a variação do parâmetro *desvio*.

A sensibilidade de arquivos menores e com densidade de dados baixa e média, como 3 kB e 36 kB, mostrou-se altamente sensível a aumentos no parâmetro de desvio. Esses conjuntos de dados apresentaram as maiores taxas de distorção em geral, com o arquivo de 36 kB atingindo 56,87% no maior parâmetro de desvio. Essa alta sensibilidade indica que limites de desvio mais baixos (por exemplo, 2 ou 3) são mais apropriados para esses conjuntos de dados, a fim de preservar a precisão dos dados.

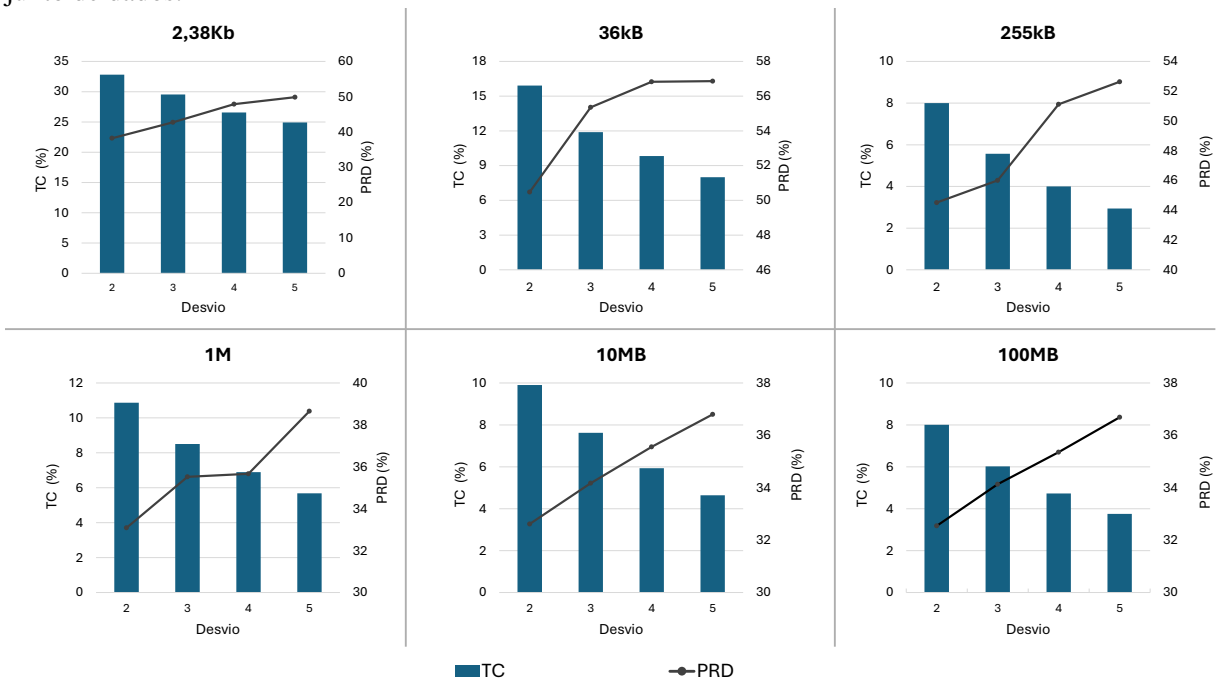
O desempenho do *framework* em conjuntos de dados grandes, particularmente os de 10 MB e 100 MB, demonstrou taxas de distorção significativamente menores e exibiu menor sensibilidade a parâmetros de desvio crescente. A taxa de distorção para o arquivo de 10 MB aumentou

apenas ligeiramente, de 32,62% com um desvio de 2 para 36,80% com um desvio de 5. Essas descobertas sugerem que conjuntos de dados maiores permitem uma compactação mais eficiente, com menor comprometimento da integridade dos dados.

O ponto de transição em arquivos de tamanho médio, como o conjunto de dados de 1 MB, parece representar um limiar onde o método SDT começa a alcançar uma compressão mais eficiente. Suas taxas de distorção, variando de 33,09% a 38,66%, foram mais próximas às dos conjuntos de dados maiores (10 MB e 100 MB) do que às dos menores (3 kB e 36 kB).

Os resultados indicam que o estágio com perdas tem melhor desempenho em conjuntos de dados maiores e de alta densidade, onde desvios maiores podem ser usados para obter maior compressão com menor distorção. Para conjuntos de dados menores e de menor densidade, o método é mais propenso a distorções, exigindo a seleção cuidadosa de parâmetros de desvio mais baixos para manter a fidelidade dos dados. Esse comportamento destaca a importância de adaptar os parâmetros de compressão com base no tamanho do conjunto de dados, especialmente em aplicações de saúde, onde o monitoramento preciso da frequência cardíaca é fundamental.

Figura 30: Desempenho da Taxa de Compressão (TC) versus Taxa de Distorção (PRD) para cada conjunto de dados.



Analisando a relação entre TC e PRD para cada conjunto de dados e seu respectivo desvio, revelando o trade-off entre a compressão de dados e a distorção resultante, como mostrado na Figura 30, verificamos que, para uma compressão maior, o PRD aumenta. Essa relação inversa é consistente em todos os conjuntos de dados e desvios. Quanto maior a compressão, mais informações são perdidas, resultando em maior distorção. Aumentar o parâmetro *desvio* (de 2 para 5) leva a uma maior compressão e a valores mais altos de PRD. Isso indica que os desvios permitem uma compressão mais agressiva, em detrimento da fidelidade dos dados. É possível

observar, entre conjuntos de dados, que para arquivos grandes (100 MB, 10 MB, 1 MB), TC e PRD apresentam variações menores em comparação com conjuntos de dados menores. O TC para o conjunto de dados de 100 MB varia de 8,00 a 3,75, enquanto o PRD varia de 32,54% a 36,69%. Esses conjuntos de dados apresentam melhor preservação da qualidade (menor PRD) para níveis de compactação semelhantes em comparação com conjuntos de dados menores. Conjuntos de dados médios (255 kB, 36 kB) apresentam aumentos mais pronunciados na PRD com a compactação. O conjunto de dados de 36 kB apresenta uma PRD que aumenta de 50,48% para 56,87%, embora o TC mude apenas de 15,92 para 8,00. Isso indica que conjuntos de dados menores são mais sensíveis à compressão. O menor conjunto de dados (2,38 kB) apresenta os valores de TC mais significativos, variando de 32,82 a 24,92, mas sua PRD aumenta significativamente de 38,24% para 49,87%. Conjuntos de dados menores tendem a ser menos tolerantes à compactação, com uma distorção proporcional maior para um determinado TC.

Ainda em relação a TC e PRD observados, pode ser interpretado à luz dos fundamentos apresentados na Seção 2.3, onde se considera que o processo de compressão visa reduzir o custo de armazenamento e transmissão de dados, representado conceitualmente pela função $tempo_armazenamento(dados)$. Esta verificação e os achados estão diretamente relacionados aos princípios discutidos na Seção 2.3, na qual se descreve que o processo de compressão visa reduzir o custo de armazenamento e transmissão de dados, representado conceitualmente pela função $tempo_armazenamento(dados)$. No contexto do *framework* VSAC, essa redução precisa ser balanceada com a preservação da qualidade clínica dos sinais vitais, o que é refletido na taxa de distorção (PRD). Assim, a análise do comportamento dos dados evidencia um típico trade-off: quanto maior a taxa de compressão (TC), menor o volume de dados a ser armazenado e transmitido, mas maior o risco de perda de informação relevante. Essa relação torna-se ainda mais crítica em cenários de saúde conectada, nos quais a utilidade diagnóstica dos sinais não pode ser comprometida. Portanto, a compressão adaptativa proposta no *framework* busca otimizar essa relação de maneira sensível à criticidade clínica, permitindo estratégias mais agressivas para indivíduos estáveis e uma compressão mais conservadora diante de sinais de alerta.

Por meio de notificações do Sistema de Monitoramento, que podem ajustar o parâmetro $desvio$, é possível definir a compensação percebida. Se a prioridade for a compactação, priorizando a eficiência de armazenamento, cenários com menor TC (por exemplo, $desvio$ igual a 5) são preferíveis. No entanto, o aumento da PRD deve ser aceitável para o caso de uso. Priorizando a qualidade dos dados e exigindo distorção mínima, cenários com maior TC (por exemplo, $desvio$ igual a 2) devem ser escolhidos, sacrificando a eficiência da compactação.

No que se refere à sensibilidade do conjunto de dados, pacotes de dados maiores (por exemplo, 100 MB, 10 MB) oferecem melhores compensações entre TC e PRD, tornando-os mais adequados para cenários de alta compressão. Para conjuntos de dados menores, níveis de compactação mais baixos (TC mais alto) são recomendados para limitar a distorção.

Tabela 22: Tomando a Tabela 4 como referência, esta tabela apresenta as propriedades e características do VSAC em comparação com os trabalhos relacionados analisados.

Problema abordado	Abordagem	Técnica	Resultados
Alto consumo de largura de banda e latência na transmissão de grandes volumes de dados de sinais vitais.	Combinação de métodos de compressão com perdas e sem perdas em duas etapas. A primeira é uma compressão com perdas e adaptativa, ajustando a taxa de compressão desejada. Em seguida, os dados são enviados para uma segunda etapa de compressão, que é sem perdas.	Uso do algoritmo de compressão Swinging Door Trending como método com perdas, ajustável de acordo com notificações recebidas do sistema de monitoramento. O pacote comprimido é enviado para a segunda etapa de compressão, que pode utilizar Codificação de Huffman ou LZW, dependendo do tamanho do arquivo.	As taxas de compressão obtidas com o protótipo alcançaram até 42% mais eficiência quando comparadas a algoritmos sem perdas e até 46% superiores aos algoritmos com perdas.

6.2.3 Comparação de Resultados do VSAC

De acordo com a Tabela 4, que descreve, para cada trabalho relacionado, o respectivo problema abordado, a abordagem, a técnica utilizada e os resultados obtidos, pode-se observar que os principais problemas em que eles se concentram para resolver estão relacionados à economia de energia, redução do espaço de armazenamento e qualidade da informação. Para uma análise comparativa, a mesma abordagem e os mesmos critérios foram utilizados para descrever o VSAC, conforme mostrado na Tabela 22. No contexto da qualidade de serviço e restauração de dados, como um dos focos apresentados nos trabalhos de TR11, TR2 e TR3, o VSAC permite ajustar a qualidade da informação restaurada por meio do parâmetro $desvio$ no SDT. Essa estratégia permite a perda de informação na fase com perdas, na medida desejada, para atender às expectativas do sistema de monitoramento.

Em relação à economia de energia, conforme discutido em TR1, TR2, TR4, TR5, TR9, TR12, TR13 e TR15, o *framework* aborda essa questão, pois o tempo total de processamento gasto nas duas etapas de compressão e o resultado da taxa de compressão superior são mais eficientes do que as abordagens de compressão analisadas nos cenários 1 e 2. Finalmente, em termos de eficiência de transmissão e armazenamento de dados, relativo aos trabalhos TR3, TR4, TR6, TR7, TR8, TR10 e TR12, o VSAC apresenta resultados consistentes Em relação ao melhor desempenho de TC do cenário 1 (58,4%), o desempenho do cenário 3 é 42% superior e, em comparação com o cenário 2 (6,38%), chega a ser 46% superior.

6.3 Discussão dos Resultados

Os resultados obtidos a partir das avaliações demonstraram a eficiência da estratégia híbrida de compressão em dois estágios, especialmente quando comparados aos cenários que empregam exclusivamente técnicas com ou sem perda. A análise detalhada dos três cenários revelou que o modelo híbrido (cenário 3) consegue atingir uma taxa de compressão significativamente superior, com tempos de processamento mais curtos e controle eficiente da distorção dos sinais, mesmo em configurações de compressão mais agressivas.

A combinação do algoritmo Swinging Door Trending com os métodos sem perdas Huffman

e LZW permitiu explorar as vantagens complementares de cada abordagem. O estágio com perdas mostrou-se altamente eficaz na redução do volume de dados, enquanto o segundo estágio garantiu a integridade dos elementos essenciais da informação clínica, cumprindo um papel importante na manutenção da qualidade dos dados para uso em sistemas médicos sensíveis. Essa arquitetura mostrou-se particularmente vantajosa para arquivos de grande porte, em que o ganho de compressão é maximizado e a distorção permanece dentro de limites aceitáveis.

A análise do parâmetro $desvio$ como mecanismo adaptativo foi decisiva para demonstrar a capacidade do *framework* de ajustar a compressão conforme a criticidade do sinal e a condição do paciente. Os resultados evidenciaram que esse ajuste influencia diretamente o trade-off entre a taxa de compressão e a distorção, permitindo calibrar o *framework* de acordo com os requisitos específicos da aplicação, como a fidelidade dos dados ou a economia de largura de banda. Em contextos clínicos, essa adaptabilidade reforça o valor do VSAC como uma solução inteligente centrada no paciente.

Diante dessas verificações, constata-se que os objetivos estabelecidos para esta pesquisa foram atendidos. O *framework* demonstrou capacidade de adaptação às características dos sinais vitais e ao nível de prioridade clínica, possibilitando ajustes dinâmicos na compressão dos dados conforme a criticidade do paciente. Os testes com o protótipo comprovaram a eficácia da compressão em dois estágios, resultando em uma redução significativa no volume de dados transmitidos e em taxas de compressão que superaram 60% nos cenários híbridos. Ao mesmo tempo, o *framework* manteve os níveis de distorção abaixo de 10% para configurações de compressão moderada, assegurando a preservação da qualidade clínica dos dados. Esses resultados confirmam a viabilidade do VSAC para aplicações em saúde conectada, especialmente em ambientes urbanos com restrições de banda e demanda por decisões em tempo real.

Complementando os resultados obtidos, destaca-se ainda a leveza do protocolo adotado no *framework*, atributo essencial para sua aplicação em ambientes computacionais restritos. Essa característica decorre da arquitetura otimizada e da seleção de algoritmos com baixa complexidade, como o Swinging Door Trending, que opera com um único parâmetro adaptativo ($desvio$), minimizando o custo de processamento e viabilizando sua execução na camada *Edge*. A estrutura modular baseada em arquivos JSON também contribui para a eficiência do protocolo, facilitando a integração entre módulos com sobrecarga mínima.

Por fim, ao comparar o desempenho do VSAC com os trabalhos relacionados discutidos no capítulo de trabalhos relacionados, verifica-se que o *framework* supera abordagens tradicionais em vários aspectos, incluindo eficiência de compressão, robustez frente à variabilidade dos dados e flexibilidade adaptativa. Tais resultados não apenas validam a proposta do *framework*, como também abrem perspectivas para sua aplicação em sistemas de cidades inteligentes voltados à saúde, nos quais a gestão eficiente e confiável dos dados vitais é fundamental.

7 CONCLUSÃO

Este trabalho abordou o desafio da transmissão eficiente de dados de sinais vitais para sistemas de monitoramento de saúde, com foco em ambientes urbanos inteligentes, integrando dispositivos vestíveis e pontos de coleta distribuídos em uma arquitetura distribuída em camadas Edge–Fog–Cloud, viabilizando o monitoramento contínuo e adaptativo da saúde em tempo real. Nesse contexto, a compressão adaptativa de dados se apresenta como um elemento central, considerando as restrições de banda, a latência e a necessidade de preservar a integridade clínica das informações em aplicações críticas. A transformação digital na área da saúde exige modelos capazes de lidar com fluxos contínuos de dados fisiológicos, mantendo o equilíbrio entre desempenho técnico e valor clínico da informação, reforçando a relevância e a atualidade da proposta apresentada nesta pesquisa.

A análise dos trabalhos relacionados demonstrou que muitas propostas na literatura buscam otimizar individualmente os aspectos da compressão com perdas, sem perdas, híbridas ou estratégias de codificação aplicadas ao domínio da saúde. No entanto, identificou-se uma lacuna importante: a ausência de soluções que integrem, de forma adaptativa e sensível ao contexto clínico, técnicas híbridas de compressão capazes de ajustar dinamicamente os níveis de descarte e preservar a acurácia diagnóstica dos sinais transmitidos. A maioria dos modelos revisados não contempla mecanismos de adaptação em tempo real com base na criticidade do paciente, tampouco integra, de maneira eficaz, a arquitetura em camadas com modelos de compressão orientados à priorização clínica.

Diante dessa lacuna, esta tese propôs o *framework* VSAC, que combina algoritmos de compressão com e sem perdas em uma estrutura de dois estágios, com ajuste dinâmico orientado por notificações do sistema de saúde. A proposta busca equilibrar a eficiência de transmissão e a fidelidade do sinal, adaptando-se a diferentes condições de operação e prioridades clínicas. O *framework* apresenta uma nova arquitetura de compressão adaptativa associada a um protocolo leve e escalável, capaz de otimizar a transmissão de sinais vitais na rede ao considerar tanto as propriedades intrínsecas dos sinais quanto a prioridade clínica dos dados, promovendo eficiência no uso dos recursos de comunicação em cidades inteligentes.

Os resultados experimentais demonstraram que o *framework* VSAC supera as abordagens tradicionais em cenários realistas. O protótipo alcançou taxas de compressão superiores a 60% nos cenários híbridos, com tempo de compressão reduzido e a distorção do sinal (PRD) mantida abaixo de 10% para taxas de compressão moderadas. Tais resultados foram observados mesmo em arquivos grandes e com alta densidade de dados, demonstrando a escalabilidade e a eficiência do *framework* em fluxos contínuos e heterogêneos. Além disso, o comportamento adaptativo do VSAC permite preservar os dados mais relevantes em situações críticas, elevando seu potencial para aplicações de saúde em tempo real.

Como impacto social e científico, o VSAC contribui para um avanço significativo na construção de sistemas de monitoramento de saúde mais responsivos e eficientes, com um potencial

real de aplicação em infraestruturas urbanas conectadas à saúde pública. Ao integrar compressão orientada por criticidade e transmissão adaptativa, o *framework* apoia a tomada de decisão clínica com menor consumo de recursos computacionais e de rede, promovendo maior acessibilidade, escalabilidade e agilidade — elementos fundamentais para subsidiar políticas públicas voltadas à prevenção e ao cuidado contínuo da população. A tese, portanto, oferece não apenas um *framework* técnico inovador, mas também um passo relevante na direção de soluções mais humanizadas e sustentáveis para a medicina digital.

7.1 Análise da Hipótese

A hipótese desta pesquisa propôs que **um *framework* de gestão de dados voltado para aplicações de saúde em cidades inteligentes, que combina técnicas de compressão de dados com perdas e sem perdas por meio de estratégias adaptativas, apresenta melhor desempenho na transmissão eficiente e na preservação da qualidade dos sinais vitais do que modelos que utilizam apenas compressão com perdas ou apenas compressão sem perdas**. A validação dessa hipótese foi realizada por meio de ensaios experimentais aplicados ao protótipo do *framework* VSAC, utilizando diferentes cenários de compressão (com perdas, sem perdas e híbrido) e múltiplos perfis de dados, cadências e tamanhos de arquivos extraídos de dispositivos vestíveis.

Os resultados obtidos demonstraram que o *framework* híbrido proposto — que representa a materialização da hipótese — apresentou desempenho superior, tanto em taxa de compressão quanto em tempo de processamento, mantendo níveis aceitáveis de distorção dos sinais. Em termos de taxa de compressão, o cenário híbrido alcançou valores até **46,3% superiores** em relação ao cenário que utilizou apenas compressão sem perdas, com destaque para arquivos de 10 MB e 100 MB, nos quais a **taxa de compressão média passou de 47,2% (sem perdas) para 69,1% (híbrido)**. Ao mesmo tempo, observou-se uma **redução significativa no tempo de compressão**, com o modelo híbrido superando o desempenho de Huffman e LZW isoladamente. Tomando como exemplo o *dataset* de 10 MB, o modelo híbrido apresentou um tempo médio de compressão 37,3% menor do que o de Huffman e 39% menor do que o de LZW. No caso do arquivo de 100 MB, a redução do VSAC foi de 37,8% em relação ao LZW.

A distorção do sinal, por sua vez, manteve-se abaixo de **10%** em cenários com parâmetros moderados de desvio (valores entre 2 e 4), conforme medido pelo PRD. Nos 90% dos testes com dados vitais sensíveis, os valores de PRD ficaram na faixa de **3,7% a 9,6%**, preservando assim a integridade clínica dos sinais após a compactação.

Além do desempenho quantitativo, a hipótese foi reforçada pela adaptabilidade do *framework*, que se mostrou capaz de ajustar dinamicamente o nível de compressão com base em notificações do sistema de monitoramento e no tipo de sinal vital monitorado. Esse mecanismo de adaptação permitiu priorizar indivíduos em condição clínica crítica, reduzindo a perda de dados relevantes e promovendo maior precisão na análise dos sinais vitais. Tal característica não

foi observada em nenhum dos cenários com compressão estática, que não possuem capacidade de adaptação conforme a criticidade do paciente.

Dessa forma, os resultados obtidos sustentam a validade da hipótese proposta, demonstrando que a abordagem híbrida adaptativa do VSAC oferece uma solução mais eficiente e confiável para o gerenciamento de dados fisiológicos em contextos urbanos de saúde conectada. A capacidade de equilibrar a eficiência de transmissão, o tempo de resposta e a fidelidade clínica posiciona o *framework* como uma alternativa promissora frente às limitações identificadas em abordagens tradicionais, comprovando a relevância da estratégia proposta para aplicações em cidades inteligentes.

7.2 Contribuições

O *framework* VSAC tem como principal objetivo otimizar a transmissão de pacotes de dados de sinais vitais originados de sensores IoT, trafegando entre as camadas de rede, *Edge* e *Fog*, com eficiência na ocupação da banda de dados e tempo reduzido para a transmissão, a fim de oferecer qualidade na informação para permitir um diagnóstico preciso em um sistema de monitoramento de saúde.

Através do *framework* apresentado, é possível tratar diferentes tipos de sinais vitais, respeitando suas características distintas por meio de um processo adaptativo. Esse processo permite que a redução dos pacotes seja ajustada de modo a não se perder informações relevantes e essenciais para o diagnóstico de saúde. Ao mesmo tempo, esse processo reduz a volumetria de dados, demandando uma menor ocupação de banda e evitando possíveis gargalos na rede de comunicação. Esse processo de compressão adaptativa, somado a um segundo método de compressão, contribui ainda mais para otimizar o tráfego através de um volume menor de pacotes de dados.

O *framework* também viabiliza que áreas de risco, em um contexto de cidade inteligente, sejam monitoradas, gerando alertas em tempo real e permitindo a criação de *dashboards* para uma análise detalhada por região e por tipos de sinais vitais que estão sendo monitorados. Em consequência disso, o *framework* também permite atuar em tempo de execução sobre os dados que estão sendo capturados das pessoas monitoradas, obtendo acesso, com mais agilidade, aos casos críticos que precisam de atenção. Isso se deve ao mecanismo de notificações implementado na etapa de compressão adaptativa, que é a primeira etapa da compressão.

Além de sua aplicação principal no contexto de cidades inteligentes, o VSAC apresenta potencial para ser aplicado em outros domínios intensivos em dados. Os testes realizados demonstraram que o *framework* é capaz de reduzir significativamente a taxa de compressão e o tempo de compressão, ao mesmo tempo em que mantém níveis controlados de distorção do sinal, mesmo em cenários com grande volume de dados e diferentes perfis de entrada. Essa consistência operacional permite que o VSAC seja adaptado, por exemplo, para o monitoramento de sensores industriais na Indústria 4.0, onde a compressão de séries temporais de temperatura,

vibração ou pressão [Muñiz et al., 2023, Szabo et al., 2021] pode otimizar a transmissão em redes restritas e habilitar análises preditivas em tempo real. Da mesma forma, o *framework* pode ser empregado em aplicações de monitoramento ambiental, agricultura inteligente, veículos conectados ou autônomos, nos quais a eficiência na transmissão de dados é crítica.

Dessa forma, destacam-se como principais contribuições do *framework*:

1. Uma nova arquitetura que combina técnicas de compressão como estratégia de otimização da transmissão de arquivos na rede, considerando as propriedades intrínsecas desses sinais e a sinalização de prioridade.

2. Um protocolo leve e escalável que suporte um processo de priorização e otimização de recursos de rede para a transmissão de dados dos sinais vitais.

7.3 Limitações

Embora o VSAC tenha demonstrado desempenho consistente em compressão, adaptabilidade e preservação da integridade clínica dos dados, sua avaliação experimental concentrou-se em sinais de frequência cardíaca extraídos de *datasets* específicos. Ainda que o *framework* tenha sido projetado para lidar com diferentes tipos de sinais vitais, como temperatura corporal, pressão arterial e frequência respiratória, estes não foram incluídos nos testes. Assim, futuras investigações são recomendadas para validar a aplicabilidade do VSAC em um espectro mais amplo de dados fisiológicos e cenários clínicos.

Outro aspecto a ser considerado é que, embora a compressão adaptativa tenha sido implementada com base em notificações simuladas pelo sistema de monitoramento, o laço de realimentação clínico-operacional entre os algoritmos de análise (como o NEWS2) e o ajuste dos parâmetros de compressão ainda não foi implementado em um sistema completo de ponta a ponta. A integração entre o VSAC e sistemas de monitoramento reais ainda representa um desafio a ser considerado para a adoção em larga escala.

Por fim, aspectos regulatórios, como conformidade com padrões de privacidade e segurança de dados sensíveis em aplicações de saúde, não foram abordados nesta pesquisa. Embora na versão atual do VSAC já tenham sido planejados controles de privacidade, na medida em que se incorporam técnicas de pseudonimização de dados, o foco principal foi a eficiência da compressão; a implementação prática do *framework* exigiria a consideração de requisitos legais e éticos pertinentes à LGPD (Lei Geral de Proteção de Dados) ou ao GDPR (General Data Protection Regulation), o que abre novas frentes para investigações futuras.

7.4 Publicações

O desenvolvimento do VSAC foi acompanhado por um processo de escrita de artigos científicos, refletindo-se em quatro publicações que apoiam os fundamentos teóricos, a proposta

técnica e os resultados experimentais obtidos ao longo da pesquisa. Essas publicações reforçam a relevância científica do trabalho e atestam sua contribuição para o avanço da área de compressão adaptativa de dados em sistemas de saúde baseados em IoT.

A primeira publicação, publicada no ERAD-RS (Escola Regional de Alto Desempenho da Região Sul): **Compressão de Dados MIoT para Melhorar Desempenho da Rede Mantendo QoS** [Andrade et al., 2022] propõe uma arquitetura que integra computação em nuvem e neblina com compressão de dados para dispositivos médicos de IoT (MIoT). O foco está na redução da latência e no controle do consumo de banda em ambientes distribuídos, contribuindo com fundamentos arquiteturais que inspiraram a estruturação do *framework* VSAC em camadas.

Em seguida, foi publicado no *journal Smart Health*, o artigo intitulado **IoT-Based Vital Sign Monitoring: A Literature Review** [Andrade et al., 2024], que apresenta uma revisão sistemática da literatura sobre monitoramento de sinais vitais com dispositivos IoT. O artigo analisou 78 estudos e apontou lacunas importantes, como a escassez de soluções para aquisição qualificada de dados e a necessidade de mecanismos de compressão integrados a arquiteturas Edge–Fog–Cloud. A revisão destacou dois temas emergentes: (i) a conexão entre dispositivos de IoT e aplicações para a coleta eficiente de dados críticos, e (ii) o uso de compressão para otimizar o consumo de banda e recursos computacionais em cidades inteligentes — ambos diretamente relacionados à concepção do VSAC.

Complementarmente, o artigo publicado no *journal FGCS - Future Generation Computer Systems*, intitulado **Blending Lossy and Lossless Data Compression Methods to Support Health Data Streaming in Smart Cities** [Andrade et al., 2025] descreve a proposta técnica do *framework* e seus primeiros resultados experimentais. O trabalho introduz a estratégia híbrida de compressão em dois estágios, detalhando como os algoritmos com perdas e sem perdas são integrados para garantir a eficiência na transmissão e a preservação da qualidade clínica dos dados.

Por fim, a publicação também no ERAD-RS: **Compressão Adaptativa para Streaming de Dados de Saúde com IoT em Cidades Inteligentes** [Andrade and da Rosa Righi, 2025] apresenta uma versão aplicada do VSAC, enfatizando sua capacidade de operar de forma adaptativa em arquiteturas distribuídas. A pesquisa detalha a geração de alertas em tempo real com base em notificações do sistema de monitoramento e demonstra, por meio de protótipos, a superioridade do *framework* em relação às abordagens convencionais, especialmente no equilíbrio entre compressão e fidelidade dos dados transmitidos.

7.5 Trabalhos Futuros

A título de trabalhos futuros, cabe considerar a evolução do protótipo, contemplando mais funcionalidades que o aproximem do modelo final proposto. Com esse novo protótipo, devem ser repetidos os ensaios já realizados e incluídos novos cenários de testes, considerando i) utilizar, no primeiro e no segundo estágio de compressão, algoritmos sem perda e ii) para

os dois estágios, algoritmos com perda. Aplicando as análises já desenvolvidas nesta tese e acrescentando novas análises, como tempo de restauração e qualidade do sinal restaurado.

Além desses novos cenários e análises, também serão incluídos novos *datasets*, além da frequência cardíaca, para verificar o desempenho do *framework* com a temperatura corporal, a SpO2 e a frequência respiratória. Esses novos *datasets* também devem contemplar tamanhos variados de arquivos e cadências para diferentes tipos de indivíduos monitorados, quanto ao sexo e à idade.

Está em desenvolvimento um novo artigo que utiliza os pacotes de dados processados de VSAC e aplica o método de score NEWS2 para análise e notificação do *framework*, aplicando as técnicas de priorização e cadência de leituras dos sinais vitais. Com isso, vislumbra-se a extensão do protótipo para suportar múltiplos tipos de sinais vitais em paralelo, com análise de desempenho multivariado e a incorporação de mecanismos de aprendizado adaptativo para calibração automática dos parâmetros de compressão, conforme o histórico de cada paciente. Essas evoluções podem ampliar significativamente a aplicabilidade do *framework* e consolidá-lo como uma solução estratégica para a gestão eficiente de dados em sistemas de saúde conectados.

Conforme verificado na análise dos resultados, também será necessário ajustar o processo de decisão do algoritmo de compressão sem perdas do segundo estágio e, com isso, melhorar o desempenho ao tratar arquivos prioritários.

REFERÊNCIAS

Vinicius Facco Rodrigues, Rodrigo da Rosa Righi, Cristiano André da Costa, Felipe André Zeiser, Bjoern Eskofier, Andreas Maier, and Daeyoung Kim. Digital health in smart cities: Rethinking the remote health monitoring architecture on combining edge, fog, and cloud. **Health and Technology**, pages 1–24, 2023.

Tanzima Azad, MA Hakim Newton, Jarrod Trevathan, and Abdul Sattar. Iot edge network interoperability. **Computer Communications**, 236:108125, 2025.

Shuchi Bhadula and Sachin Sharma. Personalized self adaptive internet-of-things enabled sustainable healthcare architecture for digital transformation. In **2023 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)**, pages 1–6. IEEE, 2023.

Ilenia Ficili, Maurizio Giacobbe, Giuseppe Tricomi, and Antonio Puliafito. From sensors to data intelligence: Leveraging iot, cloud, and edge computing with ai. **Sensors**, 25(6):1763, 2025.

Sanjeevikumar Padmanaban, Mostafa Azimi Nasab, Mohammad Ebrahim Shiri, Hamid Haj Seyyed Javadi, Morteza Azimi Nasab, Mohammad Zand, and Tina Samavat. The role of internet of things in smart homes. **Artificial Intelligence-based Smart Power Systems**, pages 259–271, 2023.

R Vasanth, M Paranthaman, and P Sivaprakash. Ml and iot coupled biomedical applications in healthcare: Smart growth and upcoming challenges. In **Internet of Things enabled Machine Learning for Biomedical Applications**, pages 1–22. CRC Press, 2025.

Prabhat Kumar and S Suresh. Deeptranshar: a novel clustering-based transfer learning approach for recognizing the cross-domain human activities using grus (gated recurrent units) networks. **Internet of Things**, 21:100681, 2023.

Wassila Lalouani, Mohamed Younis, Ian White-Gittens, Roland N Emokpae Jr, and Lloyd E Emokpae. Energy-efficient collection of wearable sensor data through predictive sampling. **Smart Health**, 21:100208, 2021.

DonHee Lee and Seong No Yoon. Application of artificial intelligence-based technologies in the healthcare industry: Opportunities and challenges. **International Journal of Environmental Research and Public Health**, 18(1):271, 2021.

Tao Lu, Wen Xia, Xiangyu Zou, and Qianbin Xia. Adaptively compressing iot data on the resource-constrained edge. In **HotEdge**, 2020.

Ali Kadhum Idrees and Marwa Saieed Khlief. Efficient compression technique for reducing transmitted eeg data without loss in iomt networks based on fog computing. **The Journal of Supercomputing**, pages 1–26, 2023.

Sujit Kumar Dwivedi, Jaya Yadav, Syed Anas Ansar, Mohd Waris Khan, Dhirendra Pandey, and Raees Ahmad Khan. A novel paradigm: Cloud-fog integrated iot approach. In **2022 3rd International Conference on Computation, Automation and Knowledge Management (ICCAKM)**, pages 1–5. IEEE, 2022.

Vaneet Aggarwal and Tian Lan. Modeling and optimization of latency in erasure-coded storage systems. **arXiv preprint arXiv:2005.10855**, 2020.

Charu Krishna, Dinesh Kumar, and Dharmender Singh Kushwaha. A comprehensive survey on pandemic patient monitoring system: Enabling technologies, opportunities, and research challenges. **Wireless Personal Communications**, pages 1–48, 2023.

Breno Renó, Edvard Martins Oliveira, and Adler Diniz Souza. A systematic literature review on trustworthiness for applications used in ehealth environments. **Journal of Multidisciplinary Healthcare**, pages 3393–3403, 2023.

Milan Kumar Verma, Sagar Choudhary, Hritik Gupta, et al. Iot-based healthcare monitoring system towards improving quality of life. **International Journal of Sciences and Innovation Engineering**, 2(5):1032–1042, 2025.

Vinicius F Rodrigues, Rodrigo R Righi, Cristiano A Costa, Rodolfo S Antunes, Rodrigo Bazo, Eduardo S Reis, Lucas A Seewald, Luis GS Junior, and Björn M Eskofier. Healthstack: providing an iot middleware for malleable qos service stacking for hospital 4.0 operating rooms. **IEEE Internet of Things Journal**, 9(19):18406–18430, 2022.

Aniello Castiglione, Muhammad Umer, Saima Sadiq, Mohammad S Obaidat, and Pandi Vijayakumar. The role of internet of things to control the outbreak of covid-19 pandemic. **IEEE Internet of Things Journal**, 8(21):16072–16082, 2021.

T Senthilkumar, B Manikandan, M Ramya Devi, and S Lokesh. Technologies enduring in internet of medical things (iomt) for smart healthcare system. **International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN**, pages 2456–3307, 2018.

Gabriel Souto Fischer, Rodrigo da Rosa Righi, Vinicius Facco Rodrigues, and Cristiano André da Costa. Use of internet of things with data prediction on healthcare environments: A survey. **International Journal of E-Health and Medical Communications (IJEHMC)**, 11(2):1–19, 2020.

Taher M Ghazal, Mohammad Kamrul Hasan, Muhammad Turki Alshurideh, Haitham M Alzoubi, Munir Ahmad, Syed Shehryar Akbar, Barween Al Kurdi, and Iman A Akour. Iot for smart cities: Machine learning approaches in smart healthcare—a review. **Future Internet**, 13 (8):218, 2021.

Vania Almeida and Ian T Nabney. Detecting dynamical changes in vital signs using switching kalman filter. In **2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)**, pages 2223–2226. IEEE, 2017.

Argyro Mavrogiorgou, Athanasios Kiourtis, and Dimosthenis Kyriazis. A pluggable iot middleware for integrating data of wearable medical devices. **Smart Health**, 26:100326, 2022.

Annika Hinze, Judy Bowen, and Jemma Lynette König. Wearable technology for hazardous remote environments: Smart shirt and rugged iot network for forestry worker health. **Smart Health**, 23:100225, 2022.

Manoharan Poongodi, Ashutosh Sharma, Mounir Hamdi, Ma Maode, and Naveen Chilamkurti. Smart healthcare in smart cities: wireless patient monitoring system using iot. **The Journal of Supercomputing**, pages 1–26, 2021.

Wen Qi Mok, Wenru Wang, and Sok Ying Liaw. Vital signs monitoring to detect patient deterioration: An integrative literature review. **International journal of nursing practice**, 21: 91–98, 2015.

Mostafa Haghi Kashani, Mona Madanipour, Mohammad Nikravan, Parvaneh Asghari, and Ebrahim Mahdipour. A systematic review of iot in healthcare: Applications, techniques, and trends. **Journal of Network and Computer Applications**, 192:103164, 2021.

CC Sobin. A survey on architecture, protocols and challenges in iot. **Wireless Personal Communications**, 112(3):1383–1429, 2020.

Sharu Bansal and Dilip Kumar. Iot ecosystem: A survey on devices, gateways, operating systems, middleware and communication. **International Journal of Wireless Information Networks**, 27(3):340–364, 2020.

Sreelakshmi Krishnamoorthy, Amit Dua, and Shashank Gupta. Role of emerging technologies in future iot-driven healthcare 4.0 technologies: A survey, current challenges and future directions. **Journal of Ambient Intelligence and Humanized Computing**, pages 1–47, 2021.

Hatem A Alharbi, Barzan A Yosuf, Mohammad Aldossary, and Jaber Almutairi. Energy and latency optimization in edge-fog-cloud computing for the internet of medical things. **Computer Systems Science & Engineering**, 47(1), 2023.

Shahnawaz Ahmad, Iman Shakeel, Shabana Mehfuz, and Javed Ahmad. Deep learning models for cloud, edge, fog, and iot computing paradigms: Survey, recent advances, and future directions. **Computer Science Review**, 49:100568, 2023.

Gabriele Proietti Mattia, Antonio Pietrabissa, and Roberto Beraldi. A load balancing algorithm for equalising latency across fog or edge computing nodes. **IEEE Transactions on Services Computing**, 2023.

Mostafa Hanoune and Mezui Eya'a Guy Lysmos. Data compression mechanisms in an intelligent e-health gateway for medical monitoring applications. **Procedia Computer Science**, 175:578–584, 2020.

S Kalaivani, C Tharini, K Saranya, and Kosireddy Priyanka. Design and implementation of hybrid compression algorithm for personal health care big data applications. **Wireless Personal Communications**, 113(1):599–615, 2020.

Cainã Passos, Carlos Pedroso, Agnaldo Batista, Michele Nogueira, and Aldri Santos. Grown: Local data compression in real-time to support energy efficiency in wban. In **2020 IEEE Latin-American Conference on Communications (LATINCOM)**, pages 1–6, 2020. doi: 10.1109/LATINCOM50620.2020.9282319.

Marwa Saieed Khlief and Ali Kadhum Idrees. Efficient eeg data compression technique for internet of health things networks. In **2022 IEEE World Conference on Applied Intelligence and Computing (AIC)**, pages 403–409. IEEE, 2022.

Ling Xiao, Yu Meng, and Kai Wu. Adaptive compressed classification for energy efficient activity recognition in wireless body sensor networks. In **2018 4th International Conference on Big Data Computing and Communications (BIGCOM)**, pages 41–45. IEEE, 2018.

Joseph Azar, Abdallah Makhoul, Mahmoud Barhamgi, and Raphaël Couturier. An energy efficient iot data compression approach for edge machine learning. **Future Generation Computer Systems**, 96:168–175, 2019.

Delaram Amiri, Janne Takalo-Mattila, Luca Bedogni, Marco Levorato, and Nikil Dutt. Sic-edge: Semantic iterative ecg compression for edge-assisted wearable systems. In **2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)**, pages 377–385. IEEE, 2022.

Subrata Kumar Das and Mohammad Zahidur Rahman. A compression technique for electronic health data through encoding. In **2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)**, pages 1–6. IEEE, 2021.

Ali Kadhum Idrees, Sara Kadhum Idrees, Raphael Couturier, and Tara Ali-Yahiya. An edge-fog computing-enabled lossless eeg data compression with epileptic seizure detection in iomt networks. **IEEE Internet of Things Journal**, 9(15):13327–13337, 2022.

Abdulbaghi Hossein and Amir Sheikh. A novel hybrid medical data compression using huffman coding and lzw in iot. **IETE Journal of Research**, pages 1–15, 2022.

Long Zhang, Bin Cao, Yun Li, Mugen Peng, and Gang Feng. A multi-stage stochastic programming-based offloading policy for fog enabled iot-ehealth. **IEEE Journal on Selected Areas in Communications**, 39(2):411–425, 2020.

Samodha Pallewatta, Vassilis Kostakos, and Rajkumar Buyya. Microservices-based iot application placement within heterogeneous and resource constrained fog computing environments. In **Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing**, pages 71–81, 2019.

Syed MH Rizvi, Tahir Syed, and Jalaluddin Qureshi. Real-time forecasting of petrol retail using dilated causal cnns. **Journal of Ambient Intelligence and Humanized Computing**, pages 1–12, 2022.

Tuan Nguyen Gia, L Qingqing, J Pena Queralta, Hannu Tenhunen, Zhuo Zou, and Tomi Westerlund. Lossless compression techniques in edge computing for mission-critical applications in the iot. In **2019 Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU)**, pages 1–2. IEEE, 2019a.

Alexandre Andrade, Arthur Tassinari Cabral, Bárbara Bellini, Vinicius Facco Rodrigues, Rodrigo da Rosa Righi, Cristiano André da Costa, and Jorge Luis Victória Barbosa. Iot-based vital sign monitoring: A literature review. **Smart Health**, page 100462, 2024.

S Sree Sanjanaa Bose and C Santhosh Kumar. Combining the multiple features for improving the performance of multi-parameter patient monitor. In **2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)**, pages 647–651. IEEE, 2019.

Karen J Mendoza Bautista, María G Ramírez Sotelo, and Agustín I Cabrera Llanos. A prioritization system for failures of a vital signs monitor. In **2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI)**, pages 1–4. IEEE, 2018.

Manju Lata Sahu, Mithilesh Atulkar, Mitul Kumar Ahirwal, and Afsar Ahamad. Vital sign monitoring system for healthcare through iot based personal service application. **Wireless Personal Communications**, 122(1):129–156, 2022a.

Manju Sahu, Mithilesh Atulkar, Mitul Ahirwal, and Afsar Ahamad. Vital sign monitoring system for healthcare through iot based personal service application. **Wireless Personal Communications**, 122:1–28, 01 2022b. doi: 10.1007/s11277-021-08892-4.

Seda Savaşcı Şen, Murtaza Cicioğlu, and Ali Çalhan. Iot-based gps assisted surveillance system with inter-wban geographic routing for pandemic situations. **Journal of Biomedical Informatics**, 116:103731, 2021.

David Nanchen. Resting heart rate: what is normal?, 2018.

Jouko Karjalainen and Matti Viitasalo. Fever and cardiac rhythm. **Archives of internal medicine**, 146(6):1169–1171, 1986.

Denisse Castaneda, Aibhlin Esparza, Mohammad Ghamari, Cinna Soltanpur, and Homer Nazeran. A review on wearable photoplethysmography sensors and their potential future applications in health care. **International journal of biosensors & bioelectronics**, 4(4):195, 2018.

Eric J Chow, Noah G Schwartz, Farrell A Tobolowsky, Rachael LT Zacks, Melinda Huntington-Frazier, Sujan C Reddy, and Agam K Rao. Symptom screening at illness onset of health care personnel with sars-cov-2 infection in king county, washington. **Jama**, 323(20): 2087–2089, 2020.

Yan Zhang, Yan Lu, Juan Tang, Yu Sun, Ze-Hua Zhao, Xiang-Dong Jian, and Xi-Mei Gao. Strategies for screening, occupational prevention, and management of covid-19 in outpatient clinics in shandong. **Frontiers in public health**, 10:911364, 2022.

John T Kanegaye, Jefferson M Jones, Jane C Burns, Sonia Jain, Xiaoying Sun, Susan Jimenez-Fernandez, Erika Berry, Joan M Pancheri, Preeti Jaggi, Octavio Ramilo, et al. Axillary, oral, and rectal routes of temperature measurement during treatment of acute kawasaki disease. **The Pediatric infectious disease journal**, 35(1):50, 2016.

Kapil S Meleveedu, John Miskovsky, Joseph Meharg, Abd Abdelrahman, Richa Tandon, Ashley E Moody, Priscilla Dasilva, Gabrielle Masse, Jason LaPorte, Abdul Saied Calvino, et al. Tocilizumab for severe covid-19 related illness–a community academic medical center experience. **Cytokine: X**, 2(4):100035, 2020.

Jagadeesh Kumar V and K Ashoka Reddy. Pulse oximetry for the measurement of oxygen saturation in arterial blood. **Studies in Skin Perfusion Dynamics: Photoplethysmography and its Applications in Medical Diagnostics**, pages 51–78, 2021.

Hugo RW Touw, Milou H Verheul, Pieter R Tuinman, Jeroen Smit, Deirdre Thöne, Patrick Schober, and Christa Boer. Photoplethysmography respiratory rate monitoring in patients receiving procedural sedation and analgesia for upper gastrointestinal endoscopy. **Journal of clinical monitoring and computing**, 31:747–754, 2017.

Roberto De Fazio, Marco Stabile, Massimo De Vittorio, Ramiro Velázquez, and Paolo Visconti. An overview of wearable piezoresistive and inertial sensors for respiration rate monitoring. **Electronics**, 10(17):2178, 2021.

Reena Tiwari, Ravindra Kumar, Sujata Malik, Tilak Raj, and Punit Kumar. Analysis of heart rate variability and implication of different factors on heart rate variability. **Current cardiology reviews**, 17(5):74–83, 2021.

Angela AT Schuurmans, Peter de Looff, Karin S Nijhof, Catarina Rosada, Ron HJ Scholte, Arne Popma, and Roy Otten. Validity of the empatica e4 wristband to measure heart rate variability (hrv) parameters: A comparison to electrocardiography (ecg). **Journal of medical systems**, 44:1–11, 2020.

Nicolas Bourdillon, Sasan Yazdani, Jean-Marc Vesin, Laurent Schmitt, and Grégoire P Millet. Rmssd is more sensitive to artifacts than frequency-domain parameters: implication in athletes' monitoring. **Journal of Sports Science & Medicine**, 21(2):260, 2022.

London RCoP. National early warning score (news): standardising the assessment of acute-illness severity in the nhs. **Report of working party. London: Royal College of Physicians**, 2012.

Matteo A Scrugli, Daniela Loi, Luigi Raffo, and Paolo Meloni. An adaptive cognitive sensor node for ecg monitoring in the internet of medical things. **IEEE Access**, 10:1688–1705, 2021.

Jawwad Latif, Pouyan Mehryar, Lei Hou, and Zulfiqur Ali. An efficient data compression algorithm for real-time monitoring applications in healthcare. In **2020 5th International Conference on Computer and Communication Systems (ICCCS)**, pages 71–75. IEEE, 2020.

Urmish Thakker, Igor Fedorov, Chu Zhou, Dibakar Gope, Matthew Mattina, Ganesh Dasika, and Jesse Beu. Compressing rnns to kilobyte budget for iot devices using kronecker products. **ACM Journal on Emerging Technologies in Computing Systems (JETC)**, 17(4):1–18, 2021.

Anil Pise, Byungun Yoon, and Saurabh Singh. Enabling ambient intelligence of things (aiot) healthcare system architectures. **Computer Communications**, 198:186–194, 2023.

Wu Wen, Umit Demirbaga, Amritpal Singh, Anish Jindal, Ranbir Singh Batth, Peiying Zhang, and Gagangeet Singh Aujla. Health monitoring and diagnosis for geo-distributed edge ecosystem in smart city. **IEEE Internet of Things Journal**, 2023.

Hafedh Ben Hassen, Nadia Ayari, and Belgacem Hamdi. A home hospitalization system based on the internet of things, fog computing and cloud computing. **Informatics in Medicine Unlocked**, 20:100368, 2020.

Sheikh Nooruddin, Md Milon Islam, and Falguni Ahmed Sharna. An iot based device-type invariant fall detection system. **Internet of Things**, 9:100130, 2020.

Tuan Nguyen Gia, Imed Ben Dhaou, Mai Ali, Amir M Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen. Energy efficient fog-assisted iot system for monitoring diabetic patients with cardiovascular disease. **Future Generation Computer Systems**, 93: 198–211, 2019b.

Sandeep K Sood, Amandeep Kaur, and Vaishali Sood. Energy efficient iot-fog based architectural paradigm for prevention of dengue fever infection. **Journal of Parallel and Distributed Computing**, 150:46–59, 2021.

Olivier Debauche, Saïd Mahmoudi, Pierre Manneback, and Abdessamad Assila. Fog iot for health: A new architecture for patients and elderly monitoring. **Procedia Computer Science**, 160:289–297, 2019.

Antonio Iyda Paganelli, Pedro Elkind Velmovitsky, Pedro Miranda, Adriano Branco, Paulo Alencar, Donald Cowan, Markus Endler, and Plinio Pelegrini Morita. A conceptual iot-based early-warning architecture for remote monitoring of covid-19 patients in wards and at home. **Internet of Things**, page 100399, 2021.

Zhiqin Qian, Yuchen Lin, Weiji Jing, Zhekai Ma, Hao Liu, Ruixue Yin, Zezhi Li, Zhuming Bi, and Wenjun Zhang. Development of a real time wearable fall detection system in the context of internet of things. **IEEE Internet of Things Journal**, 2022.

Kishore Medhi, Nurzaman Ahmed, and Md Iftekhar Hussain. Dew-based offline computing architecture for healthcare iot. **ICT Express**, 2021.

Tahani Aladwani. Scheduling iot healthcare tasks in fog computing based on their importance. **Procedia Computer Science**, 163:560–569, 2019.

Arman Anzanpour, Delaram Amiri, Iman Azimi, Marco Levorato, Nikil Dutt, Pasi Liljeberg, and Amir M Rahmani. Edge-assisted control for healthcare internet of things: A case study on ppg-based early warning score. **ACM Transactions on Internet of Things**, 2(1):1–21, 2020.

Chaomin Li. Information processing in internet of things using big data analytics. **Computer Communications**, 160:718–729, 2020.

Argyro Mavrogiorgou, Athanasios Kiourtis, and Dimosthenis Kyriazis. A plug 'n' play approach for dynamic data acquisition from heterogeneous iot medical devices of unknown nature. **Evolving Systems**, pages 1–21, 04 2019. doi: 10.1007/s12530-019-09286-5.

Nailah Saleh Alhassoun, Md Yusuf Sarwar Uddin, and Nalini Venkatasubramanian. Context-aware energy optimization for perpetual iot-based safe communities. **Sustainable Computing: Informatics and Systems**, 22:96–106, 2019.

Abdel Kader Chabi Sika Boni, Youssef Hablatou, Hassan Hassan, and Khalil Drira. Distributed deep reinforcement learning architecture for task offloading in autonomous iot systems. In **Proceedings of the 12th International Conference on the Internet of Things**, pages 112–118, 2022.

Shuvabrata Bandopadhaya, Rajiv Dey, and Ashok Suhag. Integrated healthcare monitoring solutions for soldier using the internet of things with distributed computing. **Sustainable Computing: Informatics and Systems**, 26:100378, 2020.

Ehsan Moghadas, Javad Rezazadeh, and Reza Farahbakhsh. An iot patient monitoring based on fog computing and data mining: Cardiac arrhythmia usecase. **Internet of Things**, 11: 100251, 2020.

Jayneel Vora, Sudeep Tanwar, Sudhanshu Tyagi, Neeraj Kumar, and Joel JPC Rodrigues. Faal: Fog computing-based patient monitoring system for ambient assisted living. In **2017 IEEE 19th international conference on e-health networking, applications and services (Healthcom)**, pages 1–6. IEEE, 2017.

Abinaya Inbamani, A Siva Sakthi, et al. Internet of things embedded in healthcare applications. In **2022 6th International Conference on Computing Methodologies and Communication (ICCMC)**, pages 382–389. IEEE, 2022.

Amir Masoud Rahmani, Zahra Babaei, and Alireza Souri. Event-driven iot architecture for data analysis of reliable healthcare application using complex event processing. **Cluster Computing**, pages 1–14, 06 2021. doi: 10.1007/s10586-020-03189-w.

Shahram Shirani. Data compression: The complete reference (by d. salomon; 2007)[book review]. **IEEE Signal Processing Magazine**, 25(2):147–149, 2008.

Sanku Kumar Roy and Ioanis Nikolaidis. Limited size lossy compression for wsns. In **2022 IEEE 47th Conference on Local Computer Networks (LCN)**, pages 359–362. IEEE, 2022.

Md Arif Khan, John W Pierre, Josh I Wold, Daniel J Trudnowski, and Matthew K Donnelly. Impacts of swinging door lossy compression of synchrophasor data. **International Journal of Electrical Power & Energy Systems**, 123:106182, 2020.

Yue Yin, Syed Muhammad Abubakar, Songyao Tan, Jiahua Shi, Peilin Yang, Wendi Yang, Hanjun Jiang, Zhihua Wang, Wen Jia, and U Seng-Pan. A 2.63 μw ecg processor with adaptive arrhythmia detection and data compression for implantable cardiac monitoring device. **IEEE Transactions on Biomedical Circuits and Systems**, 15(4):777–790, 2021.

Jorge Biolchini, Paula Gomes Mian, Ana Candida Cruz Natali, and Guilherme Horta Travassos. Systematic review in software engineering. **System engineering and computer science department COPPE/UFRJ, Technical Report ES**, 679(05):45, 2005.

Staffs Keele et al. Guidelines for performing systematic literature reviews in software engineering, 2007.

Yangyang Chang and Gerald E Sobelman. Lightweight lossy/lossless ecg compression for medical iot systems. **IEEE Internet of Things Journal**, 11(7):12450–12458, 2023.

Mahdieh Hajiloo Vakil and Zahra Shirmohammadi. Lprlc: Linear predictive run length coding to improve energy consumption of wireless body area networks. **Results in Engineering**, page 105622, 2025.

Sara Kadhum Idrees, Joseph Azar, Raphaël Couturier, Ali Kadhum Idrees, and Franck Gechter. Sz4iot: an adaptive lightweight lossy compression algorithm for diverse iot devices and data types. **The Journal of Supercomputing**, 81(2):392, 2025.

Sudeshna Baliarsingh and Mihir Narayan Mohanty. Electrocardiogram compression using adaptive decomposition techniques for smart healthcare system. **Cureus Journals**, 2(1), 2025.

Sara Kadhum Idrees and Ali Kadhum Idrees. New fog computing enabled lossless eeg data compression scheme in iot networks. **Journal of Ambient Intelligence and Humanized Computing**, pages 1–14, 2022.

Gary B Smith, Oliver C Redfern, Marco AF Pimentel, Stephen Gerry, Gary S Collins, James Malycha, David Prytherch, Paul E Schmidt, and Peter J Watkinson. The national early warning score 2 (news2). **Clinical Medicine**, 19(3):260, 2019.

Juan David Arias Correa, Alex Sandro Roschildt Pinto, Carlos Montez, and Erico Meneses Leao. Swinging door trending compression algorithm for iot environments. In **Anais Estendidos do IX Simpósio Brasileiro de Engenharia de Sistemas Computacionais**, pages 143–148. SBC, 2019.

Gajendra Sharma. Analysis of huffman coding and lempel–ziv–welch (lzw) coding as data compression techniques. **International Journal of Scientific Research in Computer Science and Engineering**, 8(1):37–44, 2020.

Iwona Cygankiewicz and Wojciech Zareba. Heart rate variability. **Handbook of clinical neurology**, 117:379–393, 2013.

Stefan Sammito and Irina Böckelmann. Reference values for time-and frequency-domain heart rate variability measures. **Heart Rhythm**, 13(6):1309–1316, 2016.

Ashraf Mohamed Ali Hassan and Saeed Mohsen. Compression of electrocardiogram signals using compressive sensing technique based on curvelet transform toward medical applications. **Multimedia Tools and Applications**, pages 1–17, 2024.

Supriya Rajankar, Omprakash Rajankar, Sanjay Talbar, and Vrushali Raut. Signal adaptive threshold for ecg signal compression using false discovery rate approach. **Circuits, Systems, and Signal Processing**, pages 1–25, 2024.

Tahereh Moein, Mohammad Ali Keramati, and Hossein Moinzad. Optimizing the performance reliability of diagnostic equipment and wearable sensors and medical devices in iomt. **International Journal of Nonlinear Analysis and Applications**, 2024.

Douglas C Montgomery. **Design and analysis of experiments**. John wiley & sons, 2017.

Jay L Devore. Probability and statistics for engineering and the sciences . boston, ma: Cengage learning. Technical report, ISBN 978-0-538-73352-6, 2011.

Raj Jain. **The art of computer systems performance analysis**. john wiley & sons, 1990.

Rubén Muñiz, Fernando Nuño, Juan Díaz, María González, Miguel J. Prieto, and Óliver Menéndez. Real-time monitoring solution with vibration analysis for industry 4.0 ventilation systems. **The Journal of Supercomputing**, 79(6):6203–6227, 2023.

Ioan Szabo, Andrei Alexandru Tulbure, Dorin Fleseriu, and Adrian Alexandru Tulbure. Vibration and temperature sensor network solutions: Case study for industry 4.0. In **2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA)**, pages 678–682. IEEE, 2021.

Alexandre Andrade, Arthur Cabral, Barbara Bellini, Rodrigo da Rosa Righi, and Vinicius Facco Rodrigues. Compressao de dados miot para melhorar desempenho da rede mantendo qos. In **Escola Regional de Alto Desempenho da Região Sul (ERAD-RS)**, pages 57–58. SBC, 2022.

Alexandre Andrade, Cristiano André da Costa, Alex Roehrs, Debora Muchaluat-Saade, and Rodrigo da Rosa Righi. Blending lossy and lossless data compression methods to support health data streaming in smart cities. **Future Generation Computer Systems**, page 107748, 2025.

Alexandre Andrade and Rodrigo da Rosa Righi. Compressão adaptativa para streaming de dados de saúde com iot em cidades inteligentes. In **Escola Regional de Alto Desempenho da Região Sul (ERAD-RS)**, pages 137–138. SBC, 2025.