

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
FACULDADE DE COMPUTAÇÃO
ENGENHARIA DE COMPUTAÇÃO

EXPLORANDO A VIABILIDADE DA
COMPUTAÇÃO QUÂNTICA EM DADOS
BIOMÉDICOS: UM *BENCHMARKING*
ENTRE PROCESSAMENTO DIGITAL
CLÁSSICO E ALGORITMOS QUÂNTICOS

MATHEUS NANTES FONSECA DUARTE

CAMPO GRANDE – MS

2026

MATHEUS NANTES FONSECA DUARTE

EXPLORANDO A VIABILIDADE DA
COMPUTAÇÃO QUÂNTICA EM DADOS
BIOMÉDICOS: UM *BENCHMARKING*
ENTRE PROCESSAMENTO DIGITAL
CLÁSSICO E ALGORITMOS QUÂNTICOS

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação da Universidade Federal de Mato Grosso do Sul, como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Milton Ernesto Romero Romero

CAMPO GRANDE – MS

2026

MATHEUS NANTES FONSECA DUARTE

**EXPLORANDO A VIABILIDADE DA COMPUTAÇÃO
QUÂNTICA EM DADOS BIOMÉDICOS: UM
BENCHMARKING ENTRE PROCESSAMENTO DIGITAL
CLÁSSICO E ALGORITMOS QUÂNTICOS**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação da Universidade Federal de Mato Grosso do Sul, como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação.

Aprovado em 10 de junho de 2026.

BANCA EXAMINADORA

Prof. Dr. Milton Ernesto Romero Romero – Orientador
Universidade Federal de Mato Grosso do Sul

Profa. Dra. Bruna da Silva Sousa – Membro da Banca
Universidade de Brasília

Profa. Dra. Vera Regina Fernandes da Silva Maraes – Membro da Banca
Universidade de Brasília

CAMPO GRANDE – MS

2026

*À minha família, Julio, Seli e Lucas,
pelo apoio, pela paciência e pelo exemplo que sustentaram este caminho.*

*À Giulia,
por ser presença, abrigo e sentido nos dias em que o tempo exigiu mais do que parecia
possível.*

AGRADECIMENTOS

Agradeço à minha família, Julio, Seli e Lucas, pelo apoio, pela paciência e pelo exemplo que sustentaram este caminho, inclusive nos períodos em que ele exigiu mais tempo e maturidade. Muito do que se conclui neste trabalho nasce também da segurança, do afeto e da força que sempre encontrei em casa.

À Giulia, meu amor, pela presença constante, pelo cuidado e por permanecer ao meu lado nos dias em que a caminhada exigiu mais do que eu parecia ter. Sua presença deu leveza aos períodos difíceis e sentido às conquistas que vieram depois deles.

Ao meu orientador, Prof. Dr. Milton Ernesto Romero Romero, pela orientação, pela confiança e por ter reacendido uma curiosidade que eu julgava adormecida. A forma como conduziu as aulas e as discussões em sala teve papel decisivo não apenas na construção deste trabalho, mas também na maneira como voltei a me relacionar com o conhecimento.

Estendo meus agradecimentos à Profa. Dra. Bruna da Silva Sousa e à Profa. Dra. Vera Regina Fernandes da Silva Maraes por aceitarem o convite para compor a banca examinadora e pelo tempo dedicado à leitura e avaliação deste trabalho.

Agradeço também ao Prof. Dr. Victor Leonardo Yoshimura, à Profa. Dra. Bianca de Almeida Dantas e ao Prof. Dr. Renan Albuquerque Marks, professores que, em momentos diferentes da graduação, deixaram marcas importantes na minha formação. Seja pelo rigor, pelo acolhimento, pela exigência ou pela forma de conduzir o aprendizado, cada um contribuiu para que eu compreendesse melhor não apenas os conteúdos da Engenharia de Computação, mas também a responsabilidade envolvida em pensar, construir e resolver problemas com seriedade.

Aos professores da Faculdade de Computação da Universidade Federal de Mato Grosso do Sul, pelas contribuições ao longo da graduação e pela formação técnica que tornou possível o desenvolvimento deste trabalho.

Aos colegas de curso, pelas conversas, trocas de conhecimento e momentos compartilhados durante a graduação.

Por fim, agradeço a todos que, direta ou indiretamente, fizeram parte desta trajetória e contribuíram para que este ciclo pudesse ser concluído.

*“Há caminhos que só se revelam quando aceitamos atravessar
o limite entre aquilo que sabemos medir
e aquilo que ainda precisamos compreender.”*

EXPLORANDO A VIABILIDADE DA COMPUTAÇÃO QUÂNTICA EM DADOS BIOMÉDICOS: UM *BENCHMARKING* ENTRE PROCESSAMENTO DIGITAL CLÁSSICO E ALGORITMOS QUÂNTICOS

Matheus Nantes Fonseca Duarte

Faculdade de Computação (FACOM) – Universidade Federal de Mato Grosso do Sul (UFMS)

matheus.duarte@ufms.br

Resumo. A análise computacional de dados biomédicos envolve sinais de natureza ruidosa, não estacionária e frequentemente não linear. Entre esses sinais, a Variabilidade da Frequência Cardíaca, analisada a partir de séries de intervalos RR derivados do eletrocardiograma, fornece informações relevantes sobre a regulação autonômica da atividade cardíaca e pode auxiliar na distinção entre padrões fisiológicos associados a indivíduos normotensos e indivíduos com hipertensão arterial em intervenção farmacológica. Este trabalho investiga a viabilidade de algoritmos de *Quantum Machine Learning* aplicados a esse cenário, por meio de um *benchmarking* entre processamento digital clássico e modelos quânticos híbridos simulados. A metodologia contempla a extração de descritores temporais, entrópicos, fractais, geométricos e espectrais, a construção de uma linha de base clássica com modelos de *ensemble* e o desenvolvimento de arquiteturas quânticas baseadas em *Quantum Support Vector Classifier* e *Quantum Feature Maps*. O estudo compara marcos consolidados de experimentação, resultantes de sucessivos testes de engenharia, incluindo variações de codificação angular, seleção de características, profundidade de circuito, topologia de emaranhamento e *Data Re-uploading*. Os resultados indicam que o modelo clássico de referência alcançou média global de 83,73% e pico de 91,67%, enquanto a configuração quântica híbrida mais estável alcançou média de 80,17% e pico de 90,00% em simulação ideal, utilizando apenas quatro descritores principais redistribuídos em oito *qubits*. Conclui-se que a abordagem quântica não substitui, no estado atual, os métodos clássicos consolidados, mas demonstra competitividade experimental e eficiência dimensional em um problema biomédico realista baseado em sinais cardíacos. Os resultados devem ser interpretados como evidência exploratória em uma base de escala limitada, e não como validação clínica definitiva. A pesquisa também evidencia limitações importantes da era de computação quântica de escala intermediária e ruidosa, como custo de simulação, restrições de codificação, sensibilidade a ruído e dependência de pré-processamento clássico rigoroso.

Palavras-chave: Computação Quântica. Dados Biomédicos. Variabilidade da Frequência Cardíaca. Processamento Digital de Sinais. Aprendizado de Máquina Quântico. QSVC.

EXPLORING THE FEASIBILITY OF QUANTUM COMPUTING IN BIOMEDICAL DATA: A *BENCHMARK* BETWEEN CLASSICAL DIGITAL PROCESSING AND QUANTUM ALGORITHMS

Matheus Nantes Fonseca Duarte

Faculty of Computing (FACOM) – Federal University of Mato Grosso do Sul (UFMS)
matheus.duarte@ufms.br

Abstract. The computational analysis of biomedical data involves noisy, non-stationary, and often nonlinear signals. Among these signals, Heart Rate Variability, obtained from electrocardiogram-derived RR interval time series, provides relevant information about autonomic regulation of cardiac activity and may support the distinction between physiological patterns associated with normotensive individuals and individuals with hypertension under pharmacological intervention. This work investigates the feasibility of *Quantum Machine Learning* algorithms in this context through a *benchmark* between classical digital processing and simulated hybrid quantum models. The methodology includes the extraction of temporal, entropic, fractal, geometric, and spectral descriptors, the construction of a classical baseline using *ensemble* models, and the development of quantum architectures based on *Quantum Support Vector Classifier* and *Quantum Feature Maps*. The study compares consolidated experimental milestones obtained after successive engineering tests, including variations in angular encoding, feature selection, circuit depth, entanglement topology, and *Data Re-uploading*. The results indicate that the classical reference model achieved a global mean accuracy of 83.73% and a peak accuracy of 91.67%, while the most stable hybrid quantum configuration achieved a mean accuracy of 80.17% and a peak accuracy of 90.00% in ideal simulation, using only four main descriptors redistributed across eight *qubits*. The findings suggest that the quantum approach does not currently replace mature classical methods, but it shows experimental competitiveness and dimensional efficiency in a realistic biomedical problem based on cardiac signals. The results should be interpreted as exploratory evidence obtained from a limited-scale dataset, rather than as definitive clinical validation. The research also highlights relevant limitations of noisy intermediate-scale quantum computing, such as simulation cost, encoding restrictions, noise sensitivity, and dependence on rigorous classical preprocessing.

Keywords: Quantum Computing. Biomedical Data. Heart Rate Variability. Digital Signal Processing. Quantum Machine Learning. QSVC.

LISTA DE ILUSTRAÇÕES

1.1	Visão geral da proposta experimental: séries RR derivadas do ECG são convertidas em descritores da atividade cardíaca; em seguida, uma linha de base clássica robusta é comparada a uma arquitetura quântica híbrida simulada, com foco em desempenho, estabilidade e eficiência dimensional.	3
2.1	Exemplo de medição do intervalo RR e série RR utilizada no estudo	10
2.2	Síntese conceitual das famílias de descritores utilizadas para representar séries de intervalos RR, incluindo medidas temporais, entrópicas, energéticas, fractais, geométricas e espectrais.	12
2.3	Exemplo de gráfico de Poincaré construído a partir de uma série RR, no qual cada ponto representa a relação entre RR_n e RR_{n+1}	15
2.4	Representação esquemática de estados de um <i>qubit</i> na esfera de Bloch, ilustrando como diferentes amplitudes e fases correspondem a diferentes orientações do vetor de estado.	18
2.5	Exemplo esquemático de circuito quântico ilustrando preparação em superposição, codificação angular por rotações e criação de correlações entre <i>qubits</i> . A figura tem finalidade conceitual e não representa, necessariamente, a decomposição exata do circuito final utilizado nos experimentos.	20
2.6	Fluxo conceitual de um modelo quântico híbrido baseado em QSVC: descritores clássicos são escalonados, codificados como parâmetros de um circuito quântico, transformados por um <i>Quantum Feature Map</i> e utilizados para compor uma matriz de kernel, posteriormente analisada por um classificador SVM clássico.	22
3.1	Organização experimental da base de dados utilizada neste trabalho. As séries RR derivadas do ECG foram separadas em duas classes, transformadas em vetores de características e utilizadas nos protocolos clássico e quântico híbrido.	26
3.2	Famílias de descritores utilizadas para representar diferentes aspectos das séries RR. Cada grupo de características observa o sinal por uma perspectiva complementar: tempo, irregularidade, energia local, complexidade de escala, geometria entre intervalos consecutivos e comportamento espectral.	28

3.3	Representação conceitual da entropia espectral acelerada. A Transformada de Fourier tradicional observa a distribuição de frequências, enquanto a modificação por aceleração busca sensibilizar a fase a mudanças locais da dinâmica da série RR.	30
3.4	Representação esquemática da relevância relativa de variáveis utilizadas na seleção de entrada para o modelo quântico. A seleção priorizou descritores compactos, com alta densidade informacional e significado compatível com a dinâmica das séries RR.	31
3.5	Fluxo metodológico do <i>pipeline</i> clássico de referência. As etapas aprendidas, como escalonamento e balanceamento sintético, são ajustadas exclusivamente no conjunto de treino, enquanto o conjunto de teste permanece composto apenas por amostras originais.	33
3.6	Fluxo metodológico do modelo quântico híbrido. Os descritores clássicos são selecionados, escalonados, codificados como ângulos, processados por um <i>Quantum Feature Map</i> e convertidos em uma matriz de kernel utilizada por um SVM clássico.	34
3.7	Esquema da validação cega por amostras retidas. Uma amostra normotensa e uma amostra da classe hipertensa são removidas antes dos ajustes aprendidos; o treinamento ocorre sem acesso a essas amostras, que são avaliadas apenas ao final como observações inéditas.	37
4.1	Comparação do <i>recall</i> médio por classe entre o <i>benchmark</i> clássico e o modelo quântico híbrido final. A figura evidencia a maior facilidade dos modelos em reconhecer a classe hipertensa em intervenção farmacológica, bem como a maior dificuldade relativa de identificação da classe normotensa.	40
4.2	Evolução dos principais marcos quânticos híbridos avaliados ao longo do desenvolvimento experimental. A queda observada no marco Q-V10 ilustra que maior profundidade de circuito não implicou melhor separabilidade para as séries RR analisadas.	44
4.3	Matriz de confusão do melhor caso observado para o modelo quântico híbrido Q-V14. A matriz evidencia maior acerto na classe hipertensa em intervenção farmacológica e desempenho inferior, embora ainda relevante, na classe normotensa.	46
4.4	Comparação entre o desempenho do <i>benchmark</i> clássico e do modelo quântico híbrido final. O gráfico destaca a proximidade entre os resultados de pico e a diferença moderada entre as acurácias médias, sustentando a leitura de competitividade experimental em simulação ideal.	48

5.1	Síntese interpretativa dos resultados do trabalho. O modelo clássico estabeleceu uma referência robusta de desempenho, enquanto o modelo quântico híbrido Q-V14 demonstrou competitividade experimental com entrada dimensionalmente mais compacta. A interpretação permanece limitada ao contexto de simulação ideal e base de escala reduzida.	52
-----	--	----

LISTA DE TABELAS

3.1	Síntese metodológica da evolução do <i>pipeline</i> clássico.	32
3.2	Síntese metodológica dos principais marcos quânticos avaliados.	36
4.1	Resultado consolidado do <i>benchmark</i> clássico de referência.	40
4.2	Evolução resumida dos principais marcos quânticos híbridos.	44
4.3	Resultado consolidado do modelo quântico híbrido Q-V14.	45
4.4	Comparação entre o <i>benchmark</i> clássico e o modelo quântico final.	47
A.1	Síntese de reprodutibilidade dos experimentos finais.	57

LISTA DE ABREVIATURAS E SIGLAS

ECG	Eletrocardiograma
RR	Intervalo entre picos R consecutivos do ECG
VFC	Variabilidade da Frequência Cardíaca
SNA	Sistema Nervoso Autônomo
ML	Aprendizado de Máquina (<i>Machine Learning</i>)
QML	Aprendizado de Máquina Quântico (<i>Quantum Machine Learning</i>)
QSVC	Classificador de Vetores de Suporte Quântico (<i>Quantum Support Vector Classifier</i>)
SVM	Máquina de Vetores de Suporte (<i>Support Vector Machine</i>)
NISQ	Computação quântica de escala intermediária e ruidosa (<i>Noisy Intermediate-Scale Quantum</i>)
QPU	Unidade de Processamento Quântico (<i>Quantum Processing Unit</i>)
PCA	Análise de Componentes Principais (<i>Principal Component Analysis</i>)
SMOTE	Técnica de superamostragem sintética da classe minoritária (<i>Synthetic Minority Oversampling Technique</i>)
TKEO	Operador de Energia de Teager-Kaiser (<i>Teager-Kaiser Energy Operator</i>)
SampEn	Entropia de Amostra (<i>Sample Entropy</i>)
PE	Entropia de Permutação (<i>Permutation Entropy</i>)
MSE	Entropia Multiescala (<i>Multiscale Entropy</i>)
DFA	Análise de Flutuação Destendenciada (<i>Detrended Fluctuation Analysis</i>)
DWT	Transformada Wavelet Discreta (<i>Discrete Wavelet Transform</i>)
WPD	Decomposição por Pacotes Wavelet (<i>Wavelet Packet Decomposition</i>)

SUMÁRIO

LISTA DE ILUSTRAÇÕES	8
LISTA DE TABELAS	9
LISTA DE ABREVIATURAS E SIGLAS	10
1 Introdução	1
1.1 Contextualização	1
1.2 Definição do Problema	3
1.3 Objetivos	5
1.3.1 Objetivo Geral	5
1.3.2 Objetivos Específicos	5
1.4 Justificativa	6
1.5 Estrutura do Trabalho	7
2 Fundamentação Teórica	9
2.1 Sinais Biomédicos e Dinâmica Não-Linear	9
2.1.1 Características Lineares e Limitações Estatísticas	11
2.1.2 Entropia de Amostra	12
2.1.3 Entropia de Permutação	13
2.1.4 Entropia Multiescala e Complexidade Temporal	13
2.1.5 Energia de Teager-Kaiser e Fragmentação do Sinal	14
2.1.6 Descritores Fractais, Geométricos e de Poincaré	14
2.1.7 Síntese para a Modelagem Computacional	14
2.2 <i>Machine Learning</i> Clássico e Prevenção de <i>Data Leakage</i>	15
2.2.1 Modelos Baseados em Árvores e <i>Ensemble Learning</i>	16
2.2.2 Escalonamento, Balanceamento e Seleção de Características	16
2.2.3 <i>Data Leakage</i> : Definição e Riscos	17
2.2.4 Validação Estocástica e Interpretação de Desempenho	17
2.3 Fundamentos de Computação Quântica	18
2.3.1 Espaço de Hilbert e Crescimento Dimensional	18
2.3.2 Portas Quânticas e Circuitos	19

2.3.3	Codificação de Dados Clássicos	19
2.3.4	Simulação Clássica de Circuitos Quânticos	20
2.4	<i>Quantum Machine Learning</i> , QSVC e Era NISQ	21
2.4.1	Kernels e Classificadores de Vetores de Suporte	21
2.4.2	<i>Quantum Feature Maps</i>	21
2.4.3	Era NISQ	22
2.4.4	Platôs Estéreis e Profundidade de Circuito	23
2.4.5	<i>Data Re-uploading</i> e Eficiência Dimensional	23
2.4.6	Interpretação Cuidadosa de Resultados em QML	23
3	Metodologia e Desenvolvimento Experimental	25
3.1	Aquisição, Organização e Preparação dos Dados	25
3.2	Engenharia de Características Clássicas	27
3.2.1	Descritores Temporais e Estatísticos	28
3.2.2	Descritores Entrópicos e Não-Lineares	29
3.2.3	Energia, Fragmentação e Transformações Derivadas	29
3.2.4	Seleção de Características para o Modelo Quântico	31
3.3	Construção do <i>Pipeline</i> Clássico de Referência	31
3.4	Arquitetura Quântica Híbrida	33
3.4.1	Codificação Angular	33
3.4.2	<i>Feature Maps</i> e Topologias de Emaranhamento	34
3.4.3	Redução por PCA, Seleção Direta e Redundância Lateral	35
3.4.4	Controle Clássico do Hiperplano	35
3.4.5	Síntese dos Marcos Quânticos Consolidados	36
3.5	Métricas de Avaliação	36
4	Resultados, <i>Benchmarking</i> e Discussão	39
4.1	Desempenho do <i>Benchmark</i> Clássico	39
4.2	Evolução dos Modelos Quânticos	41
4.2.1	Compressão Linear e Perda de Informação	41
4.2.2	Ajuste Angular e Geometria da Esfera de Bloch	42
4.2.3	Expressividade versus Estabilidade	42
4.2.4	Expansão Lateral e Redundância Inspirada em <i>Data Re-uploading</i>	42
4.2.5	Profundidade e Colapso de Desempenho	43
4.2.6	Síntese Numérica dos Principais Marcos	43
4.3	Desempenho do Modelo Quântico Híbrido Final	45
4.4	Discussão: Eficiência Dimensional e Limites do <i>Benchmarking</i>	47
5	Conclusões e Trabalhos Futuros	50
5.1	Conclusões Gerais	50

5.2 Limitações e Trabalhos Futuros	52
Referências	54
A Parâmetros de reprodutibilidade dos experimentos	56
B Código-fonte do benchmark clássico final	58
C Código-fonte do modelo quântico híbrido final	68
D Código-fonte da validação cega por amostras retidas	73

Capítulo 1

Introdução

1.1 Contextualização

O processamento de sinais biomédicos representa um dos campos mais desafiadores da Engenharia de Computação contemporânea, pois envolve a análise de fenômenos fisiológicos que raramente se comportam de maneira linear, estacionária ou completamente previsível. Diferentemente de sinais artificiais produzidos sob condições controladas, séries temporais biológicas carregam ruído instrumental, variações individuais, influência ambiental, estados transitórios do organismo e mecanismos regulatórios que se manifestam simultaneamente em diferentes escalas de tempo. Essa combinação torna a extração de conhecimento um problema de engenharia complexo, no qual a simples observação visual do sinal ou a aplicação de estatísticas descritivas elementares não é suficiente para caracterizar estados fisiológicos distintos.

Entre os sinais biomédicos de maior interesse neste estudo está a Variabilidade da Frequência Cardíaca (VFC), analisada a partir dos intervalos RR, isto é, os intervalos temporais entre picos R consecutivos em um eletrocardiograma (ECG). A VFC é um biomarcador indireto da regulação autonômica da atividade cardíaca, pois reflete a interação dinâmica entre os ramos simpático e parassimpático do Sistema Nervoso Autônomo. Em condições fisiológicas saudáveis, o coração não opera como um metrônomo perfeito; ao contrário, pequenas variações entre batimentos sucessivos indicam capacidade de adaptação, flexibilidade regulatória e resposta eficiente a estímulos internos e externos. Em condições patológicas, como a hipertensão arterial, essa dinâmica tende a se tornar menos complexa, mais rígida e mais previsível, refletindo alterações na modulação autonômica cardíaca (1, 2).

Neste estudo, a classe hipertensa deve ser interpretada de modo específico: trata-se de indivíduos com hipertensão arterial em intervenção farmacológica. Essa delimitação é relevante porque a tarefa computacional não consiste em separar indivíduos saudáveis de pacientes sem tratamento, mas em verificar se ainda permanecem padrões dinâmicos

detectáveis mesmo sob intervenção farmacológica. Em termos conceituais, caso existisse uma intervenção capaz de normalizar integralmente todos os aspectos da dinâmica da atividade cardíaca observável nos intervalos RR, a distinção entre normotensos e indivíduos da classe hipertensa tenderia a desaparecer no espaço de atributos. Assim, qualquer separabilidade observada deve ser entendida como evidência experimental de padrões residuais ou não completamente normalizados no sinal, e não como afirmação clínica de diagnóstico automático.

A natureza desses sinais exige métodos capazes de capturar padrões que não aparecem de forma explícita no domínio do tempo. Por esse motivo, a análise moderna da VFC combina métricas tradicionais, como média e desvio dos intervalos RR, com descritores não-lineares, entrópicos e fractais. Métricas como a Entropia de Amostra (*Sample Entropy*) (3) e a Entropia de Permutação (*Permutation Entropy*) (4) buscam quantificar a irregularidade e a complexidade estrutural do sinal, permitindo avaliar se a série temporal preserva diversidade dinâmica ou se apresenta padrões repetitivos associados à perda de adaptabilidade fisiológica. Além dessas métricas, técnicas de energia, fragmentação, análise fractal e representações geométricas em espaços de fase ampliam a capacidade de descrição da atividade cardíaca (5, 6, 7, 8).

Na prática computacional clássica, essas características são extraídas e utilizadas como entrada para algoritmos de *Machine Learning*. Modelos baseados em árvores de decisão, florestas aleatórias, *Extra Trees*, *Gradient Boosting* e comitês de estimadores são frequentemente empregados por sua capacidade de lidar com relações não-lineares e interações complexas entre variáveis. O desempenho desses métodos, contudo, tende a depender de um processo intensivo de *feature engineering*. Quanto mais complexa a fisiologia investigada, maior costuma ser a quantidade de descritores extraídos para que o modelo encontre fronteiras de decisão adequadas. Essa abordagem é eficaz, mas pode se tornar custosa, pouco interpretável e sensível a vieses metodológicos.

Em paralelo ao avanço do processamento digital clássico, a computação quântica tem emergido como um paradigma alternativo de representação e manipulação de informação. Enquanto a computação clássica opera sobre bits definidos em estados discretos, a computação quântica utiliza *qubits*, que podem ser descritos por vetores de estado em espaços complexos de alta dimensão. A combinação de superposição, interferência e emaranhamento permite representar dados de maneira geometricamente distinta da abordagem clássica. O campo do *Quantum Machine Learning* (QML) investiga justamente se essas propriedades podem ser usadas para construir classificadores, kernels e modelos híbridos capazes de explorar estruturas latentes nos dados (9, 10).

Este trabalho situa-se nessa interseção entre sinais biomédicos, aprendizado de máquina clássico e aprendizado de máquina quântico. O problema experimental escolhido consiste em classificar padrões da atividade elétrica cardíaca, derivados de séries de intervalos RR, associados a indivíduos normotensos e indivíduos com hipertensão arterial

em intervenção farmacológica. A pesquisa parte de uma premissa central: para avaliar honestamente a viabilidade da computação quântica em dados biomédicos, não basta comparar um circuito quântico contra um modelo clássico simples. É necessário construir uma linha de base clássica forte, auditável e competitiva, e somente então verificar se uma arquitetura quântica híbrida consegue alcançar desempenho próximo usando uma representação dimensionalmente mais compacta.

Visão geral do benchmark do trabalho

Comparação entre processamento clássico e computação quântica híbrida

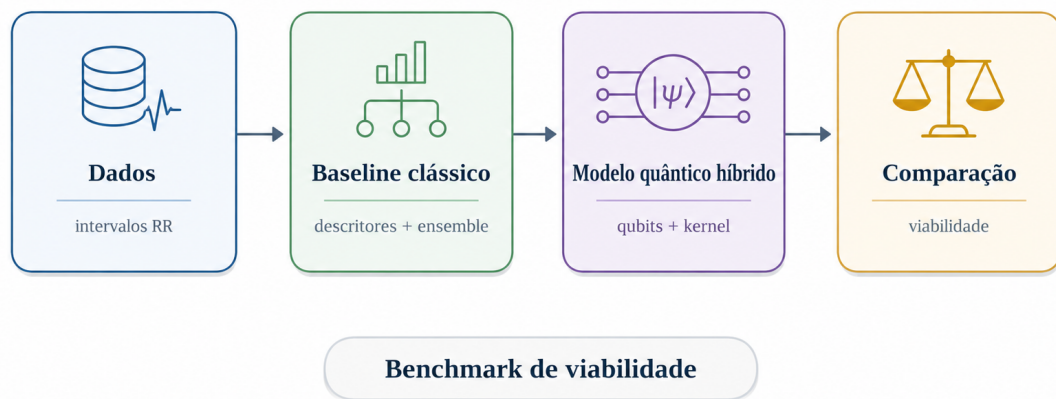


Figura 1.1: Visão geral da proposta experimental: séries RR derivadas do ECG são convertidas em descritores da atividade cardíaca; em seguida, uma linha de base clássica robusta é comparada a uma arquitetura quântica híbrida simulada, com foco em desempenho, estabilidade e eficiência dimensional.

Fonte: elaboração própria.

Assim, o foco deste trabalho não é afirmar uma vantagem quântica ampla ou definitiva. O objetivo é mais específico e tecnicamente mais defensável: investigar, em ambiente simulado, se circuitos quânticos rasos e compatíveis com as restrições da era NISQ conseguem atuar como mecanismos de mapeamento geométrico eficientes para dados biomédicos não-lineares derivados da atividade elétrica cardíaca. O valor científico da pesquisa está tanto nos resultados positivos quanto nos fracassos controlados ao longo das versões experimentais, pois cada iteração revela limitações práticas do processo de codificação, do número de *qubits*, da profundidade dos circuitos e do custo de simulação.

1.2 Definição do Problema

O problema investigado neste trabalho pode ser compreendido a partir de duas dimensões complementares: a complexidade dos dados biomédicos e a eficiência da re-

apresentação computacional. Séries temporais de intervalos RR, derivadas da atividade elétrica cardíaca registrada no ECG, não apresentam uma separação trivial entre classes clínicas. A diferença entre um sinal associado a um indivíduo normotenso e um sinal associado a um indivíduo com hipertensão arterial em intervenção farmacológica pode não estar localizada em um único atributo isolado, mas distribuída em padrões de irregularidade, fragmentação, recorrência e organização temporal. Em outras palavras, o sinal contém informação fisiológica relevante, mas essa informação encontra-se espalhada em uma estrutura não-linear.

Modelos clássicos de aprendizado de máquina lidam com esse desafio por meio de ampla engenharia de características. A estratégia consiste em extrair dezenas de descritores do sinal, abrangendo domínio do tempo, frequência, entropia, geometria, energia e fractalidade. Em seguida, esses atributos são submetidos a modelos com alta capacidade de ajuste, como *ensembles* de árvores de decisão. Essa abordagem pode alcançar alto desempenho, mas envolve um custo informacional considerável: o modelo depende de grande quantidade de variáveis, múltiplas etapas de pré-processamento, balanceamento, seleção e validação. Além disso, quanto mais complexo o *pipeline*, maior a necessidade de controle metodológico para evitar sobreajuste e vazamento de dados.

O aprendizado de máquina quântico oferece uma alternativa conceitual. Em vez de ampliar indefinidamente o número de atributos clássicos, pode-se tentar mapear um conjunto reduzido de variáveis informativas para um espaço vetorial de alta dimensão induzido por um circuito quântico. Em classificadores como o *Quantum Support Vector Classifier* (QSVC), os dados clássicos são codificados em estados quânticos e a similaridade entre amostras é estimada por meio de um kernel quântico. A hipótese investigativa é que esse mapeamento possa tornar a separação entre classes mais eficiente, explorando a geometria do Espaço de Hilbert.

Entretanto, a aplicação prática desse paradigma encontra barreiras significativas. A primeira delas é o problema de codificação dos dados, ou *data embedding*. Dados clássicos precisam ser convertidos em parâmetros de portas quânticas, como ângulos de rotação ou amplitudes de estado. Uma codificação inadequada pode destruir a informação relevante antes mesmo que o circuito seja executado. Por exemplo, uma compressão linear excessiva pode eliminar relações não-lineares essenciais; por outro lado, um mapeamento angular muito amplo pode fazer amostras distintas ocuparem regiões equivalentes devido à periodicidade das funções trigonométricas.

A segunda barreira está relacionada à era NISQ (*Noisy Intermediate-Scale Quantum*) (11). Computadores quânticos atuais possuem número limitado de *qubits*, ruído significativo, conectividade imperfeita e tempos de coerência restritos. Isso torna inviável, no curto prazo, depender de circuitos profundos e altamente emaranhados. Portanto, a pesquisa precisa considerar arquiteturas rasas, com número reduzido de *qubits* e estratégias de codificação compatíveis com limitações físicas reais, mesmo quando a execução experimental ocorre

em simuladores ideais.

Diante desse cenário, a pergunta central desta pesquisa é formulada da seguinte maneira: **é possível alcançar desempenho competitivo na classificação de padrões da atividade elétrica cardíaca, derivados de séries de intervalos RR, utilizando algoritmos quânticos híbridos rasos, com menor quantidade de características, quando comparados a um *pipeline* clássico robusto e dimensionalmente mais amplo?**

Essa pergunta conduz a uma investigação de engenharia, e não apenas a uma comparação de acurácias. O estudo busca compreender quais escolhas de *feature encoding*, topologia de emaranhamento, número de *qubits*, profundidade de circuito e ajuste clássico de hiperparâmetros contribuem para estabilizar ou degradar o desempenho do classificador quântico. A pesquisa também procura delimitar com precisão o que pode ser concluído a partir de simulações: competitividade experimental e eficiência dimensional podem ser avaliadas; vantagem quântica prática em hardware físico, por outro lado, permanece como etapa futura.

1.3 Objetivos

1.3.1 Objetivo Geral

Avaliar a viabilidade técnica de algoritmos de *Quantum Machine Learning* aplicados à classificação de sinais biomédicos derivados da atividade elétrica cardíaca, realizando um *benchmarking* comparativo entre um *pipeline* clássico robusto de alta performance e arquiteturas quânticas híbridas simuladas, com foco em desempenho preditivo, eficiência dimensional, estabilidade estatística e limitações impostas pela era NISQ.

1.3.2 Objetivos Específicos

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- Caracterizar séries temporais baseadas em intervalos RR derivados do ECG por meio de métricas lineares, entrópicas, energéticas, geométricas e fractais, preservando a integridade fisiológica dos dados;
- Construir um *pipeline* clássico de referência baseado em *Ensemble Learning*, capaz de estabelecer um teto empírico de desempenho para o conjunto de dados utilizado;
- Implementar mecanismos rigorosos de prevenção de *data leakage*, garantindo que escalonamento, balanceamento sintético, seleção de atributos e treinamento sejam realizados exclusivamente nas partições adequadas;

- Desenvolver e comparar diferentes arquiteturas de *Quantum Feature Maps*, analisando o impacto de estratégias como PCA, seleção direta de atributos, compressão angular, emaranhamento linear, completo e circular, além de *Data Re-uploading*;
- Investigar a relação entre número de *qubits*, profundidade de circuito e desempenho de classificação, observando fenômenos como interferência destrutiva, saturação de fase e possíveis comportamentos associados a Platôs Estéreis;
- Comparar os modelos clássico e quântico sob a ótica de acurácia média, acurácia de pico, *recall* por classe, custo temporal de execução e quantidade de características utilizadas;
- Delimitar o alcance das conclusões obtidas em simulação, distinguindo paridade experimental, eficiência dimensional e limitações práticas para execução futura em processadores quânticos físicos.

1.4 Justificativa

A relevância deste trabalho decorre da necessidade de investigar aplicações de computação quântica em cenários mais próximos de problemas reais de engenharia. Parte significativa da literatura introdutória de *Quantum Machine Learning* utiliza bases pequenas, sintéticas ou artificialmente separáveis, o que facilita a demonstração conceitual dos algoritmos, mas limita a avaliação de sua utilidade prática. Dados biomédicos, por sua vez, oferecem um ambiente experimental mais exigente: são ruidosos, heterogêneos, não-lineares e sensíveis a escolhas de pré-processamento. Assim, ao aplicar modelos quânticos híbridos à classificação de padrões da atividade elétrica cardíaca derivados de intervalos RR, este trabalho contribui para uma avaliação mais realista do potencial e das limitações do QML.

Do ponto de vista da Engenharia de Computação, a pesquisa é justificada pela análise comparativa entre duas formas distintas de lidar com complexidade. A abordagem clássica opera por expansão de atributos e força estatística: extrai muitas características e utiliza modelos de alta capacidade para encontrar fronteiras de decisão. A abordagem quântica, em contraste, busca projetar um conjunto reduzido de descritores em um espaço de representação mais expressivo. Comparar esses paradigmas permite discutir não apenas qual modelo alcança maior acurácia, mas qual arquitetura utiliza a informação de maneira mais compacta.

Outro aspecto relevante é a documentação das falhas experimentais. Em trabalhos aplicados, é comum apresentar apenas o modelo final, omitindo iterações malsucedidas. Nesta pesquisa, as versões intermediárias possuem valor metodológico, pois mostram que a computação quântica não produz bons resultados automaticamente. O desempenho depende fortemente da forma como os dados são codificados. Experimentos com compressão

linear via PCA, mapeamentos angulares amplos, excesso de profundidade ou inclusão indiscriminada de variáveis demonstram que escolhas aparentemente razoáveis no contexto clássico podem ser inadequadas no domínio quântico.

A justificativa também se apoia na importância biomédica da VFC. A hipertensão arterial é uma condição associada a alterações autonômicas e a mudanças na dinâmica da atividade cardíaca. Embora este trabalho não tenha finalidade diagnóstica clínica imediata, ele explora a possibilidade de identificar padrões fisiológicos por meio de técnicas computacionais avançadas. A contribuição está no desenvolvimento de um arcabouço experimental que pode ser posteriormente expandido para outras bases, outros sinais e ambientes de validação mais amplos.

Por fim, a pesquisa possui relevância por adotar uma postura crítica em relação ao discurso de vantagem quântica. Em vez de afirmar superioridade de forma abstrata, o trabalho constrói um *benchmark* clássico forte e compara o modelo quântico contra esse limite. Dessa forma, as conclusões tornam-se mais equilibradas: quando o modelo quântico se aproxima do desempenho clássico utilizando menos atributos, isso sugere eficiência dimensional; quando falha, revela gargalos de codificação e limitações NISQ. Em ambos os casos, o resultado é útil para orientar futuras investigações em computação quântica aplicada a dados biomédicos baseados em sinais cardíacos.

1.5 Estrutura do Trabalho

Este trabalho está organizado em cinco capítulos principais, apresentados de forma progressiva para conduzir o leitor desde a motivação do problema até a análise final dos resultados.

O **Capítulo 1** apresenta a contextualização do problema, a definição da questão de pesquisa, os objetivos, a justificativa e a estrutura geral do documento. O **Capítulo 2** desenvolve a fundamentação teórica, abrangendo sinais biomédicos, Variabilidade da Frequência Cardíaca, métricas de dinâmica não-linear, aprendizado de máquina clássico, prevenção de vazamento de dados, fundamentos de computação quântica, QSVC, *Quantum Feature Maps* e limitações da era NISQ.

O **Capítulo 3** descreve a metodologia experimental, incluindo aquisição e organização dos dados, engenharia de características, construção do *pipeline* clássico de referência, arquitetura dos experimentos quânticos e métricas de avaliação. O **Capítulo 4** apresenta os resultados e a discussão comparativa entre o modelo clássico e o modelo quântico híbrido final, considerando acurácia, *recall*, estabilidade, custo computacional e eficiência dimensional. O **Capítulo 5** sintetiza as conclusões, delimita o alcance das afirmações obtidas por simulação e apresenta limitações e trabalhos futuros.

Por fim, os **Apêndices** apresentam os elementos de reprodutibilidade do trabalho. O Apêndice A sintetiza os principais parâmetros experimentais; os Apêndices B e C apre-

sentam, respectivamente, os códigos-fonte do *benchmark* clássico final e do modelo quântico híbrido final; e o Apêndice D documenta uma implementação auxiliar da validação cega por amostras retidas, utilizada como auditoria complementar durante o desenvolvimento.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta os fundamentos necessários para compreender a construção do *benchmarking* proposto entre processamento digital clássico e algoritmos quânticos híbridos aplicados a dados biomédicos derivados da atividade elétrica cardíaca. Inicialmente, discute-se a natureza dos sinais biomédicos utilizados, com ênfase na Variabilidade da Frequência Cardíaca (VFC), nos intervalos RR derivados do eletrocardiograma (ECG) e nas métricas de dinâmica não-linear empregadas para representar padrões de modulação autonômica cardíaca. Em seguida, revisam-se os elementos de *Machine Learning* clássico que servem como linha de base experimental, destacando os cuidados necessários para evitar vazamento de dados e sobreajuste em bases biomédicas. Por fim, são apresentados os conceitos de computação quântica, *Quantum Machine Learning* (QML), *Quantum Support Vector Classifier* (QSVC), codificação de dados em circuitos quânticos e limitações práticas da era NISQ.

2.1 Sinais Biomédicos e Dinâmica Não-Linear

Sinais biomédicos são registros de processos fisiológicos produzidos por organismos vivos. Entre os exemplos mais comuns estão o eletrocardiograma (ECG), o eletroencefalograma (EEG), o eletromiograma (EMG), sinais respiratórios, pressão arterial, saturação de oxigênio e séries derivadas de sensores vestíveis. Esses sinais possuem importância clínica e computacional porque representam, de forma indireta, o comportamento de sistemas biológicos regulados por múltiplas variáveis. Diferentemente de sinais industriais ou sintéticos, os sinais biomédicos não são perfeitamente repetíveis: variam de indivíduo para indivíduo, sofrem influência de idade, intervenção farmacológica, estado emocional, condição física, postura, sono e ambiente. Essa característica é particularmente importante neste trabalho, pois a classe hipertensa analisada corresponde a indivíduos com hipertensão arterial em intervenção farmacológica, o que torna a separação entre classes mais sutil do que uma comparação entre indivíduos saudáveis e pacientes sem tratamento.

Neste trabalho, o foco recai sobre a Variabilidade da Frequência Cardíaca (VFC),

analisada por meio de séries de intervalos RR. O intervalo RR corresponde ao tempo entre dois picos R consecutivos no complexo QRS de um ECG. A sequência desses intervalos forma uma série temporal que descreve a oscilação batimento a batimento da atividade cardíaca. A Figura 2.1 ilustra a origem do intervalo RR a partir de um traçado real de ECG e apresenta, em painel separado, uma série RR real utilizada nas análises deste estudo.

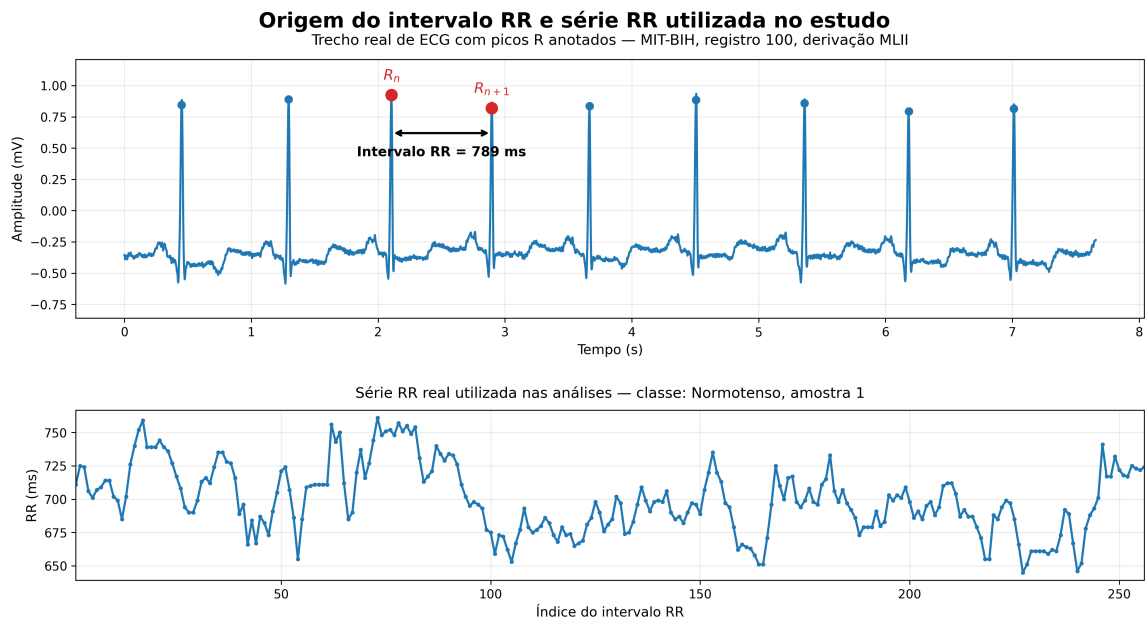


Figura 2.1: Exemplo de medição do intervalo RR em um traçado real de eletrocardiograma, acompanhado de uma série RR real utilizada neste trabalho. O painel superior apresenta um trecho do registro 100 da base MIT-BIH Arrhythmia Database, disponibilizada pelo PhysioNet, utilizado para ilustrar a identificação de dois picos R consecutivos e a obtenção do intervalo RR correspondente. O painel inferior apresenta uma série de intervalos RR do conjunto de dados empregado nos experimentos deste estudo, em que cada ponto representa um intervalo RR previamente extraído do ECG. Os painéis não correspondem ao mesmo indivíduo.

Fonte: elaboração própria com base no MIT-BIH Arrhythmia Database/PhysioNet e no conjunto de dados utilizado neste trabalho (12, 13).

A interpretação fisiológica da VFC está associada ao Sistema Nervoso Autônomo (SNA). O ramo simpático tende a aumentar a frequência cardíaca e preparar o organismo para situações de esforço, alerta ou estresse. O ramo parassimpático, especialmente por meio da atividade vagal, atua reduzindo a frequência cardíaca e favorecendo estados de repouso e recuperação. A VFC emerge da interação dinâmica entre esses dois ramos. Assim, um padrão cardíaco saudável não é simplesmente aquele que apresenta batimentos perfeitamente regulares, mas aquele que demonstra capacidade de adaptação por meio de variações sutis e complexas (1, 2).

Em indivíduos fisiologicamente resilientes, a série RR tende a apresentar irregularidade controlada, refletindo a flexibilidade da modulação autonômica cardíaca. Em

condições como hipertensão arterial, insuficiência autonômica ou estresse crônico, a variabilidade pode ser reduzida ou reorganizada em padrões menos complexos. A perda de complexidade não significa apenas redução de amplitude; ela pode aparecer como diminuição da diversidade temporal, maior previsibilidade, alteração na distribuição de padrões ordinais, mudança na geometria do espaço de fase ou redução de propriedades fractais (3, 4, 14, 6, 7, 8).

2.1.1 Características Lineares e Limitações Estatísticas

A análise clássica da VFC costuma iniciar por métricas lineares no domínio do tempo. Entre elas, destacam-se média dos intervalos RR, desvio padrão, coeficiente de variação, diferenças sucessivas entre intervalos adjacentes e contagens associadas a variações acima de determinados limiares. Essas métricas são úteis porque fornecem uma primeira descrição da dispersão global do sinal. No entanto, seu alcance é limitado quando o objetivo é capturar alterações sutis de organização temporal.

Duas séries podem apresentar média e desvio padrão semelhantes, mas possuir estruturas internas completamente distintas. Uma pode alternar padrões de forma rica e imprevisível; outra pode repetir ciclos rígidos com pequenas oscilações. Estatísticas lineares tendem a tratar ambas de forma parecida, pois resumem o sinal por valores agregados. Essa limitação motivou a adoção de métricas de dinâmica não-linear, capazes de avaliar regularidade, complexidade, recorrência e organização temporal, aspectos frequentemente discutidos na análise de séries fisiológicas e VFC (3, 4, 5).

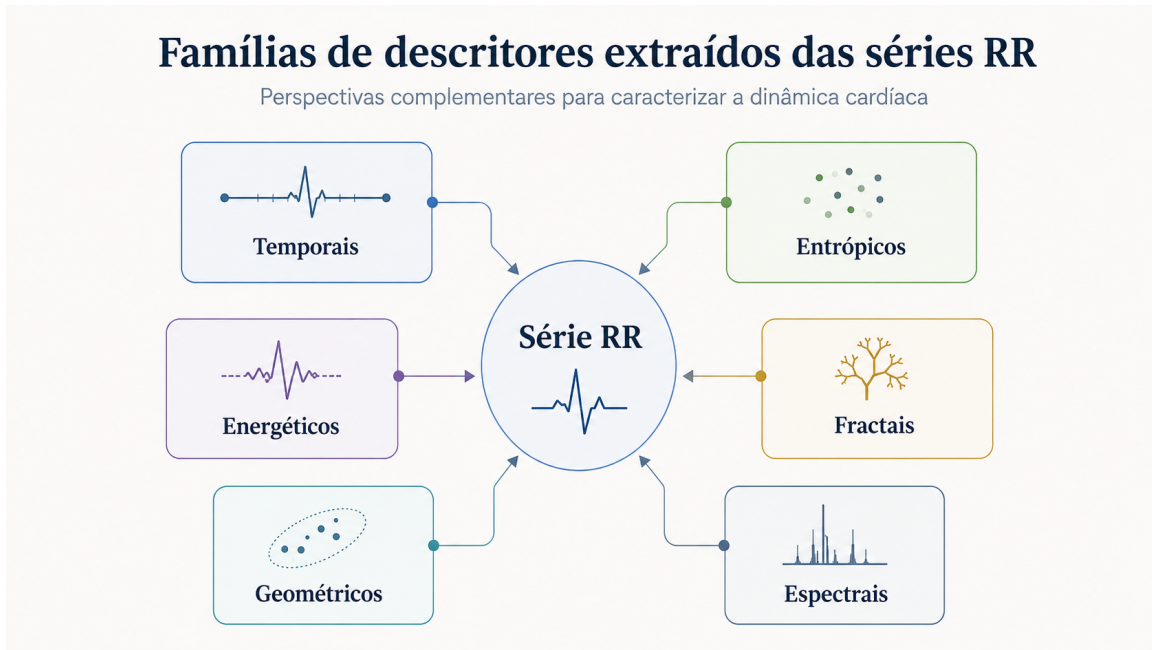


Figura 2.2: Síntese conceitual das famílias de descritores utilizadas para representar séries de intervalos RR, incluindo medidas temporais, entrópicas, energéticas, fractais, geométricas e espectrais.

Fonte: elaboração própria.

2.1.2 Entropia de Amostra

A Entropia de Amostra, ou *Sample Entropy* (SampEn), é uma medida de regularidade projetada para quantificar a probabilidade de padrões semelhantes permanecerem semelhantes quando a dimensão de observação é aumentada. Ela foi proposta como alternativa à Entropia Aproximada, reduzindo o viés causado por auto-correspondência de vetores (3).

Dado um sinal discreto de comprimento N , um comprimento de vetor de imersão m e um raio de tolerância r , a SampEn compara subsequências do sinal em um espaço de dimensão m e verifica quantas delas continuam próximas na dimensão $m + 1$. A formulação geral pode ser expressa por:

$$SampEn(m, r, N) = -\ln\left(\frac{A}{B}\right), \quad (2.1)$$

em que B representa o número de pares de vetores semelhantes em dimensão m , e A representa o número de pares que permanecem semelhantes em dimensão $m + 1$. Valores maiores indicam maior irregularidade e menor previsibilidade. Em séries RR, valores reduzidos podem sugerir maior regularidade, possivelmente associada à perda de adaptabilidade autonômica.

A SampEn é particularmente relevante em séries derivadas da atividade cardíaca porque não exige estacionariedade perfeita e pode ser aplicada a séries relativamente

curtas, desde que os parâmetros sejam escolhidos com cuidado. No contexto deste trabalho, ela compõe o conjunto de características de maior relevância para os modelos quânticos, justamente por condensar informação não-linear em um único descritor de alta densidade fisiológica.

2.1.3 Entropia de Permutação

A Entropia de Permutação, ou *Permutation Entropy* (PE), quantifica a complexidade de uma série temporal a partir da ordem relativa dos valores em janelas sucessivas (4). Em vez de depender diretamente das amplitudes, a PE transforma pequenos segmentos do sinal em padrões ordinais. Por exemplo, em uma janela de três pontos, os valores podem estar em ordem crescente, decrescente ou em outras permutações possíveis. A frequência de ocorrência desses padrões compõe uma distribuição de probabilidade.

Na implementação utilizada neste trabalho, a PE é calculada em sua forma normalizada, pela entropia de Shannon aplicada à distribuição ordinal e dividida pelo valor máximo possível para a dimensão de imersão:

$$PE_{norm} = -\frac{\sum_{i=1}^{D!} p_i \log_2(p_i)}{\log_2(D!)}. \quad (2.2)$$

em que D é a dimensão de imersão e p_i representa a probabilidade de ocorrência do i -ésimo padrão ordinal. A normalização limita a métrica ao intervalo de 0 a 1, facilitando sua comparação com outros descritores. Quando muitos padrões aparecem com frequências semelhantes, a entropia é elevada, indicando diversidade dinâmica. Quando poucos padrões dominam a série, a entropia é reduzida, indicando maior previsibilidade.

Essa métrica é robusta a ruídos de amplitude e a transformações monotônicas, pois considera a ordem relativa dos pontos. Por essa razão, é adequada para séries RR sujeitas a variações individuais e ruído fisiológico. Neste trabalho, a PE aparece como uma das características centrais para os modelos quânticos, pois traduz a complexidade ordinal do sinal em uma variável compacta e informativa.

2.1.4 Entropia Multiescala e Complexidade Temporal

A Entropia Multiescala (*Multiscale Entropy* – MSE) parte da observação de que a complexidade fisiológica não se manifesta em uma única escala temporal. Um sinal pode parecer irregular em curto prazo e organizado em longo prazo, ou o contrário. A MSE avalia a entropia do sinal após diferentes processos de reamostragem e agregação, permitindo investigar a persistência da complexidade em escalas múltiplas (5).

No caso de séries RR, essa abordagem é importante porque a regulação autonômica da atividade cardíaca ocorre em diferentes ritmos: respiração, resposta barorreflexa, atividade autonômica lenta e ajustes hemodinâmicos. Um modelo que observa apenas

uma escala pode ignorar assinaturas relevantes. Na evolução do *pipeline* clássico deste trabalho, a inclusão de componentes multiescala foi decisiva para elevar o desempenho médio, indicando que a informação discriminativa do problema não estava restrita ao domínio temporal imediato.

2.1.5 Energia de Teager-Kaiser e Fragmentação do Sinal

O Operador de Energia de Teager-Kaiser (TKEO) é uma ferramenta utilizada para estimar a energia instantânea de sinais oscilatórios (15). Em séries biomédicas, sua aplicação permite capturar mudanças abruptas e microvariações de energia que podem não ser refletidas em estatísticas globais. Para um sinal discreto $x[n]$, uma forma comum do operador é:

$$\Psi[x[n]] = x[n]^2 - x[n-1]x[n+1]. \quad (2.3)$$

Em séries RR, descritores derivados do TKEO podem revelar fragmentações, instabilidades ou variações locais de dinâmica. No histórico experimental desta pesquisa, atributos relacionados ao TKEO apareceram repetidamente entre as características mais relevantes, especialmente em combinação com medidas entrópicas.

2.1.6 Descritores Fractais, Geométricos e de Poincaré

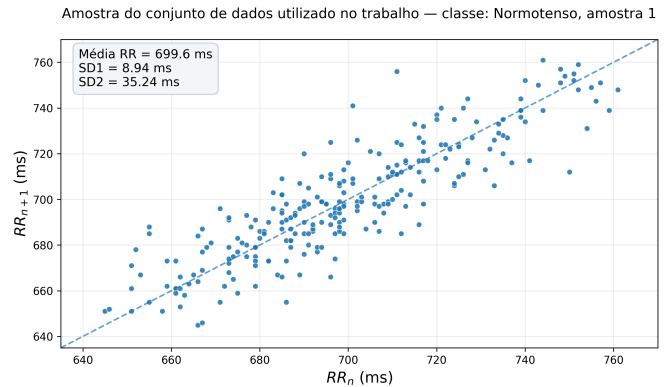
Séries RR derivadas da atividade elétrica cardíaca apresentam comportamento que pode ser interpretado sob a ótica de sistemas complexos. Métricas fractais buscam quantificar a rugosidade, a auto-similaridade ou a complexidade geométrica de uma série temporal. A Dimensão Fractal de Higuchi, por exemplo, estima a complexidade da curva em diferentes escalas, fornecendo uma indicação de quanto o sinal se afasta de uma trajetória simples (7). A Análise de Flutuação Destendenciada (DFA) também é utilizada em séries fisiológicas para quantificar propriedades de escala em sinais não estacionários (6).

Outra forma de representar a dinâmica dos intervalos RR é o gráfico de Poincaré, no qual cada intervalo RR_n é plotado contra o intervalo seguinte RR_{n+1} (14). Essa representação permite visualizar a dispersão batimento a batimento e extrair descritores como $SD1$, $SD2$ e relações entre variabilidade de curto e longo prazo. Em conjunto, essas métricas ajudam a transformar a série temporal bruta em uma matriz de características numericamente tratável.

2.1.7 Síntese para a Modelagem Computacional

A análise de sinais biomédicos neste trabalho não se limita à extração de muitas variáveis. O objetivo é identificar quais descritores carregam maior densidade informacional

Exemplo de gráfico de Poincaré aplicado a uma série real de intervalos RR



Cada ponto representa a relação entre dois intervalos RR consecutivos de uma amostra real da base experimental.

Figura 2.3: Exemplo de gráfico de Poincaré construído a partir de uma série RR, no qual cada ponto representa a relação entre RR_n e RR_{n+1} .

Fonte: elaboração própria.

para distinguir padrões da atividade elétrica cardíaca. A etapa clássica explora um conjunto amplo de características para estabelecer um limite de desempenho. A etapa quântica, por sua vez, testa a hipótese de que um subconjunto pequeno, composto por métricas como SampEn, Entropia de Permutação, TKEO e transformações específicas do sinal, pode ser suficiente quando projetado em um espaço de representação de alta dimensão induzido por circuitos quânticos.

2.2 *Machine Learning* Clássico e Prevenção de *Data Leakage*

O aprendizado de máquina supervisionado consiste em treinar modelos capazes de associar vetores de entrada a rótulos conhecidos. No contexto deste trabalho, cada amostra é representada por um conjunto de características extraídas de uma série de intervalos RR, enquanto o rótulo indica a classe considerada no experimento: normotenso ou classe hipertensa em intervenção farmacológica. O objetivo é aprender uma função de decisão capaz de classificar novas amostras a partir de padrões numéricos presentes nos descritores.

Em bases biomédicas estruturadas, modelos baseados em árvores de decisão e seus comitês são amplamente utilizados devido à sua capacidade de representar relações não-lineares, interações entre variáveis e limiares complexos. Diferentemente de modelos lineares simples, árvores de decisão particionam o espaço de atributos em regiões, permitindo que diferentes combinações de descritores sejam relevantes em diferentes partes do domínio. Quando muitas árvores são combinadas, o modelo resultante tende a apresentar maior estabilidade e desempenho.

2.2.1 Modelos Baseados em Árvores e *Ensemble Learning*

A filosofia do *Ensemble Learning* baseia-se na combinação de múltiplos estimadores para reduzir variância, controlar viés e melhorar generalização (16). Em vez de depender de um único classificador, um comitê agrega decisões de diferentes modelos. Essa agregação pode ocorrer por votação majoritária, média de probabilidades ou combinação ponderada.

No desenvolvimento do *baseline* clássico, dois modelos foram especialmente relevantes:

- ***Extra Trees Classifier***: modelo baseado em árvores extremamente aleatorizadas. Diferentemente de florestas aleatórias tradicionais, o *Extra Trees* introduz maior aleatoriedade na escolha dos pontos de divisão, o que tende a reduzir variância e aumentar robustez em bases com ruído (17);
- ***Extreme Gradient Boosting (XGBoost)***: técnica de *boosting* que constrói árvores sequencialmente, de modo que cada nova árvore busca corrigir erros residuais das anteriores. O XGBoost incorpora mecanismos de regularização e controle de complexidade, tornando-se competitivo em problemas tabulares (18).

A combinação desses modelos em um *Voting Classifier* permite explorar comportamentos complementares. O *Extra Trees* contribui com diversidade estocástica e robustez, enquanto o XGBoost contribui com refinamento iterativo das fronteiras de decisão. No entanto, a capacidade desses modelos também exige cautela: classificadores muito expressivos podem memorizar padrões espúrios se o protocolo experimental não for rigoroso.

2.2.2 Escalonamento, Balanceamento e Seleção de Características

Embora árvores de decisão sejam menos sensíveis a escalas absolutas do que modelos baseados em distância, etapas de pré-processamento ainda são importantes quando há combinação com outros algoritmos, análise de importância de atributos e aplicação de técnicas de balanceamento. O escalonamento padroniza a distribuição das variáveis, evitando que diferenças de unidade ou amplitude distorçam etapas posteriores do *pipeline*.

Em problemas biomédicos, é comum haver desbalanceamento entre classes. Quando uma classe aparece com maior frequência, o modelo pode alcançar boa acurácia simplesmente favorecendo a classe majoritária. Para mitigar esse problema, técnicas de superamostragem sintética, como SMOTE e suas variações, geram amostras artificiais no espaço de atributos (19). Neste trabalho, a versão operacional utilizada no código final foi o *BorderlineSMOTE*, que prioriza amostras próximas à fronteira entre classes (20). O objetivo é fornecer ao classificador uma fronteira de decisão mais equilibrada, especialmente em regiões próximas à separação entre classes.

A seleção de características também é uma etapa crítica. Em um modelo clássico de força bruta, muitas variáveis podem ser incluídas para maximizar desempenho. Contudo,

variáveis redundantes ou pouco informativas podem aumentar ruído, dificultar interpretação e favorecer sobreajuste. Assim, a análise de importância de atributos foi utilizada não apenas para melhorar o modelo clássico, mas também para orientar os modelos quânticos, nos quais o número de variáveis injetadas no circuito é limitado pelo número de *qubits*.

2.2.3 *Data Leakage*: Definição e Riscos

O vazamento de dados, ou *data leakage*, ocorre quando informações que deveriam estar restritas ao conjunto de teste ou validação influenciam direta ou indiretamente o treinamento. Em bases biomédicas, esse problema é particularmente perigoso porque pode produzir acurácias artificialmente elevadas, dando a impressão de que o modelo aprendeu um padrão fisiológico quando, na realidade, explorou um atalho metodológico.

Exemplos comuns de *data leakage* incluem ajustar escaladores usando a base completa antes da divisão treino-teste, aplicar balanceamento sintético antes do *split*, selecionar características usando todas as amostras, manter identificadores de pacientes na matriz de entrada ou permitir que segmentos derivados do mesmo indivíduo apareçam simultaneamente em treino e teste. Modelos de alta capacidade, como *ensembles* de árvores, são especialmente eficientes em explorar esses atalhos.

Por essa razão, todo *pipeline* confiável deve confinar as transformações aprendidas ao conjunto de treinamento. O conjunto de teste deve simular dados inéditos. Escaladores, técnicas de balanceamento, seleção de atributos e ajustes de hiperparâmetros precisam ser aplicados de forma hierarquicamente correta. Esse cuidado metodológico é indispensável para que o desempenho obtido represente generalização real, e não memorização indireta.

A representação visual completa desse fluxo é apresentada no Capítulo 3, na seção metodológica, onde o *pipeline* clássico é descrito em conjunto com sua implementação experimental.

2.2.4 Validação Estocástica e Interpretação de Desempenho

Uma única divisão treino-teste pode produzir resultados dependentes do acaso, especialmente em bases pequenas ou moderadas. Para reduzir esse risco, foram utilizadas múltiplas rodadas estocásticas de validação. A cada rodada, as partições são reorganizadas conforme uma semente aleatória, e o modelo é treinado e avaliado novamente. A média de desempenho fornece uma estimativa mais estável, enquanto o melhor resultado observado indica o potencial máximo em uma partição favorável.

Essa distinção entre média e pico é importante. O pico isolado não deve ser interpretado como desempenho esperado do modelo, mas como evidência de capacidade discriminativa em uma configuração específica dos dados. A média, por outro lado, é mais representativa da robustez. Neste trabalho, ambas as métricas são relatadas, mas a comparação principal prioriza a acurácia média e o *recall* por classe.

2.3 Fundamentos de Computação Quântica

A computação quântica é um paradigma de processamento de informação baseado nos princípios da mecânica quântica. Enquanto a computação clássica representa informação por meio de bits, que assumem valores discretos 0 ou 1, a computação quântica utiliza *qubits*. Um *qubit* pode ser descrito como uma combinação linear de dois estados de base, usualmente denotados por $|0\rangle$ e $|1\rangle$. De forma geral, seu estado é escrito como:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (2.4)$$

em que α e β são amplitudes complexas e satisfazem a condição $|\alpha|^2 + |\beta|^2 = 1$. O quadrado do módulo dessas amplitudes representa a probabilidade de observar cada estado após uma medição. Essa propriedade diferencia o *qubit* do bit clássico: antes da medição, o estado pode estar em superposição.

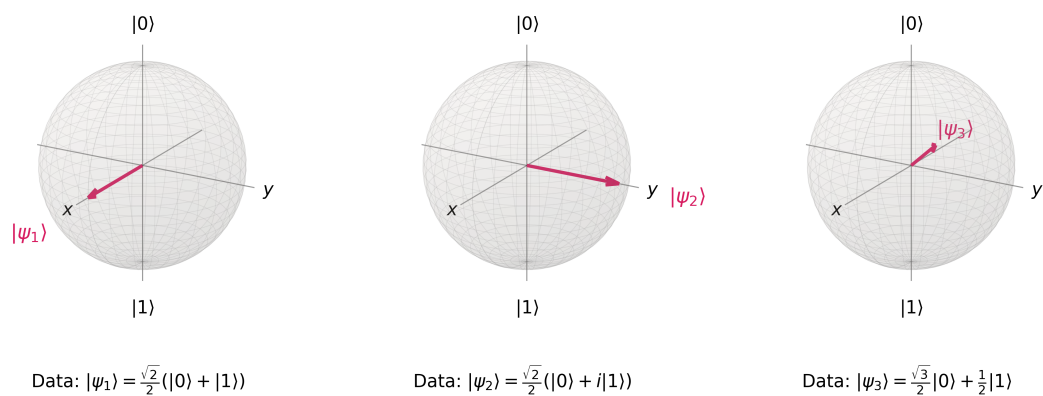


Figura 2.4: Representação esquemática de estados de um *qubit* na esfera de Bloch, ilustrando como diferentes amplitudes e fases correspondem a diferentes orientações do vetor de estado.

Fonte: elaboração própria.

2.3.1 Espaço de Hilbert e Crescimento Dimensional

O estado de um sistema quântico de n *qubits* pertence a um Espaço de Hilbert de dimensão 2^n . Assim, um sistema com 1 *qubit* possui 2 amplitudes, um sistema com 2 *qubits* possui 4 amplitudes, e um sistema com 8 *qubits* possui 256 amplitudes complexas. Esse crescimento exponencial é uma das razões pelas quais a computação quântica desperta interesse em problemas de representação de dados.

É importante observar que esse aumento dimensional não significa que toda informação possa ser lida diretamente. A medição de um sistema quântico retorna resultados probabilísticos e colapsa o estado observado. Portanto, a vantagem potencial da computação quântica não está em simplesmente armazenar muitos números, mas em manipular

amplitudes, fases e interferências de forma a tornar certas propriedades computacionalmente acessíveis.

2.3.2 Portas Quânticas e Circuitos

Operações sobre *qubits* são implementadas por portas quânticas, representadas matematicamente por matrizes unitárias. Portas de rotação, por exemplo, alteram o estado de um *qubit* em torno de eixos da Esfera de Bloch. Portas como $R_X(\theta)$, $R_Y(\theta)$ e $R_Z(\theta)$ recebem parâmetros angulares e são amplamente utilizadas em circuitos parametrizados. A escolha desses parâmetros pode ser usada para codificar dados clássicos.

Além de portas de um *qubit*, circuitos quânticos utilizam portas de dois *qubits*, como CNOT e operações controladas, para criar emaranhamento. O emaranhamento é um fenômeno no qual o estado de um *qubit* não pode ser descrito independentemente do estado de outro. Em termos computacionais, ele permite construir correlações que não possuem equivalente direto em representações clássicas simples.

Um circuito quântico é uma sequência organizada dessas portas. Em algoritmos híbridos, parte do processamento ocorre classicamente e parte ocorre por meio do circuito. O computador clássico prepara dados, ajusta parâmetros e interpreta resultados, enquanto o circuito quântico realiza o mapeamento ou transformação de estados. Essa arquitetura híbrida é central na era NISQ, pois computadores quânticos atuais ainda não possuem escala e tolerância a falhas suficientes para executar algoritmos quânticos profundos de forma autônoma.

2.3.3 Codificação de Dados Clássicos

Para aplicar computação quântica a problemas de aprendizado de máquina, é necessário converter dados clássicos em estados quânticos. Essa etapa é conhecida como codificação de dados ou *data embedding*. Existem diferentes estratégias, entre elas:

- **Angle Encoding:** cada característica clássica é mapeada para um ângulo de rotação aplicado a um *qubit*;
- **Amplitude Encoding:** um vetor clássico normalizado é codificado nas amplitudes de um estado quântico;
- **Basis Encoding:** valores discretos são associados diretamente a estados da base computacional;
- **Data Re-uploading:** os mesmos dados são injetados múltiplas vezes no circuito, aumentando expressividade sem necessariamente aumentar a quantidade de variáveis originais.

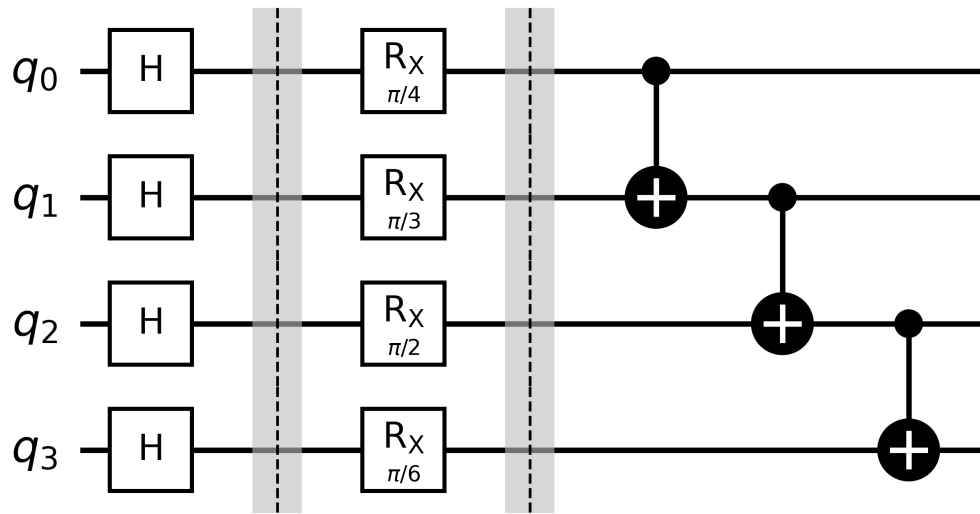


Figura 2.5: Exemplo esquemático de circuito quântico ilustrando preparação em superposição, codificação angular por rotações e criação de correlações entre *qubits*. A figura tem finalidade conceitual e não representa, necessariamente, a decomposição exata do circuito final utilizado nos experimentos.

Fonte: elaboração própria.

Neste trabalho, o foco está principalmente em variações de *Angle Encoding*, por serem compatíveis com circuitos rasos e com *Feature Maps* implementados em simuladores de QML. Contudo, essa escolha traz desafios. Como portas de rotação são periódicas, valores muito espalhados podem produzir efeitos de *wrap-around*, nos quais entradas diferentes passam a representar estados geometricamente próximos. Portanto, o escalonamento angular é uma decisão crítica.

2.3.4 Simulação Clássica de Circuitos Quânticos

Embora circuitos quânticos sejam teoricamente executados em hardware quântico, grande parte da pesquisa atual em QML ocorre em simuladores clássicos. Simuladores baseados em vetor de estado calculam exatamente a evolução das amplitudes do sistema, mas seu custo cresce exponencialmente com o número de *qubits*. Isso explica por que pequenos aumentos na largura do circuito podem gerar grandes aumentos no tempo de execução.

A simulação ideal é útil para avaliar a geometria do modelo sem ruído físico, mas não representa completamente o comportamento de uma QPU real. Em hardware físico, haveria decoerência, erros de porta, erros de medição e restrições de conectividade. Assim, resultados em simulador devem ser interpretados como validação matemática e experimental inicial, não como prova de desempenho clínico em ambiente real.

2.4 *Quantum Machine Learning*, QSVC e Era NISQ

O *Quantum Machine Learning* (QML) investiga como algoritmos quânticos podem ser combinados a métodos de aprendizado de máquina para representar, transformar ou classificar dados (9). Na prática atual, a maior parte das aplicações viáveis é híbrida: o computador clássico organiza a base de dados, executa etapas de pré-processamento e otimização, enquanto o circuito quântico realiza um mapeamento específico, como o cálculo de um kernel ou a geração de uma representação parametrizada.

2.4.1 *Kernels e Classificadores de Vetores de Suporte*

O *Support Vector Machine* (SVM) é um classificador supervisionado que busca encontrar um hiperplano capaz de separar classes com a maior margem possível (21). Quando os dados não são linearmente separáveis no espaço original, o SVM pode usar uma função kernel para calcular similaridades em um espaço de maior dimensão sem construir explicitamente essa transformação. Essa ideia é central para o QSVC.

No *Quantum Support Vector Classifier*, cada amostra clássica x é codificada em um estado quântico $|\Phi(x)\rangle$ por meio de um *Feature Map*. A similaridade entre duas amostras x e z pode ser estimada pela fidelidade entre seus estados:

$$K(x, z) = |\langle \Phi(x) | \Phi(z) \rangle|^2. \quad (2.5)$$

Essa matriz de kernel é então utilizada por um classificador SVM clássico. O circuito quântico, portanto, atua como motor de mapeamento geométrico, enquanto a etapa de decisão permanece clássica. Essa divisão torna o QSVC adequado à era NISQ, pois circuitos relativamente rasos podem ser utilizados para calcular similaridades.

2.4.2 *Quantum Feature Maps*

Um *Quantum Feature Map* define como os dados clássicos são convertidos em operações quânticas. O `ZZFeatureMap`, utilizado em várias configurações deste trabalho, aplica rotações parametrizadas e interações do tipo ZZ entre pares de *qubits*. Essas interações introduzem termos de fase dependentes de combinações entre características, criando um espaço de representação não-linear. A literatura demonstra que mapas de características quânticas podem produzir espaços difíceis de simular classicamente em determinadas condições (10). A implementação experimental deste trabalho foi realizada com Qiskit e Qiskit Machine Learning, bibliotecas voltadas à construção de circuitos, kernels e modelos híbridos (22).

Além do `ZZFeatureMap`, foram avaliadas variações baseadas em `PauliFeatureMap`, incluindo operações com termos Z e YY . Essas arquiteturas aumentam a expressividade do circuito, mas também ampliam profundidade, custo de simulação e sensibilidade a



Figura 2.6: Fluxo conceitual de um modelo quântico híbrido baseado em QSVC: descritores clássicos são escalonados, codificados como parâmetros de um circuito quântico, transformados por um *Quantum Feature Map* e utilizados para compor uma matriz de kernel, posteriormente analisada por um classificador SVM clássico.

Fonte: elaboração própria.

ruído. O equilíbrio entre expressividade e estabilidade é um dos pontos centrais em QML.

2.4.3 Era NISQ

A expressão NISQ, proposta para descrever a geração atual de processadores quânticos, refere-se a dispositivos de escala intermediária, ruidosos e ainda não tolerantes a falhas (11). Esses dispositivos podem possuir dezenas ou centenas de *qubits*, mas apresentam limitações de coerência, fidelidade de portas, conectividade e leitura. Na prática, isso impõe restrições importantes:

- circuitos muito profundos tendem a acumular erros antes da medição;
- portas de dois *qubits*, necessárias para emaranhamento, são mais ruidosas que portas de um *qubit*;
- topologias de hardware podem exigir portas adicionais de roteamento, aumentando a profundidade efetiva;
- repetição de medições é necessária para estimar probabilidades, elevando custo experimental;
- a preparação de estados complexos pode anular possíveis ganhos de representação.

Por essas razões, este trabalho prioriza circuitos rasos e interpreta os resultados em simulador como uma etapa de viabilidade. O objetivo não é ignorar o ruído, mas estudar quais arquiteturas teriam maior chance de transição futura para hardware físico.

2.4.4 Platôs Estéreis e Profundidade de Circuito

Em circuitos quânticos parametrizados, um dos fenômenos mais discutidos é o aparecimento de Platôs Estéreis (*Barren Plateaus*). Eles ocorrem quando a paisagem de otimização torna-se praticamente plana, dificultando o ajuste de parâmetros porque os gradientes se aproximam de zero (23). Embora o QSVC utilizado neste trabalho não dependa de treinamento variacional profundo da mesma forma que modelos variacionais, a noção de saturação geométrica é relevante por analogia: circuitos muito profundos ou com mapeamentos angulares excessivamente amplos podem produzir estados pouco discriminativos, degradando a matriz de kernel.

Nos experimentos descritos posteriormente, a tentativa de aumentar profundidade em circuitos de poucos *qubits* resultou em queda expressiva de desempenho. Esse comportamento reforça uma diretriz prática da era NISQ: para dados biomédicos ruidosos, pode ser mais eficiente explorar circuitos rasos com codificação cuidadosamente calibrada do que simplesmente aumentar camadas de emaranhamento.

2.4.5 *Data Re-uploading* e Eficiência Dimensional

O *Data Re-uploading* consiste, em sua formulação usual, em inserir a mesma informação clássica mais de uma vez em um circuito quântico, frequentemente ao longo de camadas sucessivas (24). Conceitualmente, essa técnica permite que o circuito construa representações mais ricas sem exigir necessariamente novas características. Em modelos quânticos, a repetição controlada dos dados pode ampliar o espaço efetivo de separação.

Neste trabalho, utilizou-se uma estratégia de redundância lateral inspirada em *Data Re-uploading*: quatro características biomédicas de alta relevância foram reinseridas no vetor de entrada para ocupar oito *qubits*. Essa estratégia mostrou-se mais estável do que simplesmente adicionar oito variáveis distintas. O resultado sugere que, no contexto investigado, a qualidade geométrica das características e sua repetição estruturada foram mais importantes do que a quantidade bruta de atributos.

2.4.6 Interpretação Cuidadosa de Resultados em QML

É fundamental distinguir três níveis de conclusão. O primeiro é a viabilidade matemática em simulador ideal: nesse nível, verifica-se se a arquitetura produz uma matriz de similaridade útil. O segundo é a viabilidade experimental em hardware ruidoso: nesse caso, é preciso medir o impacto de decoerência e erros de porta. O terceiro é a

aplicabilidade clínica: exige validação em bases maiores, protocolos independentes e análise estatística rigorosa.

Este trabalho concentra-se principalmente no primeiro nível, com discussão crítica do segundo. Portanto, as conclusões não afirmam uma vantagem quântica universal, mas demonstram que uma arquitetura quântica híbrida simulada pode alcançar desempenho competitivo frente a um *baseline* clássico forte, utilizando menos características. Esse é o fundamento do conceito de eficiência dimensional explorado nos resultados.

Capítulo 3

Metodologia e Desenvolvimento Experimental

Este capítulo descreve a metodologia adotada para comparar o desempenho de *pipelines* clássicos e quânticos híbridos na classificação de padrões da atividade elétrica cardíaca derivados de séries de intervalos RR. A abordagem foi estruturada como um processo experimental evolutivo: primeiro, definiu-se uma base de dados biomédica composta por séries RR derivadas do ECG; em seguida, foram extraídas características lineares, não-lineares, entrópicas, energéticas, fractais, geométricas e espectrais; posteriormente, construiu-se um *pipeline* clássico de referência para estabelecer um teto empírico de desempenho; por fim, foram desenvolvidas e consolidadas arquiteturas quânticas simuladas como marcos de referência, avaliando diferentes estratégias de codificação, topologias de emaranhamento, profundidades de circuito e mecanismos de controle de viés.

A metodologia foi planejada para evitar conclusões artificiais. Por esse motivo, transformações de escala, balanceamento sintético, seleção de características e treinamento de modelos foram confinados às partições corretas de treino e teste. Esse cuidado é essencial em problemas biomédicos, nos quais pequenos vazamentos de informação podem gerar acurácias aparentemente elevadas, mas sem validade científica. O objetivo metodológico principal, portanto, não foi apenas maximizar acurácia, mas construir uma comparação auditável entre paradigmas computacionais distintos.

3.1 Aquisição, Organização e Preparação dos Dados

A base experimental deste trabalho é composta por séries temporais de intervalos RR derivados do ECG. Cada série descreve a sequência de tempos entre picos R consecutivos no ECG, permitindo derivar indicadores de Variabilidade da Frequência Cardíaca e de organização temporal da atividade cardíaca. As amostras foram organizadas em duas classes de interesse: indivíduos normotensos, utilizados como grupo de controle, e indivíduos

com hipertensão arterial em intervenção farmacológica, utilizados como grupo clínico de comparação.

No arquivo experimental empregado nos códigos finais, foram carregadas 66 séries normotensas e 133 séries da classe hipertensa, cada uma representada por 256 pontos de intervalos RR derivados do ECG. Essa composição também explica a necessidade de avaliação por *recall* por classe e de técnicas de balanceamento no treinamento, pois a quantidade de amostras da classe hipertensa é superior à quantidade de amostras normotensas.



Figura 3.1: Organização experimental da base de dados utilizada neste trabalho. As séries RR derivadas do ECG foram separadas em duas classes, transformadas em vetores de características e utilizadas nos protocolos clássico e quântico híbrido.

Fonte: elaboração própria.

O tamanho amostral reduzido deve ser entendido como uma restrição metodológica relevante. A base é adequada para uma investigação exploratória e para a construção de um *benchmark* computacional controlado, mas não sustenta generalizações clínicas amplas. Em conjuntos biomédicos pequenos, a composição das partições de treino e teste pode influenciar de forma perceptível as métricas finais. Por isso, a metodologia priorizou múltiplas rodadas estocásticas, divisão estratificada, controle de vazamento e análise da estabilidade média, em vez de depender de uma única partição favorável.

As informações disponibilizadas para este estudo não continham metadados detalhados sobre a condição postural, o estado de repouso ou o protocolo fisiológico completo de aquisição das séries RR. Por esse motivo, tais variáveis não foram inferidas nem utilizadas como atributos dos modelos. Essa ausência de metadados é tratada como limitação metodológica, especialmente porque postura, respiração, repouso, esforço e condições de

aquisição podem influenciar a Variabilidade da Frequência Cardíaca.

O processo metodológico iniciou-se pela padronização das séries de entrada. Sinais biomédicos podem conter artefatos, batimentos ectópicos, ruídos de aquisição e segmentos inconsistentes. Em um estudo aplicado, essas irregularidades não podem ser ignoradas, pois atributos entrópicos, fractais e geométricos são sensíveis à estrutura temporal do sinal. Assim, a preparação dos dados priorizou a conservação da dinâmica fisiológica, evitando transformações agressivas que pudessem suavizar artificialmente os padrões de variabilidade.

A matriz experimental foi construída a partir da extração de características de cada série RR. Identificadores, índices de controle, colunas auxiliares e qualquer informação sem significado fisiológico foram removidos antes do treinamento dos modelos. Esse cuidado é indispensável porque classificadores de alta capacidade conseguem explorar identificadores numéricos como atalhos, produzindo resultados sem validade científica. Durante o desenvolvimento, auditorias da matriz de entrada reforçaram a necessidade de inspeção rigorosa dos atributos utilizados.

As partições de treino e teste foram formadas de maneira estocástica em múltiplas rodadas. Em cada rodada, o modelo foi treinado e avaliado novamente, permitindo observar média, variação e pico de desempenho. Outro princípio metodológico adotado foi a separação entre etapas ajustáveis e etapas puramente aplicadas. Escaladores, técnicas de balanceamento, seleção de características e modelos foram ajustados apenas com dados de treino. Em seguida, as transformações aprendidas foram aplicadas ao conjunto de teste. Essa regra evita que informações estatísticas do teste sejam incorporadas indiretamente ao treinamento.

A presença da classe hipertensa em intervenção farmacológica exige cautela na interpretação. O classificador não deve ser entendido como ferramenta diagnóstica isolada, mas como experimento de separabilidade entre padrões de VFC de um grupo controle e de uma classe hipertensa sob tratamento farmacológico. Caso a intervenção farmacológica normalizasse integralmente a dinâmica autonômica expressa nos intervalos RR, a distinção computacional tenderia a ser reduzida ou inexistente.

Assim, a etapa de aquisição e organização estabeleceu as bases de confiabilidade do estudo: dados fisiologicamente interpretáveis, matriz de atributos sem identificadores indevidos, divisão estocástica controlada, reconhecimento explícito do tamanho limitado da base, tratamento honesto da ausência de metadados de aquisição e prevenção de vazamento de informação.

3.2 Engenharia de Características Clássicas

A engenharia de características foi desenvolvida para transformar cada série de intervalos RR em um vetor numérico capaz de representar propriedades fisiológicas relevan-

tes. Essa etapa é decisiva porque os modelos de aprendizado, sejam clássicos ou quânticos, não operam diretamente sobre a interpretação clínica do sinal, mas sobre descritores computacionais extraídos dele. Assim, a qualidade das características define o limite informacional disponível para o classificador.

O conjunto inicial de atributos incluiu métricas no domínio do tempo, medidas de dispersão, descritores de variação sucessiva, indicadores de fragmentação, entropias, transformações energéticas, métricas fractais, representações geométricas e descritores espectrais. Essa diversidade foi necessária para estabelecer um *baseline* clássico forte. O objetivo não era escolher previamente uma única família de características, mas permitir que o modelo clássico explorasse diferentes faces da dinâmica da atividade cardíaca expressa nas séries RR.

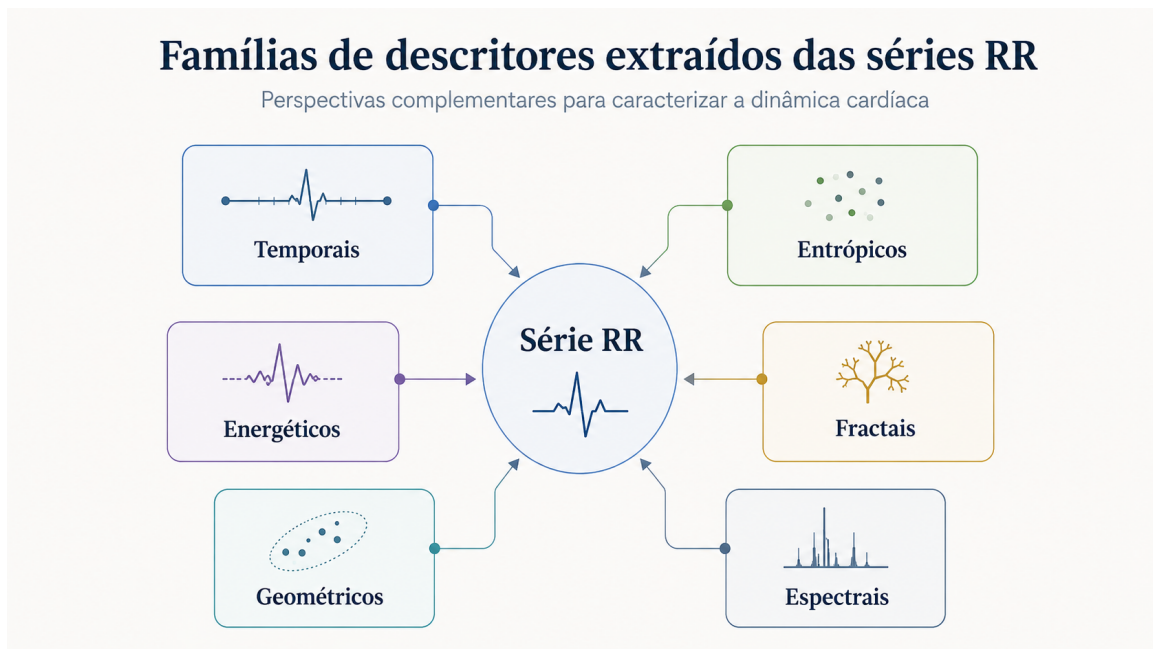


Figura 3.2: Famílias de descritores utilizadas para representar diferentes aspectos das séries RR. Cada grupo de características observa o sinal por uma perspectiva complementar: tempo, irregularidade, energia local, complexidade de escala, geometria entre intervalos consecutivos e comportamento espectral.

Fonte: elaboração própria.

3.2.1 Descritores Temporais e Estatísticos

As características temporais descrevem propriedades globais da série RR. Entre elas estão média, desvio padrão, coeficiente de variação, valores extremos e estatísticas relacionadas às diferenças sucessivas entre batimentos. Esses descritores fornecem uma visão inicial da dispersão e da estabilidade do ritmo cardíaco.

Embora úteis, essas métricas tendem a ser insuficientes isoladamente. A hipertensão arterial e outras alterações da modulação autonômica cardíaca podem modificar a

organização temporal sem necessariamente alterar de forma extrema valores médios. Por esse motivo, os descritores temporais foram tratados como parte de um conjunto maior, e não como representação completa do fenômeno.

3.2.2 Descritores Entrópicos e Não-Lineares

As métricas entrópicas foram centrais no projeto. A Entropia de Amostra, a Entropia de Permutação e variações multiescala foram utilizadas para quantificar irregularidade, diversidade ordinal e persistência de complexidade em diferentes escalas temporais (3, 4, 5). Essas características são coerentes com a hipótese fisiológica de que sistemas cardíacos com maior adaptabilidade preservam complexidade dinâmica, enquanto condições associadas a rigidez autonômica tendem a apresentar maior previsibilidade.

Além das entropias, foram incluídas medidas associadas à dinâmica não-linear e à geometria do sinal. A análise de Poincaré, por exemplo, traduz a relação entre intervalos consecutivos em uma distribuição geométrica (14). Métricas fractais, como a Dimensão de Higuchi e a DFA, avaliam propriedades de escala e complexidade da curva em sinais irregulares ou não estacionários (7, 6, 8). Parâmetros de Hjorth oferecem descritores relacionados à atividade, mobilidade e complexidade do sinal (25).

3.2.3 Energia, Fragmentação e Transformações Derivadas

O Operador de Energia de Teager-Kaiser foi utilizado para capturar variações locais de energia no sinal (15). A inclusão de atributos derivados do TKEO mostrou-se relevante durante a evolução dos modelos, pois esses descritores conseguem destacar mudanças rápidas e padrões de fragmentação que nem sempre aparecem em métricas globais.

Também foram consideradas transformações associadas a decomposições espectrais, energia em diferentes bandas, fragmentação de intervalos e coeficientes de distribuição. Em conjunto, essas variáveis aumentam a capacidade do *pipeline* clássico de detectar nuances, mas também elevam a dimensionalidade da matriz. Essa tensão entre riqueza informacional e complexidade dimensional é justamente o ponto que motiva a comparação com modelos quânticos.

Entre as transformações derivadas, foi incluído um descritor experimental de entropia espectral acelerada. Sua proposta foi incorporar uma aproximação da aceleração da série RR à fase de uma análise espectral, de modo a tornar o descritor sensível não apenas à presença de frequências, mas também a mudanças locais da dinâmica do sinal. Esse recurso deve ser interpretado como uma característica computacional experimental de morfologia dinâmica, e não como métrica clínica padronizada de VFC.

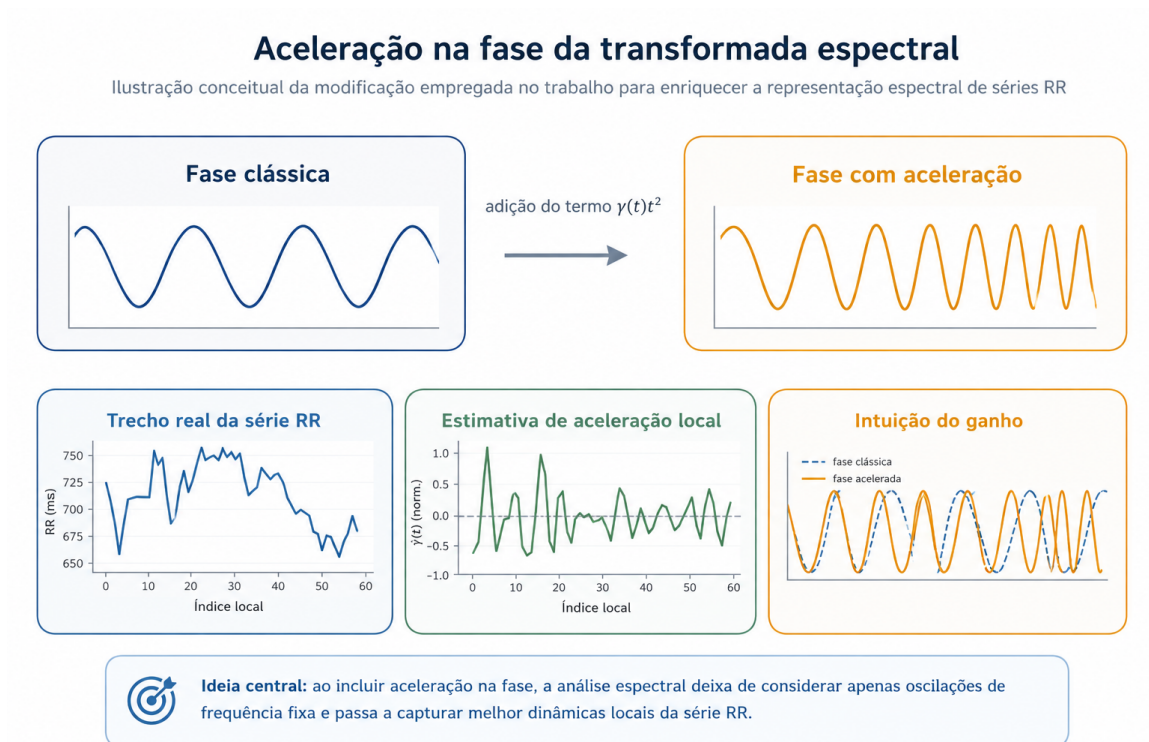


Figura 3.3: Representação conceitual da entropia espectral acelerada. A Transformada de Fourier tradicional observa a distribuição de frequências, enquanto a modificação por aceleração busca sensibilizar a fase a mudanças locais da dinâmica da série RR.

Fonte: elaboração própria.

3.2.4 Seleção de Características para o Modelo Quântico

Embora o modelo clássico pudesse operar com dezenas de atributos, os circuitos quânticos avaliados impõem uma restrição direta: a quantidade de características injetadas depende do número de *qubits* ou da estratégia de codificação. Em um circuito de 4 *qubits* com *Angle Encoding*, por exemplo, apenas quatro variáveis podem ser mapeadas diretamente em uma camada de rotação.

Por isso, foi necessário identificar um subconjunto de características de alta densidade informacional. A análise empírica ao longo dos marcos consolidados indicou que descritores como `sampen`, `permutation_entropy`, `tkeo_sum` e um descritor de entropia espectral acelerada formavam um núcleo robusto para os modelos quânticos. A escolha dessas variáveis não foi arbitrária: elas combinam complexidade temporal, organização ordinal, energia local e uma análise baseada em Fourier aplicada à aceleração do sinal. A implementação final desses descritores está registrada nos Apêndices B e C.

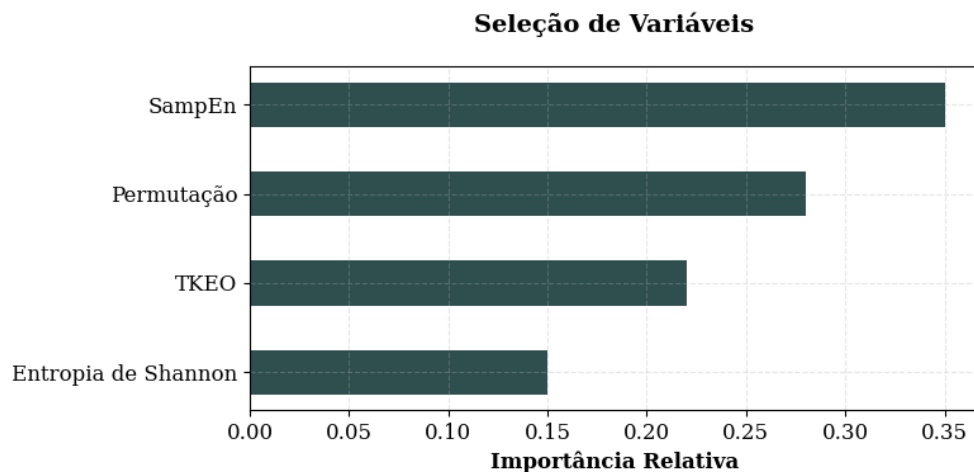


Figura 3.4: Representação esquemática da relevância relativa de variáveis utilizadas na seleção de entrada para o modelo quântico. A seleção priorizou descritores compactos, com alta densidade informacional e significado compatível com a dinâmica das séries RR.

Fonte: elaboração própria.

A engenharia clássica, portanto, cumpriu dupla função. Primeiro, permitiu construir um modelo clássico de referência com alto desempenho. Segundo, forneceu um mecanismo de seleção para reduzir a entrada do modelo quântico a um conjunto compacto de variáveis fisiologicamente significativas.

3.3 Construção do *Pipeline* Clássico de Referência

O *pipeline* clássico foi desenvolvido para atuar como linha de base forte. Essa decisão metodológica é essencial: se o modelo quântico fosse comparado apenas contra classificadores simples, qualquer ganho poderia ser interpretado como consequência de

uma linha de base fraca. Assim, buscou-se estabelecer um teto empírico de desempenho usando técnicas clássicas robustas e amplamente utilizadas em problemas tabulares.

A evolução dos modelos clássicos ocorreu por ciclos sucessivos de experimentação. Ao longo desse processo, apenas os marcos mais consolidados foram preservados como versões de referência, pois representavam configurações tecnicamente interpretáveis após ajustes, falhas intermediárias e testes descartados. As primeiras configurações exploraram combinações compactas de atributos físicos e estatísticos, com desempenho médio próximo de 70%. Em seguida, foram incorporadas métricas caóticas, fractais, parâmetros de Hjorth, fragmentação da VFC, índices de Poincaré, transformações espectrais e entropia multiescala. Essa trajetória permitiu identificar quais famílias de atributos contribuíam para o aumento de desempenho e quais apenas adicionavam complexidade sem ganho proporcional.

O modelo clássico final utilizou uma arquitetura de *Ensemble Learning* baseada em votação suave. A composição envolveu *Extra Trees* e XGBoost, ambos configurados com grande número de estimadores e regularização. O *Extra Trees* contribuiu com diversidade estocástica, enquanto o XGBoost adicionou refinamento sequencial por gradiente. A combinação por votação probabilística permitiu agregar os pontos fortes de ambos.

Para prevenir sobreajuste, foram adotadas restrições como profundidade limitada, regularização, subamostragem e controle da complexidade dos estimadores. Além disso, técnicas de balanceamento foram aplicadas somente após a divisão dos dados, dentro do conjunto de treinamento. Essa escolha é decisiva: aplicar balanceamento antes do *split* permitiria que amostras sintéticas contaminassem o conjunto de teste, gerando vazamento.

A validação do modelo clássico foi realizada em múltiplas rodadas estocásticas. Esse protocolo forneceu uma estimativa mais confiável do desempenho médio e permitiu observar a variabilidade entre partições. O resultado final do modelo clássico foi utilizado como referência para avaliar o modelo quântico. Portanto, o *baseline* não representa um classificador trivial, mas uma aproximação do limite alcançável por força estatística dentro do conjunto de dados utilizado.

A Tabela 3.1 resume a lógica evolutiva do *pipeline* clássico.

Tabela 3.1: Síntese metodológica da evolução do *pipeline* clássico.

Etapa	Descrição	Papel
Configurações iniciais	Uso de atributos compactos, estatísticas básicas e classificadores diretos.	Referência inicial
Expansão não-linear	Inclusão de entropias, fractais, TKEO, Poincaré e fragmentação.	Aumento informacional
Auditoria	Testes contra vazamento, rodadas estocásticas e isolamento de falhas metodológicas.	Validação
Modelo final	Comitê de <i>Extra Trees</i> e XGBoost com regularização e votação suave.	Teto clássico

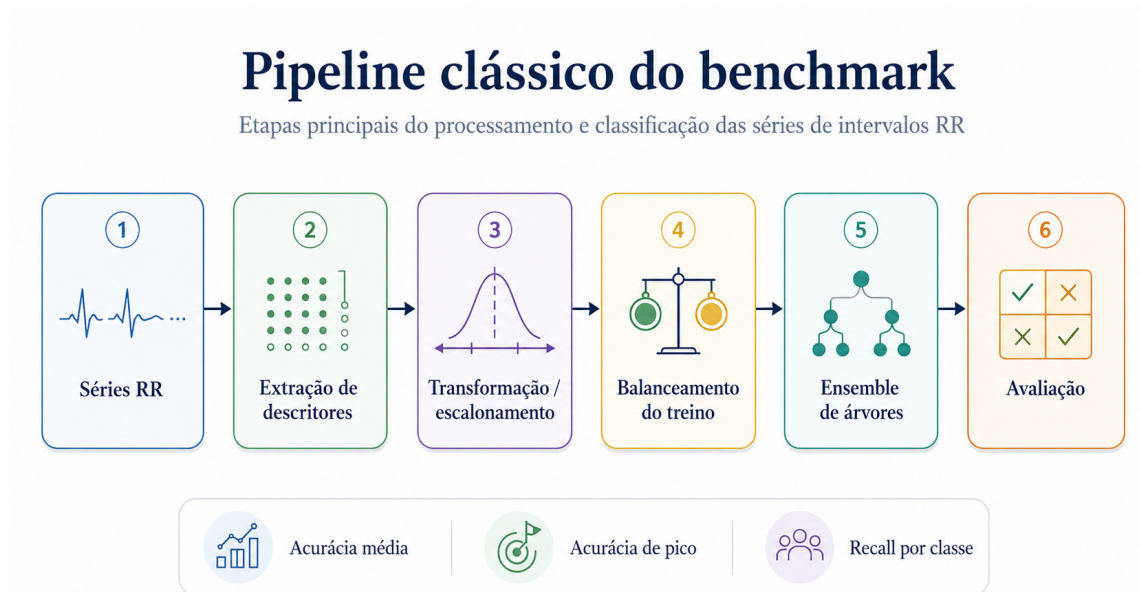


Figura 3.5: Fluxo metodológico do *pipeline* clássico de referência. As etapas aprendidas, como escalonamento e balanceamento sintético, são ajustadas exclusivamente no conjunto de treino, enquanto o conjunto de teste permanece composto apenas por amostras originais.

Fonte: elaboração própria.

3.4 Arquitetura Quântica Híbrida

A arquitetura quântica foi desenvolvida por experimentação controlada. Cada marco consolidado avaliou uma hipótese sobre codificação de dados, topologia de emaranhamento, número de *qubits*, profundidade de circuito ou ajuste do classificador clássico acoplado ao kernel. A numeração interna foi mantida apenas como rastreabilidade dos resultados consolidados, não como enumeração de todas as tentativas realizadas. Diversas configurações intermediárias foram eliminadas durante auditorias e testes cegos por amostras retidas, especialmente quando a acurácia em partições convencionais não se confirmava em observações tratadas como novidade completa. Um exemplo de implementação desse protocolo auxiliar é apresentado no Apêndice D.

O modelo geral utilizado foi o *Quantum Support Vector Classifier* (QSVC). O fluxo experimental consiste em selecionar características clássicas, escaloná-las para um domínio angular adequado, codificá-las em um *Quantum Feature Map*, calcular a matriz de kernel quântico e treinar um SVM usando essa matriz como medida de similaridade entre amostras.

3.4.1 Codificação Angular

A maior parte dos experimentos utilizou *Angle Encoding*. Nessa abordagem, cada característica clássica é convertida em um ângulo aplicado a portas de rotação. A escolha



Figura 3.6: Fluxo metodológico do modelo quântico híbrido. Os descritores clássicos são selecionados, escalonados, codificados como ângulos, processados por um *Quantum Feature Map* e convertidos em uma matriz de kernel utilizada por um SVM clássico.

Fonte: elaboração própria.

do intervalo angular foi tratada como hiperparâmetro crítico. Mapeamentos no intervalo 0 a 2π podem parecer naturais, mas a periodicidade das portas quânticas pode fazer com que valores distintos sejam projetados em regiões equivalentes. Por isso, configurações posteriores testaram intervalos mais restritos, como 0 a π , 0 a $\pi/2$ e, na configuração final, uma faixa angular compacta.

O ajuste angular teve impacto direto no desempenho. A compressão para regiões mais controladas da Esfera de Bloch preservou melhor a monotonicidade das variáveis e reduziu distorções de fase. Essa conclusão reforça que o pré-processamento clássico não é uma etapa secundária no QML; ele define a geometria inicial do problema.

3.4.2 *Feature Maps* e Topologias de Emaranhamento

O ZZFeatureMap foi adotado como estrutura central por permitir interações de fase entre pares de *qubits*. Foram avaliadas topologias linear, completa e circular. A topologia linear forma uma cadeia simples; a completa aumenta expressividade e profundidade; e a circular fecha a cadeia conectando o último *qubit* ao primeiro. A topologia circular mostrou-se relevante em configurações com 8 *qubits*, pois criou uma estrutura de correlação fechada sem exigir a densidade extrema do emaranhamento completo. Em hardware físico, porém, essa escolha dependeria da conectividade real da QPU.

É importante destacar que, no modelo final, os *qubits* não representam pontos

consecutivos da série RR bruta. Eles representam descritores compactos extraídos previamente do sinal. Por isso, a topologia circular deve ser interpretada como uma forma de induzir correlações entre atributos no espaço de representação quântico, e não como uma tentativa de reproduzir diretamente a sequência temporal dos intervalos RR.

3.4.3 Redução por PCA, Seleção Direta e Redundância Lateral

A primeira estratégia para lidar com poucos *qubits* foi o uso de PCA (26). A ideia era comprimir a matriz clássica em poucas componentes principais e injetá-las no circuito. Embora comum no aprendizado clássico, essa abordagem produziu perda significativa de desempenho no contexto quântico, sugerindo que a compressão linear eliminou relações não-lineares relevantes.

A segunda estratégia foi a seleção direta das melhores características. Em vez de comprimir muitas variáveis em componentes lineares, foram selecionados descritores fisiologicamente densos, como SampEn, Entropia de Permutação, energia TKEO acumulada e entropia espectral acelerada. Essa abordagem preservou melhor o significado das variáveis e melhorou o equilíbrio entre classes.

A terceira estratégia, decisiva para o modelo final, foi uma redundância lateral inspirada em *Data Re-uploading*. Como quatro características eram consistentemente fortes, mas quatro *qubits* limitavam a expressividade, duplicou-se o vetor de entrada para alimentar oito *qubits*. Desse modo, a mesma informação clássica foi distribuída em um espaço quântico maior. A técnica produziu aumento de desempenho e estabilidade, sugerindo que a expansão geométrica foi mais útil do que a simples adição de variáveis distintas.

3.4.4 Controle Clássico do Hiperplano

Embora o kernel seja quântico, a etapa final de decisão permanece associada ao SVM clássico. Assim, hiperparâmetros como margem C e pesos de classe influenciam a fronteira de decisão. Durante os experimentos, observou-se que o modelo tendia a favorecer a classe hipertensa em intervenção farmacológica em algumas configurações, prejudicando o *recall* dos normotensos. Para mitigar esse efeito, foram testados pesos manuais assimétricos, atribuindo maior peso à classe minoritária.

Esse resultado demonstra que o modelo quântico híbrido não deve ser interpretado como uma caixa exclusivamente quântica. O desempenho emerge da interação entre preparação clássica, codificação angular, circuito de *feature map*, cálculo de kernel e ajuste do classificador clássico.

3.4.5 Síntese dos Marcos Quânticos Consolidados

A Tabela 3.2 resume as principais hipóteses que permaneceram como marcos consolidados ao longo da evolução quântica.

Tabela 3.2: Síntese metodológica dos principais marcos quânticos avaliados.

Marco	Estratégia	Objetivo experimental
Q-V1	PCA + 4 <i>qubits</i>	Testar compressão linear para adequação ao limite de <i>qubits</i> .
Q-V2 a Q-V4	Seleção direta + ajuste angular	Preservar características não-lineares e controlar distorções de fase.
Q-V5.1	PauliFeatureMap	Aumentar expressividade com portas adicionais e avaliar estabilidade.
Q-V6	<i>Dual Scaling</i> + margem suave	Testar controle clássico contra sobreajuste.
Q-V7 a Q-V9	8 <i>qubits</i> e redundância lateral	Explorar expansão lateral e redundância geométrica.
Q-V10	Aumento de profundidade	Testar se circuitos mais profundos melhoram separação.
Q-V13 a Q-V14	Pesos manuais + redundância lateral	Integrar expansão geométrica e controle clássico do hiperplano.

3.5 Métricas de Avaliação

A avaliação dos modelos utilizou métricas capazes de representar tanto o desempenho global quanto o comportamento por classe. Em problemas biomédicos, a acurácia isolada pode ser insuficiente, especialmente quando há desbalanceamento entre grupos. Um modelo pode apresentar boa acurácia favorecendo a classe majoritária e ainda assim falhar na identificação de padrões relevantes da classe minoritária.

A acurácia global foi calculada como a proporção de classificações corretas em relação ao total de amostras avaliadas:

$$Acuracia = \frac{TP + TN}{TP + TN + FP + FN}, \quad (3.1)$$

em que TP representa verdadeiros positivos, TN verdadeiros negativos, FP falsos positivos e FN falsos negativos.

Também foi analisado o *recall* por classe, definido como a proporção de amostras de uma classe corretamente identificadas:

$$Recall = \frac{TP}{TP + FN}. \quad (3.2)$$

No contexto deste trabalho, o *recall* da classe hipertensa em intervenção farmacológica indica a capacidade do modelo de reconhecer corretamente padrões associados ao grupo clínico, enquanto o *recall* dos normotensos indica a capacidade de preservar a identificação do grupo controle. O equilíbrio entre ambos é relevante para evitar modelos enviesados.

Além das métricas pontuais, foram observadas média, pico e estabilidade ao longo de múltiplas rodadas. A média representa o comportamento esperado do modelo em diferentes partições. O pico indica a melhor configuração observada, mas deve ser interpretado com cautela, sobretudo em uma base de escala limitada. O desvio entre rodadas fornece uma noção de variabilidade estatística.

Como auditoria complementar, foi empregado um teste cego por amostras previamente retidas durante o desenvolvimento experimental. Nesse procedimento, uma amostra normotensa e uma amostra da classe hipertensa eram removidas antes de qualquer ajuste estatístico aprendido, balanceamento sintético e treinamento. A extração de características, por ser determinística e aplicada individualmente a cada série RR, não utilizava informações globais das classes. Todo o fluxo aprendível era então executado sem acesso a essas duas amostras, que eram utilizadas apenas ao final como observações inéditas. Esse processo foi repetido tipicamente entre 50 e 100 rodadas, com seleção aleatória das amostras retidas em cada repetição. O Apêndice D apresenta uma implementação utilizada nessa etapa de auditoria.



Figura 3.7: Esquema da validação cega por amostras retidas. Uma amostra normotensa e uma amostra da classe hipertensa são removidas antes dos ajustes aprendidos; o treinamento ocorre sem acesso a essas amostras, que são avaliadas apenas ao final como observações inéditas.

Fonte: elaboração própria.

Esse teste não substituiu as métricas finais obtidas por múltiplas partições treino-teste, nem foi tratado como estimador estatístico principal de desempenho, mas funcionou como critério de auditoria metodológica. Configurações que apresentavam boa acurácia no teste convencional, mas desempenho incoerente ou instável nas amostras cegas, foram descartadas antes de serem promovidas a marcos consolidados. Dessa forma, várias tentativas intermediárias não foram nomeadas como versões de referência, pois não demonstraram consistência quando avaliadas em amostras tratadas como novidade completa.

O custo computacional também foi registrado. Essa métrica é particularmente importante no *benchmarking* entre modelos clássicos e quânticos simulados, pois os simuladores de vetor de estado apresentam custo crescente com o número de *qubits* e com o tamanho da matriz de kernel. Assim, a comparação final considera não apenas acurácia, mas também tempo de execução, número de características, complexidade do modelo e viabilidade de transição futura para hardware físico.

Capítulo 4

Resultados, *Benchmarking* e Discussão

Este capítulo apresenta e discute os resultados obtidos durante o desenvolvimento dos modelos clássicos e quânticos híbridos. A análise é conduzida em quatro etapas. Primeiramente, estabelece-se o desempenho do *pipeline* clássico de referência, que representa o teto empírico obtido por força estatística e ampla engenharia de características. Em seguida, descreve-se a evolução dos experimentos quânticos, destacando acertos, falhas e fenômenos observados durante o processo de codificação dos dados no Espaço de Hilbert. Na sequência, analisa-se em maior profundidade o modelo quântico final, denominado Q-V14, que consolidou as decisões arquiteturais mais estáveis. Por fim, realiza-se o *benchmarking* comparativo entre as abordagens, discutindo eficiência dimensional, custo computacional e limites de interpretação.

4.1 Desempenho do *Benchmark* Clássico

O *benchmark* clássico foi construído para representar uma linha de base forte e competitiva. Após ciclos sucessivos de experimentação e consolidação metodológica, o modelo clássico final utilizou um comitê de votação suave composto por *Extra Trees* e XGBoost, totalizando aproximadamente 4000 árvores de decisão quando considerados os estimadores internos das duas arquiteturas. Essa configuração foi acompanhada de ampla engenharia de características e regularização para reduzir o risco de sobreajuste.

O desempenho final do modelo clássico consolidado foi de **83.73% de acurácia média** em 100 rodadas estocásticas, com **91.67% de acurácia de pico**. O *recall* médio dos normotensos foi de **69.85%**, enquanto o *recall* médio da classe hipertensa em intervenção farmacológica foi de **90.68%**. O tempo de execução foi de aproximadamente 13.67 minutos para as 100 rodadas, evidenciando que, apesar da complexidade do *ensemble*, a infraestrutura clássica atual processa esse tipo de experimento com relativa eficiência.

Esses valores definem o teto empírico clássico do conjunto de dados. A expressão “teto” não significa que seja matematicamente impossível superar esse desempenho, mas que, dentro das condições experimentais adotadas, o uso de dezenas de características e

um comitê robusto estabilizou a acurácia média em torno de 84%. Dessa forma, qualquer comparação com modelos quânticos precisa considerar esse patamar como referência.

A Tabela 4.1 apresenta a síntese do modelo clássico final.

Tabela 4.1: Resultado consolidado do *benchmark* clássico de referência.

Indicador	Resultado
Arquitetura	<i>Voting Classifier: Extra Trees</i> + XGBoost
Número aproximado de árvores	4000 estimadores
Acurácia média	83.73%
Acurácia de pico	91.67%
<i>Recall</i> médio normotenso	69.85%
<i>Recall</i> médio da classe hipertensa em intervenção farmacológica	90.68%
Rodadas estocásticas	100
Tempo total aproximado	13.67 minutos

A Figura 4.1 sintetiza visualmente a diferença de desempenho por classe, destacando que a classe hipertensa em intervenção farmacológica apresentou maior *recall* médio tanto no modelo clássico quanto no modelo quântico final.

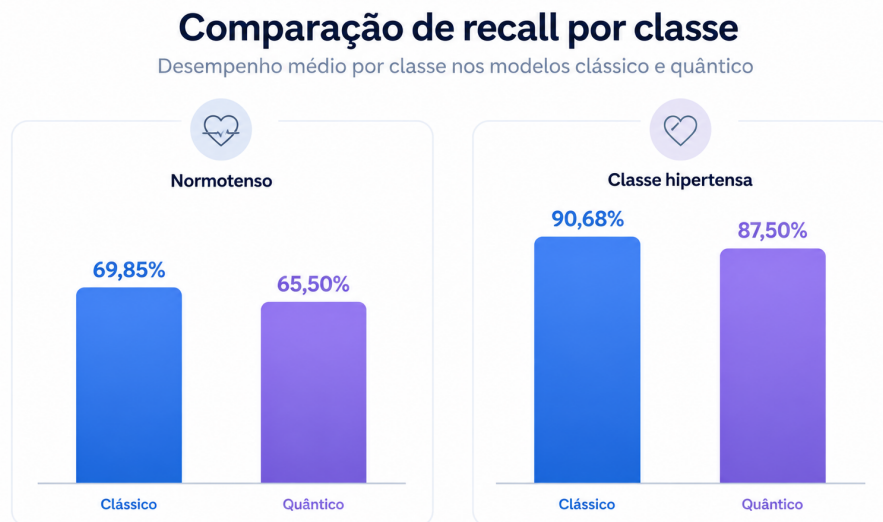


Figura 4.1: Comparação do *recall* médio por classe entre o *benchmark* clássico e o modelo quântico híbrido final. A figura evidencia a maior facilidade dos modelos em reconhecer a classe hipertensa em intervenção farmacológica, bem como a maior dificuldade relativa de identificação da classe normotensa.

Fonte: elaboração própria.

Do ponto de vista interpretativo, o modelo clássico mostrou elevada capacidade de detecção da classe hipertensa em intervenção farmacológica, mas manteve desempenho inferior na classe normotensa. Esse comportamento é coerente com problemas biomé-

dicos desbalanceados ou fisiologicamente assimétricos, nos quais a classe clínica pode apresentar padrões mais marcados nos descritores, enquanto o grupo controle contém maior variabilidade individual.

A evolução histórica do *pipeline* clássico demonstrou que o ganho de desempenho não foi obtido por um único atributo, mas pela combinação progressiva de famílias de características. As configurações iniciais consolidadas, baseadas em conjuntos compactos, permaneceram próximas de 70% de média. A inclusão de métricas fractais, entrópicas, TKEO, fragmentação e entropia multiescala elevou gradualmente o desempenho. Um dos marcos finais de desenvolvimento, baseado em otimização multiescala e maior número de estimadores, atingiu média de 85.48% no histórico experimental. Entretanto, o teste de estresse clássico final, alinhado ao código apresentado no Apêndice B, estabilizou a média em 83.73% sob protocolo de 100 rodadas.

Essa diferença entre configurações reforça que o desempenho depende do protocolo de validação. Resultados obtidos em menos rodadas ou partições específicas podem apresentar médias mais altas ou picos mais expressivos. Por isso, a análise final prioriza o teste de estresse com maior número de execuções, que fornece estimativa mais conservadora e estável do desempenho clássico.

4.2 Evolução dos Modelos Quânticos

A evolução dos modelos quânticos foi marcada por hipóteses experimentais sucessivas, das quais foram preservados no texto apenas os marcos que produziram aprendizado metodológico relevante. A numeração das configurações não representa todas as tentativas realizadas, mas somente os pontos consolidados após testes, falhas, auditorias e comparações. Parte das configurações intermediárias foi descartada antes de receber denominação formal, sobretudo quando apresentava boa acurácia em partições convencionais, mas comportamento instável nos testes cegos com amostras previamente retidas. Esse procedimento auxiliar é documentado no Apêndice D.

Durante as simulações preliminares, uma configuração reportou acurácia anormalmente elevada. A auditoria posterior identificou vazamento de informação na matriz de entrada, causado pela presença indevida de índices de controle. Esse episódio foi metodologicamente relevante porque demonstrou que modelos de alta capacidade, clássicos ou quânticos, podem explorar artefatos não fisiológicos. Após a remoção desses elementos, os resultados passaram a refletir de forma mais honesta a capacidade dos circuitos avaliados.

4.2.1 Compressão Linear e Perda de Informação

O marco Q-V1 utilizou QSVC com ZZFeatureMap em 4 *qubits*, combinada com PCA para reduzir a dimensionalidade da matriz clássica. O resultado foi uma acurácia

global de 61.67%, com *recall* normotenso de 30.00% e *recall* da classe hipertensa de 78.00%. A interpretação adotada é que o PCA, embora útil em muitos cenários clássicos, atuou como filtro destrutivo para relações não-lineares das séries RR.

A Q-V2 substituiu o PCA por seleção direta das quatro características mais relevantes. A acurácia global permaneceu em 61.67%, mas o *recall* normotenso subiu para 55.00%, enquanto o *recall* da classe hipertensa ficou em 65.00%. Esse resultado mostrou que a injeção direta preservou melhor o equilíbrio entre classes, ainda que a limitação de quatro variáveis impedisse desempenho elevado.

4.2.2 Ajuste Angular e Geometria da Esfera de Bloch

A Q-V3 ajustou o intervalo de codificação de 0 a 2π para 0 a π , com emaranhamento completo e uma repetição do *Feature Map*. O resultado foi um salto para 72.56% de média e 78.33% de pico. A Q-V4 avançou essa hipótese ao comprimir os dados para 0 a $\pi/2$, utilizando topologia circular e ajuste de pesos no SVM. O desempenho médio subiu para 76.78%, com pico de 83.33%. Esses marcos mostraram que a geometria do mapeamento angular era determinante para preservar a relação entre valores clássicos e estados quânticos.

4.2.3 Expressividade versus Estabilidade

O marco Q-V5.1 testou *PauliFeatureMap* com portas Z e YY , duas repetições e emaranhamento completo. A média global caiu para 74.67%, mas o pico atingiu 86.67%. Esse comportamento revelou um *trade-off*: aumentar expressividade pode permitir excelentes resultados em algumas partições, mas também elevar variância e instabilidade.

O marco Q-V6 tentou controlar essa instabilidade com *Dual Scaling* e margem suave ($C = 1.0$). O resultado foi um colapso para 62.89% de média. A conclusão foi que técnicas clássicas de normalização e suavização de margem não se transferem automaticamente para o domínio quântico. Ao reduzir demais a dispersão geométrica dos dados, o circuito gerou estados menos distinguíveis.

4.2.4 Expansão Lateral e Redundância Inspirada em *Data Re-uploading*

O marco Q-V7 introduziu uma das estratégias mais importantes da pesquisa: duplicar lateralmente as quatro características de elite para alimentar oito *qubits*, em uma construção inspirada em *Data Re-uploading*. Com *ZZFeatureMap*, uma repetição e emaranhamento circular, a média global subiu para 78.83%, com pico de 86.67%. Nenhuma informação clássica nova foi adicionada; o ganho veio da redistribuição da mesma informação em um espaço de Hilbert maior.

O marco Q-V8 testou a hipótese alternativa: usar oito características distintas em oito *qubits*. A média caiu para 74.00%, com pico de 80.00%. Esse resultado é contraintuitivo do ponto de vista clássico, mas coerente com a interpretação quântica: muitas variáveis ruidosas distintas em um circuito raso podem produzir interferência destrutiva, enquanto a repetição estruturada de variáveis fortes pode produzir separabilidade geométrica mais estável.

O marco Q-V9 avaliou *Re-uploading* não-linear com termos quadráticos. A média foi de 76.00%, inferior à Q-V7, sugerindo que o `ZZFeatureMap` já introduz não-linearidade suficiente e que transformações polinomiais prévias podem distorcer a fase.

4.2.5 Profundidade e Colapso de Desempenho

O marco Q-V10 testou a hipótese de que aumentar profundidade seria mais vantajoso que aumentar largura. Com 4 *qubits*, três repetições e emaranhamento completo, a acurácia média caiu para 51.22%, valor próximo de uma decisão aleatória balanceada e inferior ao que seria esperado de uma regra ingênua baseada apenas na classe majoritária. Esse resultado mostrou que a intuição clássica de que modelos mais profundos tendem a extrair representações melhores não se aplica diretamente a circuitos quânticos com mapeamento angular amplo.

O colapso foi interpretado principalmente como efeito de saturação geométrica do mapeamento. A associação com Platôs Estéreis deve ser entendida apenas como analogia conceitual com fenômenos de concentração e perda de informatividade em circuitos parametrizados profundos, e não como diagnóstico formal de uma paisagem variacional treinável. O excesso de camadas e fases fez com que diferenças fisiológicas fossem dissolvidas em interferências pouco informativas. A partir desse resultado, consolidou-se a diretriz de priorizar circuitos rasos.

4.2.6 Síntese Numérica dos Principais Marcos

A Tabela 4.2 resume os resultados principais dos marcos quânticos discutidos.

A Figura 4.2 apresenta a evolução dos principais marcos quânticos, evidenciando que a melhoria de desempenho não ocorreu por aumento linear de complexidade, mas por ajustes específicos de codificação, largura, topologia e redundância lateral.

A evolução evidencia que o melhor desempenho não foi obtido por aumento indiscriminado de complexidade. O modelo final surgiu da combinação de três princípios: seleção de poucas características densas, expansão lateral inspirada em *Data Re-uploading* e controle clássico do hiperplano por pesos assimétricos.

Tabela 4.2: Evolução resumida dos principais marcos quânticos híbridos.

Marco	Estratégia principal	Média	Pico
Q-V1	PCA + 4 <i>qubits</i>	61.67%	–
Q-V2	Seleção direta de 4 atributos	61.67%	–
Q-V3	0 a π , emaranhamento completo	72.56%	78.33%
Q-V4	0 a $\pi/2$, topologia circular	76.78%	83.33%
Q-V5.1	PauliFeatureMap expressivo	74.67%	86.67%
Q-V6	<i>Dual Scaling</i> e margem suave	62.89%	71.67%
Q-V7	8 <i>qubits</i> com redundância lateral	78.83%	86.67%
Q-V8	8 atributos nativos	74.00%	80.00%
Q-V9	<i>Re-uploading</i> não-linear	76.00%	80.00%
Q-V10	Profundidade elevada	51.22%	56.67%
Q-V13	Pesos manuais e margem ajustada	70.78%	83.33%
Q-V14	8 <i>qubits</i> , redundância lateral e pesos manuais	80.17%	90.00%



Figura 4.2: Evolução dos principais marcos quânticos híbridos avaliados ao longo do desenvolvimento experimental. A queda observada no marco Q-V10 ilustra que maior profundidade de circuito não implicou melhor separabilidade para as séries RR analisadas.

Fonte: elaboração própria.

4.3 Desempenho do Modelo Quântico Híbrido Final

O modelo quântico híbrido final, mantido no registro experimental como Q-V14, consolidou as principais lições dos marcos anteriores. Sua arquitetura utilizou QSVC com ZZFeatureMap em 8 *qubits*, uma repetição e emaranhamento circular, aplicando redundância lateral inspirada em *Data Re-uploading* às quatro características mais relevantes. O vetor original de entrada foi formado por descritores compactos de dinâmica não-linear e energia das séries RR. Esse vetor foi duplicado horizontalmente para ocupar oito dimensões, permitindo que a mesma informação fosse projetada em um Espaço de Hilbert de 256 dimensões.

O mapeamento angular foi restrito ao intervalo de 0 a 2 radianos. Essa escolha buscou evitar tanto a compressão excessiva quanto o *wrap-around* completo associado a intervalos como 0 a 2π . Além disso, foram aplicados pesos manuais assimétricos no SVM, atribuindo maior importância à classe normotensa. Essa decisão foi motivada pelos marcos anteriores, nos quais o modelo frequentemente favorecia a classe hipertensa em intervenção farmacológica.

O desempenho obtido pela Q-V14 foi de **80.17% de acurácia média**, com **90.00% de acurácia de pico**. O *recall* médio dos normotensos foi de **65.50%**, enquanto o *recall* médio da classe hipertensa em intervenção farmacológica foi de **87.50%**. Na melhor matriz de confusão observada, o modelo classificou corretamente 16 dos 20 normotensos e 38 dos 40 indivíduos da classe hipertensa, correspondendo a 80.00% de *recall* normotenso e 95.00% de *recall* da classe hipertensa.

A Tabela 4.3 apresenta a síntese do modelo final.

Tabela 4.3: Resultado consolidado do modelo quântico híbrido Q-V14.

Indicador	Resultado
Modelo	QSVC com ZZFeatureMap
Número de <i>qubits</i>	8
Profundidade lógica	1 repetição
Emaranhamento	Circular
Estratégia de entrada	4 características + redundância lateral
Acurácia média	80.17%
Acurácia de pico	90.00%
<i>Recall</i> médio normotenso	65.50%
<i>Recall</i> médio da classe hipertensa em intervenção farmacológica	87.50%
Tempo aproximado	64.15 minutos para 10 rodadas

A Figura 4.3 apresenta a matriz de confusão do melhor caso observado para o modelo Q-V14. Essa visualização complementa a leitura das médias, pois mostra concretamente que, na rodada de pico, o modelo reconheceu 16 das 20 amostras normotensas e 38 das 40

amostras da classe hipertensa em intervenção farmacológica.

Matriz de Confusão: Modelo Quântico V14

Rótulo Verdadeiro	Normotenso	16	4
	Hipertenso	2	38
		Normotenso	Hipertenso
		Rótulo Predito	

Figura 4.3: Matriz de confusão do melhor caso observado para o modelo quântico híbrido Q-V14. A matriz evidencia maior acerto na classe hipertensa em intervenção farmacológica e desempenho inferior, embora ainda relevante, na classe normotensa.

Fonte: elaboração própria.

A interpretação mais importante desse resultado é que o modelo quântico aproximou-se do *baseline* clássico utilizando uma representação muito mais compacta. Enquanto o modelo clássico explorou dezenas de características e milhares de estimadores, a Q-V14 utilizou apenas quatro descritores principais, conforme o código do Apêndice C, redistribuídos em oito *qubits* por redundância lateral. Isso não autoriza uma conclusão absoluta sobre vantagem quântica, mas evidencia competitividade experimental em eficiência dimensional.

O custo temporal da Q-V14 foi superior ao do modelo clássico quando considerado o número de rodadas executadas. Esse ponto é essencial: a simulação clássica de circuitos quânticos não deve ser confundida com a execução em hardware quântico. O alto custo observado decorre do cálculo da matriz de kernel por simulador de vetor de estado. Em uma QPU física, o custo teria outra natureza, envolvendo número de medições, ruído, profundidade e conectividade. Portanto, o tempo de simulação é uma limitação experimental do estudo, não necessariamente uma propriedade definitiva do algoritmo.

Do ponto de vista dos padrões derivados da atividade elétrica cardíaca, o modelo apresentou maior facilidade em reconhecer a classe hipertensa em intervenção farmacológica do que a classe normotensa. Esse comportamento também ocorreu no *baseline* clássico e sugere que a classe hipertensa possui assinatura mais separável no espaço de atributos utilizado. A melhora do *recall* normotenso na matriz de pico mostra, contudo, que o ajuste de pesos e a geometria da redundância lateral foram capazes de reduzir parcialmente esse

viés.

4.4 Discussão: Eficiência Dimensional e Limites do *Benchmarking*

A comparação entre o modelo clássico e o modelo quântico final deve ser interpretada para além da acurácia. O modelo clássico alcançou 83.73% de média e 91.67% de pico, enquanto a Q-V14 alcançou 80.17% de média e 90.00% de pico. Em termos absolutos, o modelo clássico permaneceu superior na média. Entretanto, a diferença de desempenho é relativamente pequena quando comparada à diferença de complexidade informacional entre as abordagens.

O modelo clássico utilizou dezenas de características e um comitê com milhares de árvores. Essa estratégia é poderosa, mas depende de ampla engenharia de atributos e de uma infraestrutura estatística pesada. O modelo quântico, por sua vez, utilizou quatro características principais e um circuito raso de oito *qubits*. A proximidade entre os resultados sugere que o mapeamento quântico foi capaz de extrair separabilidade significativa a partir de um vetor de entrada compacto.

Tabela 4.4: Comparação entre o *benchmark* clássico e o modelo quântico final.

Critério	Modelo clássico	Modelo quântico Q-V14
Arquitetura	<i>Extra Trees</i> + XGBoost	QSVC + ZZFeatureMap
Entrada	Dezenas de características	4 características com redundância lateral
Estrutura de decisão	Comitê com milhares de árvores	Kernel quântico + SVM
Acurácia média	83.73%	80.17%
Acurácia de pico	91.67%	90.00%
<i>Recall</i> médio da classe hipertensa em intervenção farmacológica	90.68%	87.50%
<i>Recall</i> médio normotenso	69.85%	65.50%
Custo temporal observado	13.67 min / 100 rodadas	64.15 min / 10 rodadas
Interpretação principal	Força estatística e engenharia ampla	Eficiência dimensional em simulação

A Figura 4.4 apresenta a comparação visual entre acurácia média e acurácia de pico dos dois modelos finais. A visualização reforça que ambos ultrapassaram 80% de média e atingiram ou superaram 90% de pico no contexto experimental adotado.

O conceito de eficiência dimensional, neste trabalho, refere-se à capacidade de alcançar desempenho competitivo utilizando menos características de entrada. A Q-V14 não apresentou a maior acurácia média, mas demonstrou que uma representação quântica

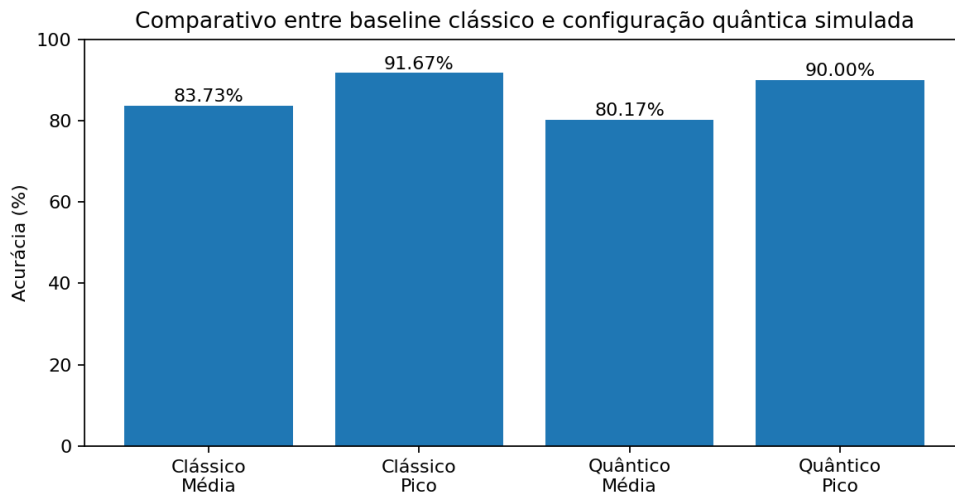


Figura 4.4: Comparação entre o desempenho do *benchmark* clássico e do modelo quântico híbrido final. O gráfico destaca a proximidade entre os resultados de pico e a diferença moderada entre as acurácias médias, sustentando a leitura de competitividade experimental em simulação ideal.

Fonte: elaboração própria.

bem calibrada pode se aproximar do teto clássico com uma fração do vetor informacional. Esse resultado é relevante porque problemas biomédicos frequentemente sofrem com dimensionalidade elevada, redundância de atributos e dificuldade de interpretação.

A interpretação dos resultados, contudo, deve considerar a escala reduzida da base experimental. Em conjuntos biomédicos pequenos, pequenas alterações na composição das partições podem produzir variações perceptíveis nas métricas finais, especialmente em problemas com classes desbalanceadas. Por essa razão, os valores médios ao longo de múltiplas rodadas são mais informativos do que picos isolados de acurácia. O teste cego por amostras retidas também contribuiu para essa leitura: configurações promissoras em teste convencional, mas frágeis diante de amostras completamente reservadas, foram descartadas antes de integrarem a sequência de marcos consolidados. Por ter caráter complementar e escala reduzida, essa auditoria foi utilizada como filtro de consistência, não como substituto das métricas médias reportadas.

A análise dos marcos quânticos revela que eficiência dimensional não surge automaticamente. A Q-V1, baseada em PCA, teve desempenho baixo. A Q-V8, com oito variáveis distintas, foi inferior à Q-V7, que duplicava lateralmente quatro variáveis fortes. A Q-V10, mais profunda, colapsou. Esses resultados mostram que a vantagem potencial do QML depende da geometria da codificação. Em séries RR ruidosas e biologicamente variáveis, preservar características não-lineares densas e distribuí-las adequadamente no circuito foi mais eficaz do que aumentar indiscriminadamente variáveis ou camadas.

Também é necessário reconhecer os limites do estudo. A execução ocorreu em simuladores ideais, sem ruído físico; portanto, não se pode afirmar que a Q-V14 manteria

80.17% de média em uma QPU real. Circuitos com emaranhamento circular em 8 *qubits* podem exigir conectividade específica e sofrer degradação por erros de porta. Além disso, o cálculo de kernels quânticos envolve matrizes de similaridade cujo custo cresce com o número de amostras, representando um desafio de escalabilidade.

Apesar dessas limitações, o resultado final responde positivamente à pergunta de pesquisa em um sentido técnico delimitado: *em simulação ideal, um modelo quântico híbrido raso, com codificação cuidadosamente ajustada, alcançou desempenho competitivo frente a um modelo clássico robusto, utilizando número substancialmente menor de características.* Esse é o principal achado do *benchmarking*.

Capítulo 5

Conclusões e Trabalhos Futuros

Este capítulo consolida as conclusões obtidas a partir do *benchmarking* entre aprendizado de máquina clássico e aprendizado de máquina quântico híbrido aplicado a sinais biomédicos derivados da atividade elétrica cardíaca. A discussão final retoma a pergunta central de pesquisa, avalia o cumprimento dos objetivos propostos e delimita o alcance das afirmações que podem ser feitas com base nos resultados experimentais. Em seguida, são apresentadas limitações metodológicas e tecnológicas, bem como propostas de continuidade para futuras investigações em *Quantum Machine Learning* aplicado a dados biomédicos.

5.1 Conclusões Gerais

O objetivo central deste trabalho foi avaliar a viabilidade técnica de algoritmos de *Quantum Machine Learning* aplicados à classificação de sinais biomédicos derivados da atividade elétrica cardíaca, realizando um *benchmarking* contra um modelo clássico robusto. A pesquisa foi motivada pela seguinte questão: circuitos quânticos híbridos rasos, compatíveis com as restrições conceituais da era NISQ, conseguem alcançar desempenho competitivo usando menos características do que modelos clássicos de alta capacidade?

A resposta obtida é afirmativa, desde que interpretada dentro dos limites experimentais do estudo. O modelo clássico final alcançou 83,73% de acurácia média e 91,67% de pico, utilizando dezenas de características e um comitê com milhares de árvores. O modelo quântico final Q-V14 alcançou 80,17% de média e 90,00% de pico, utilizando apenas quatro características principais, redistribuídas em oito *qubits* por redundância lateral inspirada em *Data Re-uploading*. Embora o modelo clássico tenha permanecido superior em média, o modelo quântico demonstrou competitividade preditiva com maior compactação dimensional.

Do ponto de vista biomédico, os resultados reforçam que séries de intervalos RR derivados do ECG carregam informação discriminativa para diferenciar padrões associados a indivíduos normotensos e à classe hipertensa em intervenção farmacológica. Métricas

não-lineares, entrópicas e energéticas mostraram-se especialmente importantes, indicando que a diferença entre classes não se limita a valores médios, mas envolve organização temporal, irregularidade e complexidade da atividade cardíaca. Essa interpretação deve considerar que o grupo clínico avaliado estava sob intervenção farmacológica e que a base possui escala limitada, o que restringe generalizações clínicas amplas.

Do ponto de vista clássico, a pesquisa demonstrou a eficácia de *ensembles* robustos em problemas tabulares biomédicos. O comitê baseado em *Extra Trees* e XGBoost estabeleceu um teto empírico elevado, mas exigiu grande número de estimadores e ampla engenharia de atributos. Esse resultado confirma que o paradigma clássico ainda é extremamente competitivo, eficiente em tempo de execução e mais maduro operacionalmente.

Do ponto de vista quântico, a principal contribuição foi mostrar que o desempenho do QSVC depende criticamente do *data embedding*. O uso de PCA degradou a informação não-linear; a injeção direta preservou melhor o equilíbrio entre classes; o ajuste angular melhorou o desempenho; o excesso de profundidade colapsou a separabilidade; e a duplicação lateral estruturada de características ampliou a expressividade geométrica sem adicionar novas variáveis.

Os objetivos específicos foram atendidos. Foram extraídas características relevantes de séries RR; foi implementado um modelo clássico forte; foram desenvolvidas e comparadas arquiteturas quânticas híbridas; foram analisadas topologias, profundidades, codificações e pesos de classe; e foi realizada uma comparação final considerando acurácia, *recall*, custo computacional e eficiência dimensional. Além disso, testes cegos por amostras retidas foram utilizados como auditoria complementar durante o desenvolvimento, conforme exemplificado no Apêndice D, contribuindo para descartar configurações instáveis antes da consolidação dos marcos discutidos.

A conclusão principal deste trabalho é que o *Quantum Machine Learning*, em simulação ideal, mostrou-se tecnicamente viável como abordagem experimental para classificação de dados biomédicos não-lineares derivados de séries RR. O resultado não deve ser interpretado como comprovação de vantagem quântica prática em hardware real, nem como validação clínica definitiva. O achado mais sólido é a demonstração de **paridade experimental aproximada e eficiência dimensional**: um circuito quântico híbrido raso conseguiu se aproximar do desempenho de um modelo clássico muito mais amplo, utilizando um conjunto reduzido de características.

A Figura 5.1 sintetiza a interpretação final do estudo, destacando a relação entre referência clássica, compactação quântica e limites de aplicação.

Essa conclusão aponta uma direção concreta para pesquisas futuras. Em vez de tratar a computação quântica como substituta imediata dos métodos clássicos, os resultados sugerem que ela pode atuar como mecanismo complementar de representação geométrica, especialmente em problemas nos quais poucas variáveis densas precisam ser projetadas em espaços de alta expressividade. O desafio passa a ser transformar essa viabilidade simulada



Figura 5.1: Síntese interpretativa dos resultados do trabalho. O modelo clássico estabeleceu uma referência robusta de desempenho, enquanto o modelo quântico híbrido Q-V14 demonstrou competitividade experimental com entrada dimensionalmente mais compacta. A interpretação permanece limitada ao contexto de simulação ideal e base de escala reduzida.

Fonte: elaboração própria.

em desempenho robusto sob ruído físico, bases maiores e protocolos independentes.

5.2 Limitações e Trabalhos Futuros

Embora os resultados sejam promissores, este trabalho possui limitações importantes. A primeira é a execução em simulador ideal baseado em vetor de estado, que permite avaliar a geometria matemática do circuito sem ruído, mas não reproduz integralmente uma QPU física, sujeita a erros de porta, decoerência, ruído de leitura, conectividade limitada e múltiplas medições.

A segunda limitação está relacionada à escala dos dados. A base permitiu um *benchmark* controlado entre processamento clássico e aprendizado quântico híbrido, mas seu tamanho amostral reduzido limita generalizações clínicas. Em bases pequenas, a composição das partições influencia as métricas, justificando múltiplas rodadas, análise média e auditorias contra vazamento. Estudos futuros devem incluir bases maiores, independentes e preferencialmente multicêntricas, com separações por indivíduo, condição clínica e regime farmacológico.

A terceira limitação envolve a interpretação biomédica. O modelo demonstrou capacidade de distinguir classes no conjunto avaliado, mas isso não equivale a um sistema de diagnóstico médico. Para aplicação clínica, seriam necessários estudos adicionais com

sensibilidade, especificidade, calibração probabilística, comparação com variáveis clínicas tradicionais e análise ética de uso.

A quarta limitação diz respeito aos metadados de aquisição das séries RR. As informações disponibilizadas para este estudo não detalhavam, de forma suficiente, condição postural, estado de repouso, padrão respiratório ou protocolo fisiológico completo de coleta. Como esses fatores podem influenciar a Variabilidade da Frequência Cardíaca, trabalhos futuros devem utilizar bases acompanhadas de metadados clínicos e experimentais mais completos, permitindo análises estratificadas e maior controle fisiológico.

Do ponto de vista técnico, pesquisas futuras devem executar arquiteturas simplificadas ou otimizadas a partir da Q-V14 em processadores quânticos reais, avaliando conectividade da QPU, transpilação, profundidade efetiva, portas adicionais e escolha de *backend*. Também devem comparar simulação ideal, simulação ruidosa e execução física, utilizando técnicas de mitigação de erros.

Também devem ser exploradas novas estratégias de codificação, como *Amplitude Encoding*, codificações híbridas e *Feature Maps* fisiologicamente informados. Além do QSVC, podem ser testados classificadores variacionais, redes neurais quânticas rasas, *quantum kernels* treináveis e modelos híbridos que combinem camadas clássicas e quânticas.

Por fim, a validação estatística deve ser ampliada. Este trabalho utilizou múltiplas rodadas e testes cegos por amostras retidas como auditoria complementar; estudos futuros devem expandir esse procedimento para bases maiores e independentes, incluindo intervalos de confiança, testes de hipótese, F1-score macro, AUC, especificidade e sensibilidade.

Referências

- 1 Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology. Heart rate variability: standards of measurement, physiological interpretation and clinical use. *Circulation*, v. 93, n. 5, p. 1043–1065, 1996.
- 2 SHAFFER, F.; GINSBERG, J. P. An overview of heart rate variability metrics and norms. *Frontiers in Public Health*, v. 5, p. 258, 2017.
- 3 RICHMAN, J. S.; MOORMAN, J. R. Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, v. 278, n. 6, p. H2039–H2049, 2000.
- 4 BANDT, C.; POMPE, B. Permutation entropy: a natural complexity measure for time series. *Physical Review Letters*, v. 88, n. 17, p. 174102, 2002.
- 5 COSTA, M.; GOLDBERGER, A. L.; PENG, C.-K. Multiscale entropy analysis of complex physiologic time series. *Physical Review Letters*, v. 89, n. 6, p. 068102, 2002.
- 6 PENG, C.-K. et al. Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series. *Chaos*, v. 5, n. 1, p. 82–87, 1995.
- 7 HIGUCHI, T. Approach to an irregular time series on the basis of the fractal theory. *Physica D: Nonlinear Phenomena*, v. 31, n. 2, p. 277–283, 1988.
- 8 GOLDBERGER, A. L. et al. Fractal dynamics in physiology: Alterations with disease and aging. *Proceedings of the National Academy of Sciences*, v. 99, n. suppl. 1, p. 2466–2472, 2002.
- 9 BIAMONTE, J. et al. Quantum machine learning. *Nature*, v. 549, n. 7671, p. 195–202, 2017.
- 10 HAVLÍČEK, V. et al. Supervised learning with quantum-enhanced feature spaces. *Nature*, v. 567, n. 7747, p. 209–212, 2019.
- 11 PRESKILL, J. Quantum computing in the nisq era and beyond. *Quantum*, v. 2, p. 79, 2018.
- 12 GOLDBERGER, A. L. et al. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, v. 101, n. 23, p. e215–e220, 2000.
- 13 MOODY, G. B.; MARK, R. G. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, v. 20, n. 3, p. 45–50, 2001.

- 14 BRENNAN, M.; PALANISWAMI, M.; KAMEN, P. Do existing measures of poincaré plot geometry reflect nonlinear features of heart rate variability? *IEEE Transactions on Biomedical Engineering*, v. 48, n. 11, p. 1342–1347, 2001.
- 15 KAISER, J. F. On a simple algorithm to calculate the energy of a signal. In: *IEEE. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. [S.l.], 1990. p. 381–384.
- 16 HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2. ed. New York: Springer, 2009.
- 17 GEURTS, P.; ERNST, D.; WEHENKEL, L. Extremely randomized trees. *Machine Learning*, v. 63, n. 1, p. 3–42, 2006.
- 18 CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [S.l.]: ACM, 2016. p. 785–794.
- 19 CHAWLA, N. V. et al. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, v. 16, p. 321–357, 2002.
- 20 HAN, H.; WANG, W.-Y.; MAO, B.-H. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In: *Advances in Intelligent Computing*. Berlin, Heidelberg: Springer, 2005. (Lecture Notes in Computer Science, v. 3644), p. 878–887.
- 21 CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, v. 20, n. 3, p. 273–297, 1995.
- 22 ABRAHAM, H. et al. Qiskit: An open-source framework for quantum computing. *Zenodo*, 2019.
- 23 MCCLEAN, J. R. et al. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, v. 9, n. 1, p. 4812, 2018.
- 24 PÉREZ-SALINAS, A. et al. Data re-uploading for a universal quantum classifier. *Quantum*, v. 4, p. 226, 2020.
- 25 HJORTH, B. Eeg analysis based on time domain properties. *Electroencephalography and Clinical Neurophysiology*, v. 29, n. 3, p. 306–310, 1970.
- 26 JOLLIFFE, I. T. *Principal Component Analysis*. 2. ed. New York: Springer, 2002.

Apêndice A

Parâmetros de reprodutibilidade dos experimentos

Este apêndice apresenta os principais elementos necessários para a reprodução dos experimentos finais descritos neste trabalho. Os códigos-fonte completos dos modelos finais avaliados são apresentados nos Apêndices B e C. O Apêndice D apresenta uma implementação auxiliar da validação cega por amostras retidas, utilizada como auditoria complementar durante o desenvolvimento experimental.

A base experimental utilizada nos dois experimentos finais foi carregada a partir do arquivo `resultados_v3.xlsx`. A matriz utilizada nas implementações é composta por 66 séries de indivíduos normotensos e 133 séries da classe hipertensa em intervenção farmacológica, contendo 256 pontos de intervalos RR derivados do ECG por série. Dessa forma, a comparação realizada neste trabalho não representa uma separação entre indivíduos saudáveis e indivíduos com hipertensão arterial sem intervenção farmacológica, mas entre indivíduos normotensos e indivíduos com hipertensão arterial em intervenção farmacológica.

Essa distinção é relevante para a interpretação dos resultados, pois o tratamento farmacológico tende a reduzir diferenças fisiológicas observáveis entre os grupos. Em um cenário ideal, no qual a intervenção farmacológica normalizasse completamente a dinâmica da atividade cardíaca desses indivíduos, a separabilidade entre as classes tenderia a diminuir significativamente. Assim, a identificação de padrões discriminativos em séries RR da classe sob intervenção farmacológica representa um problema experimental mais sutil do que a distinção direta entre indivíduos normotensos e indivíduos com hipertensão arterial sem intervenção farmacológica.

A Tabela A.1 resume os principais elementos de reprodutibilidade considerados nos experimentos finais.

Tabela A.1: Síntese de reprodutibilidade dos experimentos finais.

Elemento	Descrição
Base de dados	Arquivo <code>resultados_v3.xlsx</code> , carregado diretamente nas implementações.
Composição experimental	66 séries normotensas e 133 séries da classe hipertensa em intervenção farmacológica.
Tamanho de cada série	256 pontos de intervalos RR derivados do ECG.
Modelo clássico final	<i>Voting Classifier</i> composto por <i>Extra Trees</i> e XGBoost, avaliado em 100 rodadas estocásticas.
Modelo quântico híbrido final	QSVC com ZZFeatureMap, 8 <i>qubits</i> , emaranhamento circular e redundância lateral inspirada em <i>Data Re-uploading</i> .
Auditoria complementar	Teste cego por amostras retidas, com remoção de uma amostra normotensa e uma da classe hipertensa em intervenção farmacológica antes dos ajustes aprendidos, do balanceamento e do treinamento.
Controle contra vazamento de dados	Escalonadores e técnica de balanceamento ajustados exclusivamente nas partições de treino.

Apêndice B

Código-fonte do benchmark clássico final

Este apêndice apresenta o código-fonte utilizado para executar o benchmark clássico final descrito na metodologia. A implementação contempla o carregamento da base de dados, a extração de descritores estatísticos, não-lineares, fractais, espectrais e tempo-frequência, o tratamento de valores inválidos, o escalonamento ajustado apenas sobre a partição de treino, o balanceamento por *BorderlineSMOTE*, o treinamento do comitê formado por *Extra Trees* e XGBoost, e a avaliação em 100 rodadas estocásticas. Os arquivos .py correspondentes acompanham a versão digital deste trabalho para facilitar reprodução e execução fora do PDF.

```
1 # =====
2 # BENCHMARK CLASSICO FINAL - ENSEMBLE SUPERVISIONADO
3 # =====
4 !pip install --quiet openpyxl imbalanced-learn tqdm xgboost seaborn PyWavelets
5
6 import os, urllib.request, warnings, math, time
7 import numpy as np
8 import pandas as pd
9 import scipy.stats
10 import scipy.signal
11 import matplotlib.pyplot as plt
12 import seaborn as sns
13 from tqdm.notebook import tqdm
14
15 from sklearn.model_selection import train_test_split
16 from sklearn.ensemble import ExtraTreesClassifier, VotingClassifier
17 from xgboost import XGBClassifier
18 from sklearn.preprocessing import StandardScaler
19 from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
20 from imblearn.over_sampling import BorderlineSMOTE
21
22 try:
23     import pywt
24     _PYWT_AVAILABLE = True
```

```

25 except ImportError:
26     _PYWT_AVAILABLE = False
27
28 warnings.filterwarnings('ignore')
29
30 # 1. CARREGAMENTO DOS DADOS
31 url = "https://github.com/rafaelgov95/data_set-h-n_v1/raw/refs/heads/main/
      resultados_v3.xlsx"
32 file_path = "resultados.xlsx"
33 if not os.path.exists(file_path): urllib.request.urlretrieve(url, file_path)
34
35 res = pd.read_excel(file_path, header=None).values
36 RR0, RR1 = res[1:67, 1:257].astype(np.float32), res[67:200, 1:257].astype(np.float32)
37 X_raw = np.concatenate([RR0, RR1], axis=0)
38 y_total = np.concatenate([np.zeros(len(RR0)), np.ones(len(RR1))], axis=0)
39
40 # =====
41 # 2. ENGENHARIA DE CARACTERISTICAS
42 # =====
43 def gini(array):
44     array = np.sort(np.array(array, dtype=np.float64))
45     n = array.shape[0]; index = np.arange(1, n + 1)
46     if n == 0 or np.sum(array) == 0: return 0.0
47     return ((np.sum((2 * index - n - 1) * array)) / (n * np.sum(array)))
48
49 def fast_sampen(x, m=2, r=0.2):
50     std_x = np.std(x)
51     if std_x == 0 or len(x) < m + 2: return 0.0
52     r_tolerance = r * std_x; N = len(x)
53     def _phi(m_len):
54         if N < m_len + 1: return 0.0
55         shape = (N - m_len + 1, m_len)
56         X = np.lib.stride_tricks.as_strided(x, shape=shape, strides=(x.itemsize, x.
      itemsize))
57         dist = np.max(np.abs(X[:, None, :] - X[None, :, :]), axis=2)
58         np.fill_diagonal(dist, np.inf)
59         count = np.sum(dist <= r_tolerance)
60         num_pairs = shape[0] * (shape[0] - 1)
61         return count / num_pairs if num_pairs > 0 else 0.0
62     p1, p2 = _phi(m), _phi(m+1)
63     if p1 > 0 and p2 > 0: return -np.log(p2 / p1)
64     return 0.0
65
66 def fast_dfa(x):
67     N = len(x)
68     if N < 10: return 0.0
69     y = np.cumsum(x - np.mean(x)); max_scale_val = N // 4

```

```

70     if max_scale_val < 4: return 0.0
71     scales = np.unique(np.logspace(np.log10(4), np.log10(max_scale_val), 10).astype(
72         int))
73     scales = scales[scales >= 4]
74     if len(scales) < 2: return 0.0
75     F, actual_scales = [], []
76     for s in scales:
77         segments = N // s
78         if segments == 0: continue
79         y_cut = y[:segments * s].reshape((segments, s)); x_axis = np.arange(s);
80         rms_values = []
81         for i in range(segments):
82             if np.std(y_cut[i]) < 1e-10: rms_values.append(0.0)
83             else:
84                 coeffs = np.polyfit(x_axis, y_cut[i], 1)
85                 fit_line = np.polyval(coeffs, x_axis)
86                 rms_values.append(np.sqrt(np.mean((y_cut[i] - fit_line)**2)))
87         mean_rms = np.mean(rms_values)
88         if mean_rms > 1e-10: F.append(mean_rms); actual_scales.append(s)
89     if len(np.log(actual_scales)) > 1 and len(np.log(F)) > 1:
90         coeffs = np.polyfit(np.log(actual_scales), np.log(F), 1)
91         return coeffs[0]
92     return 0.0
93
94 def hjorth_mobility(x):
95     if len(x) < 2 or np.var(x) == 0: return 0.0
96     return np.sqrt(np.var(np.diff(x)) / np.var(x))
97
98 def hjorth_complexity(x):
99     if len(x) < 3: return 0.0
100     mob_x = hjorth_mobility(x)
101     if mob_x == 0: return 0.0
102     return hjorth_mobility(np.diff(x)) / mob_x
103
104 def higuchi_fd(x, k_max=8):
105     n_times = x.size
106     if n_times < 3: return 0.0
107     L = np.zeros(k_max)
108     for k in range(1, k_max + 1):
109         Lk = 0
110         for m_idx in range(k):
111             indices = np.arange(m_idx, n_times, k)
112             if len(indices) < 2: continue
113             sub = x[indices]
114             Lk += np.sum(np.abs(np.diff(sub))) * ((n_times - 1) / (k * (len(sub) - 1)))
115         )
116     L[k-1] = Lk / k if k > 0 else 0

```

```

114 k_values = np.arange(1, k_max + 1)
115 valid_indices = np.where(L > 0)[0]
116 if len(valid_indices) < 2: return 0.0
117 coeffs = np.polyfit(np.log(k_values[valid_indices]), np.log(L[valid_indices]), 1)
118 return max(0.0, -coeffs[0])
119
120 def teager_kaiser_energy(x):
121     if len(x) < 3: return np.array([0.0])
122     return x[1:-1]**2 - x[:-2] * x[2:]
123
124 def _coarse_grain(x, scale):
125     N = len(x)
126     if N < scale: return np.array([])
127     return np.mean(x[:N - (N % scale)].reshape(-1, scale), axis=1)
128
129 def multiscale_sampen(x, max_scale=10, m=2, r=0.2):
130     mse_values = []
131     if len(x) < m + 2 or np.std(x) == 0: return np.zeros(max_scale)
132     r_tol = r * np.std(x)
133     for scale in range(1, max_scale + 1):
134         cg = _coarse_grain(x, scale)
135         if len(cg) < m + 2 or np.std(cg) == 0: mse_values.append(0.0); continue
136         mse_values.append(fast_sampen(cg, m=m, r=(r_tol / np.std(cg))))
137     return np.array(mse_values)
138
139 def extract_wpd_features(rr, wavelet_name='db4', level=4):
140     feats = {}; num_bands = 2 ** level
141     if not _PYWT_AVAILABLE or len(rr) < num_bands:
142         for i in range(num_bands): feats[f'wpd_band_{i}_energy'] = 0.0; feats[f'
wpd_band_{i}_entropy'] = 0.0
143     return feats
144     try:
145         wp = pywt.WaveletPacket(data=rr, wavelet=wavelet_name, mode='symmetric',
maxlevel=level)
146         for i, node in enumerate(wp.get_level(level, order='natural')):
147             if len(node.data) > 1 and np.std(node.data) > 1e-10:
148                 feats[f'wpd_band_{i}_energy'] = np.sum(np.square(node.data))
149                 feats[f'wpd_band_{i}_entropy'] = scipy.stats.entropy(np.abs(node.data)
+ 1e-10)
150             else:
151                 feats[f'wpd_band_{i}_energy'] = 0.0; feats[f'wpd_band_{i}_entropy'] =
0.0
152     except:
153         for i in range(num_bands): feats[f'wpd_band_{i}_energy'] = 0.0; feats[f'
wpd_band_{i}_entropy'] = 0.0
154     return feats
155

```

```

156 def extract_dwt_features(rr, wavelet_name='db4', max_level=4):
157     feats = {}
158     if not _PYWT_AVAILABLE:
159         for i in range(1, max_level + 1): feats[f'dwt_detail_level_{i}_energy'] = 0.0
160         return feats
161     try:
162         coeffs = pywt.wavedec(rr, wavelet_name, level=max_level)
163         for i, detail_coeffs in enumerate(coeffs[1:]):
164             level = max_level - i
165             feats[f'dwt_detail_level_{level}_energy'] = np.sum(np.square(detail_coeffs
166 )) if len(detail_coeffs) > 0 else 0.0
167     except:
168         for i in range(1, max_level + 1): feats[f'dwt_detail_level_{i}_energy'] = 0.0
169     return feats
170
171 def extract_poincare_map_features(rr):
172     feats = {k: 0.0 for k in ['sd1', 'sd2', 'sd1_sd2_ratio', 'poincare_area', '
173     poincare_asymmetry_index']}
174     if len(rr) < 3: return feats
175     diff_rr = np.diff(rr)
176     if len(diff_rr) >= 1: feats['sd1'] = np.sqrt(0.5 * np.var(diff_rr, ddof=1) if len(
177     diff_rr) > 1 else 0.0)
178     if len(rr) >= 2:
179         sd2_term = 2 * (np.var(rr, ddof=1) if len(rr) > 1 else 0.0) - 0.5 * (np.var(
180     diff_rr, ddof=1) if len(diff_rr) > 1 else 0.0)
181         feats['sd2'] = np.sqrt(sd2_term) if sd2_term >= 0 else 0.0
182         if feats['sd2'] > 1e-10: feats['sd1_sd2_ratio'] = feats['sd1'] / feats['sd2']
183         feats['poincare_area'] = np.pi * feats['sd1'] * feats['sd2']
184
185     above = diff_rr[diff_rr > 0]; below = diff_rr[diff_rr < 0]
186     mean_above = np.mean(above) if len(above) > 0 else 0.0
187     mean_below = np.mean(np.abs(below)) if len(below) > 0 else 0.0
188     if mean_above > 1e-10: feats['poincare_asymmetry_index'] = mean_below / mean_above
189     elif mean_below > 1e-10: feats['poincare_asymmetry_index'] = 1000.0
190     return feats
191
192 def permutation_entropy(x, order=3, delay=1):
193     n = len(x)
194     if n < (order - 1) * delay + order: return 0.0
195     permutations = {}
196     if delay == 1: segments = np.lib.stride_tricks.as_strided(x, shape=(n - order + 1,
197     order), strides=(x.itemsize, x.itemsize))
198     else: segments = np.array([x[i:i + order * delay:delay] for i in range(n - (order -
199     1) * delay)])
200     if len(segments) == 0: return 0.0
201     for s in segments:
202         rank = tuple(np.argsort(np.argsort(s)))

```

```

197     permutations[rank] = permutations.get(rank, 0) + 1
198     total = sum(permutations.values())
199     if total == 0: return 0.0
200     probs = np.array(list(permutations.values())) / total
201     pe = -np.sum(probs * np.log2(probs + 1e-10))
202     max_pe = np.log2(float(math.factorial(order)))
203     return pe / max_pe if max_pe > 0 else 0.0
204
205 def multiscale_permutation_entropy(x, max_scale=10, order=3):
206     mpe = []
207     for scale in range(1, max_scale + 1):
208         cg = _coarse_grain(x, scale)
209         mpe.append(permutation_entropy(cg, order=order) if len(cg) >= order else 0.0)
210     return np.array(mpe)
211
212 def central_tendency_measure(rr, threshold_ms=5.0):
213     if len(rr) < 2: return 0.0
214     rr_n, rr_n_plus_1 = rr[:-1], rr[1:]
215     mean_rr = np.mean(rr)
216     points_in_center = np.sum((np.abs(rr_n - mean_rr) <= threshold_ms) & (np.abs(
217         rr_n_plus_1 - mean_rr) <= threshold_ms))
218     return points_in_center / len(rr_n) if len(rr_n) > 0 else 0.0
219
220 def extract_features(rr):
221     f = {}; dt = 0.00297
222     f['mean_rr'] = np.mean(rr); f['std_rr'] = np.std(rr)
223     f['cv'] = np.std(rr) / (np.mean(rr) + 1e-10) if np.mean(rr) != 0 else 0.0
224     f['gini'] = gini(rr)
225     f['pip'] = np.sum(np.diff(np.sign(np.diff(rr))) != 0) / len(rr) if len(rr) > 1
226     else 0.0
227
228     f.update(extract_poincare_map_features(rr))
229
230     f['sampen'] = fast_sampen(rr)
231     f['dfa'] = fast_dfa(rr)
232
233     if len(rr) > 2:
234         acc = np.gradient(np.gradient(rr))
235         gamma = np.clip(acc / (np.abs(rr) + 1e-10), -1e3, 1e3).reshape(-1, 1)
236         t = (np.arange(len(rr)) * dt).reshape(-1, 1); freqs = np.fft.rfftfreq(len(rr),
237             d=dt)
238         if freqs.size > 0 and t.size == len(rr):
239             phase = np.exp(-1j * (2 * np.pi * freqs[None, :] * t + gamma * t**2))
240             spec = np.abs(np.sum(rr[:, None] * phase, axis=0))
241             f['accelerated_spectral_entropy'] = scipy.stats.entropy(spec + 1e-10) if
242             np.sum(spec) > 0 else 0.0
243         else: f['accelerated_spectral_entropy'] = 0.0

```

```

240     else: f['accelerated_spectral_entropy'] = 0.0
241
242     f['hjorth_mobility'] = hjorth_mobility(rr)
243     f['hjorth_complexity'] = hjorth_complexity(rr)
244     f['higuchi_fd'] = higuchi_fd(rr)
245
246     tkeo = teager_kaiser_energy(rr)
247     if len(tkeo) > 1:
248         f['tkeo_max'] = np.max(tkeo); f['tkeo_sum_abs'] = np.sum(np.abs(tkeo)); f['
tkeo_std'] = np.std(tkeo)
249         peaks, _ = scipy.signal.find_peaks(tkeo, distance=5, prominence=max(0.05 * np.
max(tkeo), 0.01))
250         f['tkeo_num_peaks'] = len(peaks)
251         if len(peaks) > 0:
252             f['tkeo_peak_amplitude_mean'] = np.mean(tkeo[peaks])
253             total_tkeo_abs = np.sum(np.abs(tkeo))
254             f['tkeo_peak_energy_ratio'] = np.sum(np.abs(tkeo[peaks])) / total_tkeo_abs
255             if total_tkeo_abs > 1e-10 else 0.0
256             else: f['tkeo_peak_amplitude_mean'] = 0.0; f['tkeo_peak_energy_ratio'] = 0.0
257         else:
258             f['tkeo_max'] = 0.0; f['tkeo_sum_abs'] = 0.0; f['tkeo_std'] = 0.0; f['
tkeo_num_peaks'] = 0
259             f['tkeo_peak_amplitude_mean'] = 0.0; f['tkeo_peak_energy_ratio'] = 0.0
260
261     mse = multiscale_sampen(rr, max_scale=10)
262     valid_mse = mse[mse > 0]
263     f['mse_sampen_mean'] = np.mean(valid_mse) if len(valid_mse) > 0 else 0.0
264
265     if _PYWT_AVAILABLE:
266         f.update(extract_wpd_features(rr))
267         f.update(extract_dwt_features(rr))
268     else:
269         for i in range(16): f[f'wpd_band_{i}_energy'] = 0.0; f[f'wpd_band_{i}_entropy'
] = 0.0
270         for i in range(1, 5): f[f'dwt_detail_level_{i}_energy'] = 0.0
271
272     f['permutation_entropy'] = permutation_entropy(rr)
273     f['central_tendency_measure'] = central_tendency_measure(rr)
274
275     mpe = multiscale_permutation_entropy(rr, max_scale=10)
276     valid_mpe = mpe[mpe > 0]
277     f['mpe_mean'] = np.mean(valid_mpe) if len(valid_mpe) > 0 else 0.0
278     for i, val in enumerate(mpe): f[f'mpe_scale_{i+1}'] = val
279
280     return f
281
282 print(">>> [1/4] Extrairando vetor de características clássicas...")

```

```

282 df_f = pd.DataFrame([extract_features(rr) for rr in tqdm(X_raw)])
283 X_total, f_names = df_f.values, df_f.columns
284
285 # Tratamento de infs e NaNs residuais gerados pela matematica pesada
286 X_total = np.nan_to_num(X_total, nan=0.0, posinf=0.0, neginf=0.0)
287
288 # =====
289 # 3. TREINAMENTO DO MODELO CLASSICO (100 RODADAS)
290 # =====
291 print(">>> [2/4] Treinando Voting Classifier (100 Runs)...")
292 history_acc, cm_sum,imps = [], np.zeros((2, 2)), []
293 best_acc, best_seed, best_cm, best_report = 0, -1, None, None
294
295 start_time = time.time()
296
297 for seed in tqdm(range(100), desc="Classic ML Runs"):
298     X_train, X_test, y_train, y_test = train_test_split(X_total, y_total, test_size
299     =0.3, random_state=seed, stratify=y_total)
300
301     # 0 StandardScaler e ajustado apenas nos dados de treino para evitar vazamento de
302     dados
303     sc = StandardScaler().fit(X_train)
304     X_tr, X_te = sc.transform(X_train), sc.transform(X_test)
305
306     X_res, y_res = BorderlineSMOTE(random_state=42).fit_resample(X_tr, y_train)
307
308     clf = VotingClassifier(estimators=[
309         ('et', ExtraTreesClassifier(n_estimators=2000, max_depth=15, min_samples_split
310         =2, min_samples_leaf=1, random_state=seed, class_weight='balanced', n_jobs=-1)),
311         ('xgb', XGBClassifier(n_estimators=2000, learning_rate=0.01, max_depth=7,
312         subsample=0.7, colsample_bytree=0.7, gamma=0.4, min_child_weight=1, reg_lambda=5.0,
313         scale_pos_weight=1.0, verbosity=0, random_state=seed, n_jobs=-1))
314     ], voting='soft', n_jobs=1).fit(X_res, y_res)
315
316     y_pred = clf.predict(X_te)
317     acc = accuracy_score(y_test, y_pred)
318     history_acc.append(acc)
319     cm_sum += confusion_matrix(y_test, y_pred)
320
321     if hasattr(clf.named_estimators_['et'], 'feature_importances_'):
322         imps.append(clf.named_estimators_['et'].feature_importances_)
323
324     if acc > best_acc:
325         best_acc = acc; best_seed = seed
326         best_cm = confusion_matrix(y_test, y_pred)
327         best_report = classification_report(y_test, y_pred, target_names=['Normotenso',
328         'Hipertensao em intervencao farmacologica'])

```

```

323
324 tempo_total_min = (time.time() - start_time) / 60
325
326 # =====
327 # 4. RELATORIO E VISUALIZACAO
328 # =====
329 print("\n" + "="*60)
330 print(" RELATORIO DO BENCHMARK CLASSICO ")
331 print("="*60)
332 print(f"MEDIA GLOBAL (100 runs): {np.mean(history_acc)*100:.2f}%")
333 print(f"DESVIO PADRAO: {np.std(history_acc)*100:.2f}%")
334 mean_cm = cm_sum / 100
335 print(f"RECALL MEDIO NORMOTENSO: {(mean_cm[0,0]/(mean_cm[0,0]+mean_cm[0,1])*100):.2f}%")
336 print(f"RECALL MEDIO HIPERTENSA EM INTERVENCAO FARMACOLOGICA: {(mean_cm[1,1]/(mean_cm
    [1,0]+mean_cm[1,1])*100):.2f}%")
337 print("-" * 60)
338 print(f"MELHOR ACURACIA OBSERVADA: {best_acc*100:.2f}% (Seed {best_seed})")
339 print(f"TEMPO TOTAL DE TREINAMENTO: {tempo_total_min:.2f} minutos")
340 print("\n>>> Classification Report do Pico:")
341 print(best_report)
342 print("="*60)
343
344 print("\n>>> [4/4] Gerando Painel Visual...")
345 fig, ax = plt.subplots(1, 3, figsize=(20, 5))
346
347 # 1. Matriz Media
348 sns.heatmap(mean_cm, annot=True, fmt='.1f', cmap='Oranges', ax=ax[0], xticklabels=['
    Normo', 'Hiper farm.'], yticklabels=['Normo', 'Hiper farm.'])
349 ax[0].set_title(f'Matriz classica media (100 Runs)\nAcuracia: {np.mean(history_acc)
    *100:.2f}%')
350 ax[0].set_xlabel('Predicao'); ax[0].set_ylabel('Real')
351
352 # 2. Matriz do Pico
353 sns.heatmap(best_cm, annot=True, fmt='d', cmap='Greens', ax=ax[1], xticklabels=['Normo'
    , 'Hiper farm.'], yticklabels=['Normo', 'Hiper farm.'])
354 ax[1].set_title(f'Matriz classica do melhor caso (Seed {best_seed})\nAcuracia: {
    best_acc*100:.2f}%')
355 ax[1].set_xlabel('Predicao'); ax[1].set_ylabel('Real')
356
357 # 3. Top Features (ExtraTrees)
358 meanimps = np.mean(imps, axis=0)
359 feat_imp = pd.Series(meanimps, index=f_names).sort_values(ascending=True).tail(10)
360 feat_imp.plot(kind='barh', color='teal', ax=ax[2])
361 ax[2].set_title('Top 10 caracteristicas mais importantes (ExtraTrees)')
362 ax[2].set_xlabel('Importancia media')
363

```

```
364 plt.tight_layout()  
365 plt.show()
```

Listing B.1: Código-fonte do benchmark clássico final.

Apêndice C

Código-fonte do modelo quântico híbrido final

Este apêndice apresenta o código-fonte utilizado para executar o modelo quântico híbrido final descrito na metodologia. A implementação contempla o carregamento da base de dados, a extração das quatro características utilizadas pelo modelo quântico, o tratamento por *PowerTransformer*, o balanceamento por *BorderlineSMOTE*, a duplicação do vetor de entrada por redundância lateral inspirada em *Data Re-uploading*, o mapeamento angular no intervalo de 0 a 2 radianos e a avaliação do QSVC com ZZFeatureMap em 8 *qubits*. Os arquivos .py correspondentes acompanham a versão digital deste trabalho para facilitar reprodução e execução fora do PDF.

```
1 # =====
2 # MODELO QUANTICO HIBRIDO FINAL - QSVC COM 8 QUBITS E REDUNDANCIA LATERAL
3 # =====
4 !pip install --quiet qiskit qiskit-aer qiskit-machine-learning qiskit-algorithms
   pylatexenc imbalanced-learn seaborn
5
6 import os, urllib.request, warnings, time, math
7 import numpy as np
8 import pandas as pd
9 import scipy.stats
10 import matplotlib.pyplot as plt
11 from tqdm.notebook import tqdm
12
13 from sklearn.model_selection import train_test_split
14 from sklearn.preprocessing import MinMaxScaler, PowerTransformer
15 from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
16 from imblearn.over_sampling import BorderlineSMOTE
17
18 from qiskit.circuit.library import ZZFeatureMap
19 from qiskit.primitives import StatevectorSampler
20 from qiskit_algorithms.state_fidelities import ComputeUncompute
```

```

21 from qiskit_machine_learning.kernels import FidelityQuantumKernel
22 from qiskit_machine_learning.algorithms import QSVC
23
24 warnings.filterwarnings('ignore')
25
26 # 1. CARREGAMENTO DOS DADOS
27 url = "https://github.com/rafaelgov95/data_set-h-n_v1/raw/refs/heads/main/
      resultados_v3.xlsx"
28 file_path = "resultados.xlsx"
29 if not os.path.exists(file_path): urllib.request.urlretrieve(url, file_path)
30
31 res = pd.read_excel(file_path, header=None).values
32 RR0, RR1 = res[1:67, 1:257].astype(np.float32), res[67:200, 1:257].astype(np.float32)
33 X_raw = np.concatenate([RR0, RR1], axis=0)
34 y_total = np.concatenate([np.zeros(len(RR0)), np.ones(len(RR1))], axis=0)
35
36 # 2. EXTRACAO DAS QUATRO CARACTERISTICAS PRINCIPAIS
37 def fast_sampen(x, m=2, r=0.2):
38     std_x = np.std(x); r *= std_x; N = len(x)
39     if std_x == 0: return 0.0
40     def _phi(m_len):
41         if N < m_len + 1: return 0.0
42         shape = (N - m_len + 1, m_len)
43         X = np.lib.stride_tricks.as_strided(x, shape=shape, strides=(x.itemsize, x.
44         itemsize))
45         dist = np.max(np.abs(X[:, None, :] - X[None, :, :]), axis=2)
46         np.fill_diagonal(dist, np.inf)
47         return np.sum(dist <= r) / (shape[0] * (shape[0] - 1))
48     p1, p2 = _phi(m), _phi(m+1)
49     return -np.log(p2 / p1) if (p1 > 0 and p2 > 0) else 0.0
50
51 def permutation_entropy(x, order=3):
52     n = len(x)
53     if n < order: return 0.0
54     permutations = {}
55     segments = np.lib.stride_tricks.as_strided(x, shape=(n - order + 1, order),
56     strides=(x.itemsize, x.itemsize))
57     for segment in segments:
58         rank = tuple(np.argsort(np.argsort(segment)))
59         permutations[rank] = permutations.get(rank, 0) + 1
60     total = sum(permutations.values())
61     probs = np.array(list(permutations.values())) / total
62     return -np.sum(probs * np.log2(probs + 1e-10)) / np.log2(float(math.factorial(
63     order)))
64
65 def extract_top4(rr):
66     f = {}; dt = 0.00297

```

```

64 f['sampen'] = fast_sampen(rr)
65 f['pe'] = permutation_entropy(rr)
66 tkeo = rr[1:-1]**2 - rr[:-2] * rr[2:]
67 f['tkeo_sum'] = np.sum(np.abs(tkeo)) if len(tkeo) > 0 else 0.0
68 acc = np.gradient(np.gradient(rr))
69 gamma = np.clip(acc / (np.abs(rr) + 1e-10), -1e3, 1e3).reshape(-1, 1)
70 t = (np.arange(len(rr)) * dt).reshape(-1, 1)
71 freqs = np.fft.rfftfreq(len(rr), d=dt)
72 if freqs.size > 0 and t.size == len(rr):
73     phase = np.exp(-1j * (2 * np.pi * freqs[None, :] * t + gamma * t**2))
74     spec = np.abs(np.sum(rr[:, None] * phase, axis=0))
75     f['accelerated_spectral_entropy'] = scipy.stats.entropy(spec + 1e-10) if np.
sum(spec) > 0 else 0.0
76 else: f['accelerated_spectral_entropy'] = 0.0
77 return f
78
79 print(">>> [1/4] Extrairdo as quatro caracteristicas principais...")
80 X_total = pd.DataFrame([extract_top4(rr) for rr in tqdm(X_raw)]).values
81
82 # 3. CONFIGURACAO DO KERNEL QUANTICO (8 QUBITS)
83 # 0 espaco de representacao e ampliado por emaranhamento circular em 8 qubits
84 feature_map = ZZFeatureMap(feature_dimension=8, reps=1, entanglement='circular')
85 sampler = StatevectorSampler()
86 fidelity = ComputeUncompute(sampler=sampler)
87 q_kernel = FidelityQuantumKernel(fidelity=fidelity, feature_map=feature_map)
88
89 # 4. TREINAMENTO E AVALIACAO (10 RODADAS)
90 print(">>> [2/4] Executando 10 rodadas do modelo quantico hibrido final...")
91 history_acc, cm_sum = [], np.zeros((2, 2))
92 best_acc, best_seed, best_cm, best_report = 0, -1, None, None
93 TOTAL_RUNS = 10
94
95 start_time_total = time.time()
96
97 for seed in tqdm(range(TOTAL_RUNS), desc="QSVM V14"):
98     X_train, X_test, y_train, y_test = train_test_split(X_total, y_total, test_size
=0.3, random_state=seed, stratify=y_total)
99
100     # Padronizacao e SMOTE
101     pt = PowerTransformer().fit(X_train)
102     X_tr_norm, X_te_norm = pt.transform(X_train), pt.transform(X_test)
103     X_res, y_res = BorderlineSMOTE(random_state=42).fit_resample(X_tr_norm, y_train)
104
105     # Redundancia lateral inspirada em Data Re-uploading: duplicacao das 4
caracteristicas para formar 8 entradas
106     X_res_8q = np.hstack([X_res, X_res])
107     X_te_8q = np.hstack([X_te_norm, X_te_norm])

```

```

108
109 # Mapeamento angular no intervalo de 0 a 2 radianos
110 mm_base = MinMaxScaler(feature_range=(0, 2)).fit(X_res_8q)
111 X_train_q = mm_base.transform(X_res_8q)
112 X_test_q = mm_base.transform(X_te_8q)
113
114 # Pesos manuais para compensar a classe normotensa e margem C=2.0
115 pesos = {0: 1.3, 1: 1.0}
116 qsvc = Q SVC(quantum_kernel=q_kernel, C=2.0, class_weight=pesos)
117
118 qsvc.fit(X_train_q, y_res)
119 y_pred_q = qsvc.predict(X_test_q)
120
121 acc = accuracy_score(y_test, y_pred_q)
122 history_acc.append(acc)
123 cm_sum += confusion_matrix(y_test, y_pred_q)
124
125 if acc > best_acc:
126     best_acc = acc
127     best_seed = seed
128     best_cm = confusion_matrix(y_test, y_pred_q)
129     best_report = classification_report(y_test, y_pred_q, target_names=['
Normotenso', 'Hipertensao em intervencao farmacologica'])
130
131 # 5. RELATORIO TEXTUAL
132 mean_cm = cm_sum / TOTAL_RUNS
133 tempo_total_min = (time.time() - start_time_total) / 60
134
135 print(f"\n{'='*65}")
136 print(f" RELATORIO DO MODELO QUANTICO HIBRIDO FINAL: Q SVC (8 QUBITS) ")
137 print(f"{'='*65}")
138 print(f"-> MEDIA GLOBAL (10 runs): {np.mean(history_acc)*100:.2f}%")
139 print(f"-> DESVIO PADRAO: {np.std(history_acc)*100:.2f}%")
140 print(f"-> MELHOR ACURACIA OBSERVADA: {best_acc*100:.2f}% (Seed {best_seed})")
141 print(f"-> TEMPO TOTAL: {tempo_total_min:.2f} min")
142 print("-" * 65)
143 print(">>> MATRIZ DE CONFUSAO MEDIA (10 RUNS):")
144 print(f"          | Predito Normotenso | Predito Hipertensao em intervencao
farmacologica |")
145 print(f" Real Normotenso | {mean_cm[0,0]:18.2f} | {mean_cm[0,1]:18.2f} |")
146 print(f" Real Hipertensao em intervencao farmacologica | {mean_cm[1,0]:18.2f} | {
mean_cm[1,1]:18.2f} |")
147 print("-" * 65)
148 print(f">>> MATRIZ DE CONFUSAO DO MELHOR CASO (SEED {best_seed}):")
149 print(f"          | Predito Normotenso | Predito Hipertensao em intervencao
farmacologica |")
150 print(f" Real Normotenso | {best_cm[0,0]:18d} | {best_cm[0,1]:18d} |")

```

```
151 print(f" Real Hipertensao em intervencao farmacologica | {best_cm[1,0]:18d} | {best_cm  
    [1,1]:18d} |")  
152 print("-" * 65)  
153 print(">>> CLASSIFICATION REPORT DO MELHOR CASO:")  
154 print(best_report)  
155 print("="*65)
```

Listing C.1: Código-fonte do modelo quântico híbrido final.

Apêndice D

Código-fonte da validação cega por amostras retidas

Este apêndice apresenta uma implementação utilizada como auditoria complementar durante o desenvolvimento experimental. O procedimento remove aleatoriamente uma amostra normotensa e uma amostra da classe hipertensa em intervenção farmacológica antes dos ajustes aprendidos de pré-processamento, do balanceamento sintético e do treinamento. Em seguida, o modelo é treinado sobre o conjunto remanescente em múltiplas partições internas, e as duas amostras previamente retidas são avaliadas apenas ao final como observações inéditas. A extração de características é determinística e aplicada individualmente a cada série RR, sem uso de estatísticas globais das classes.

Essa validação não substitui as métricas finais reportadas nos Capítulos 4 e 5, pois sua escala é reduzida e seu objetivo é metodológico. Seu papel foi verificar se configurações promissoras em partições convencionais mantinham comportamento coerente diante de amostras completamente reservadas. Configurações que não se sustentavam nesse procedimento foram descartadas antes de integrarem os marcos consolidados discutidos no texto.

```
1 # =====
2 # VALIDACAO CEGA POR AMOSTRAS RETIDAS - AUDITORIA COMPLEMENTAR
3 # =====
4 !pip install --quiet openpyxl imbalanced-learn tqdm xgboost seaborn
5
6 import os
7 import urllib.request
8 import warnings
9 import random
10 import numpy as np
11 import pandas as pd
12 import scipy.stats
13 import matplotlib.pyplot as plt
14 import seaborn as sns
```

```

15 from tqdm.notebook import tqdm
16
17 from sklearn.model_selection import train_test_split
18 from sklearn.ensemble import ExtraTreesClassifier, VotingClassifier
19 from xgboost import XGBClassifier
20 from sklearn.preprocessing import PowerTransformer
21 from sklearn.metrics import accuracy_score, confusion_matrix
22 from imblearn.over_sampling import BorderlineSMOTE
23
24 warnings.filterwarnings("ignore")
25
26 TESTES_CEGOS = 50
27 ITERACOES_POR_TREINO = 100
28 RANDOM_STATE = 42
29
30 random.seed(RANDOM_STATE)
31 np.random.seed(RANDOM_STATE)
32
33 # =====
34 # 1. CARREGAMENTO DOS DADOS
35 # =====
36 url = "https://github.com/rafaelgov95/data_set-h-n_v1/raw/refs/heads/main/
37     resultados_v3.xlsx"
38 file_path = "resultados.xlsx"
39
40 if not os.path.exists(file_path):
41     urllib.request.urlretrieve(url, file_path)
42
43 res = pd.read_excel(file_path, header=None).values
44 RR0 = res[1:67, 1:257].astype(np.float32)
45 RR1 = res[67:200, 1:257].astype(np.float32)
46
47 X_raw = np.concatenate([RR0, RR1], axis=0)
48 y_total = np.concatenate([np.zeros(len(RR0)), np.ones(len(RR1))], axis=0)
49
50 # =====
51 # 2. ENGENHARIA DE CARACTERISTICAS
52 # =====
53 def gini(array):
54     array = np.sort(np.array(array, dtype=np.float64))
55     n = array.shape[0]
56     index = np.arange(1, n + 1)
57     total = np.sum(array)
58     if n == 0 or total == 0:
59         return 0.0
60     return np.sum((2 * index - n - 1) * array) / (n * total)

```

```

61
62 def fast_sampen(x, m=2, r=0.2):
63     std_x = np.std(x)
64     if std_x == 0 or len(x) < m + 2:
65         return 0.0
66
67     r_tol = r * std_x
68     n = len(x)
69
70     def _phi(m_len):
71         if n < m_len + 1:
72             return 0.0
73         shape = (n - m_len + 1, m_len)
74         embedded = np.lib.stride_tricks.as_strided(
75             x,
76             shape=shape,
77             strides=(x.itemsize, x.itemsize),
78         )
79         dist = np.max(np.abs(embedded[:, None, :] - embedded[None, :, :]), axis=2)
80         np.fill_diagonal(dist, np.inf)
81         denominator = shape[0] * (shape[0] - 1)
82         return np.sum(dist <= r_tol) / denominator if denominator > 0 else 0.0
83
84     p1 = _phi(m)
85     p2 = _phi(m + 1)
86     return -np.log(p2 / p1) if p1 > 0 and p2 > 0 else 0.0
87
88
89 def higuchi_fd(x, k_max=8):
90     n_times = x.size
91     if n_times < 3:
92         return 0.0
93
94     lengths = np.zeros(k_max)
95     for k in range(1, k_max + 1):
96         length_k = 0.0
97         for m_idx in range(k):
98             indices = np.arange(m_idx, n_times, k)
99             if len(indices) < 2:
100                 continue
101             sub = x[indices]
102             norm = (n_times - 1) / (k * (len(sub) - 1))
103             length_k += np.sum(np.abs(np.diff(sub))) * norm
104         lengths[k - 1] = length_k / k if k > 0 else 0.0
105
106     k_values = np.arange(1, k_max + 1)
107     valid = lengths > 0

```

```

108     if np.sum(valid) < 2:
109         return 0.0
110
111     coeffs = np.polyfit(np.log(k_values[valid]), np.log(lengths[valid]), 1)
112     return max(0.0, -coeffs[0])
113
114
115 def extract_features(rr):
116     features = {}
117     dt = 0.00297
118
119     features["cv"] = np.std(rr) / (np.mean(rr) + 1e-10)
120     features["gini"] = gini(rr)
121     features["sampen"] = fast_sampen(rr)
122     features["higuchi_fd"] = higuchi_fd(rr)
123
124     d1 = np.diff(rr)
125     d2 = np.diff(d1)
126     var_rr = np.var(rr)
127     var_d1 = np.var(d1)
128
129     features["hjorth_mobility"] = np.sqrt(var_d1 / var_rr) if var_rr > 0 else 0.0
130     if var_d1 > 0 and features["hjorth_mobility"] > 0:
131         features["hjorth_complexity"] = np.sqrt(np.var(d2) / var_d1) / features["
132         hjorth_mobility"]
133     else:
134         features["hjorth_complexity"] = 0.0
135
136     tkeo = rr[1:-1] ** 2 - rr[:-2] * rr[2:]
137     features["tkeo_sum"] = np.sum(np.abs(tkeo)) if len(tkeo) > 0 else 0.0
138
139     acceleration = np.gradient(np.gradient(rr))
140     gamma = np.clip(acceleration / (np.abs(rr) + 1e-2), -50, 50).reshape(-1, 1)
141     t = (np.arange(len(rr)) * dt).reshape(-1, 1)
142     freqs = np.fft.rfftfreq(len(rr), d=dt)
143
144     if freqs.size > 0:
145         phase = np.exp(-1j * (2 * np.pi * freqs[None, :] * t + gamma * t**2))
146         spectrum = np.abs(np.sum(rr[:, None] * phase, axis=0))
147         features["accelerated_spectral_entropy"] = scipy.stats.entropy(spectrum + 1e
148         -10)
149     else:
150         features["accelerated_spectral_entropy"] = 0.0
151
152     return features

```

```

153 print(">>> Preparando dados e extraindo características...")
154 df_features = pd.DataFrame([extract_features(rr) for rr in X_raw])
155 X_all = np.nan_to_num(df_features.values, nan=0.0, posinf=0.0, neginf=0.0)
156 feature_names = df_features.columns
157
158 idx_normotenso = np.where(y_total == 0)[0]
159 idx_hipertenso = np.where(y_total == 1)[0]
160
161 # =====
162 # 3. VALIDACAO CEGA POR AMOSTRAS RETIDAS
163 # =====
164 blind_results = []
165
166 for teste in range(1, TESTES_CEGOS + 1):
167     print("\n" + "=" * 80)
168     print(f"Iniciando validacao cega {teste}/{TESTES_CEGOS}")
169     print("=" * 80)
170
171     amostra_normotensa = random.choice(idx_normotenso)
172     amostra_hipertensa = random.choice(idx_hipertenso)
173
174     X_blind = X_all[[amostra_normotensa, amostra_hipertensa]]
175     y_blind = y_total[[amostra_normotensa, amostra_hipertensa]]
176
177     train_mask = np.ones(len(y_total), dtype=bool)
178     train_mask[[amostra_normotensa, amostra_hipertensa]] = False
179
180     X_pool = X_all[train_mask]
181     y_pool = y_total[train_mask]
182
183     history_acc = []
184     cm_sum = np.zeros((2, 2))
185     importances = []
186     best_acc = 0.0
187     best_seed = -1
188     best_cm = None
189     best_model = None
190     best_transformer = None
191
192     for seed in tqdm(range(ITERACOES_POR_TREINO), desc=f"Treinamento interno {teste}",
193                     leave=False):
194         X_train, X_test, y_train, y_test = train_test_split(
195             X_pool,
196             y_pool,
197             test_size=0.3,
198             random_state=seed,
199             stratify=y_pool,

```

```

199     )
200
201     transformer = PowerTransformer().fit(X_train)
202     X_train_norm = transformer.transform(X_train)
203     X_test_norm = transformer.transform(X_test)
204
205     X_resampled, y_resampled = BorderlineSMOTE(random_state=42).fit_resample(
206         X_train_norm,
207         y_train,
208     )
209
210     classifier = VotingClassifier(
211         estimators=[
212             ("et", ExtraTreesClassifier(n_estimators=500, max_depth=15,
213 random_state=42, n_jobs=-1)),
214             ("xgb", XGBClassifier(n_estimators=500, learning_rate=0.03, max_depth
215 =6, random_state=42, n_jobs=-1)),
216         ],
217         voting="soft",
218     )
219
220     classifier.fit(X_resampled, y_resampled)
221     y_pred = classifier.predict(X_test_norm)
222     acc = accuracy_score(y_test, y_pred)
223
224     history_acc.append(acc)
225     cm_sum += confusion_matrix(y_test, y_pred)
226     importances.append(classifier.named_estimators_["et"].feature_importances_)
227
228     if acc > best_acc:
229         best_acc = acc
230         best_seed = seed
231         best_cm = confusion_matrix(y_test, y_pred)
232         best_model = classifier
233         best_transformer = transformer
234
235     X_blind_norm = best_transformer.transform(X_blind)
236     blind_prediction = best_model.predict(X_blind_norm)
237
238     blind_correct_normotenso = int(blind_prediction[0] == y_blind[0])
239     blind_correct_hipertenso = int(blind_prediction[1] == y_blind[1])
240     blind_results.append([blind_correct_normotenso, blind_correct_hipertenso])
241
242     mean_cm = cm_sum / ITERACOES_POR_TREINO
243
244     print("\nRelatorio da rodada")
245     print(f"Media global do pool: {np.mean(history_acc) * 100:.2f}%")

```

```

244 print(f"Desvio padrao do pool: {np.std(history_acc) * 100:.2f}%")
245 print(f"Recall medio normotenso: {(mean_cm[0, 0] / (mean_cm[0, 0] + mean_cm[0, 1]) * 100):.2f}%")
246 print(f"Recall medio hipertensao em intervencao farmacologica: {(mean_cm[1, 1] / (mean_cm[1, 0] + mean_cm[1, 1]) * 100):.2f}%")
247 print(f"Melhor acuracia interna: {best_acc * 100:.2f}% (seed {best_seed})")
248 print("Resultado das amostras retidas:")
249 print(f"Amostra normotensa retida: {'correta' if blind_correct_normotenso else 'incorreta'}")
250 print(f"Amostra classe hipertensa em intervencao farmacologica retida: {'correta' if blind_correct_hipertenso else 'incorreta'}")
251
252 fig, ax = plt.subplots(1, 3, figsize=(18, 5))
253
254 sns.heatmap(mean_cm, annot=True, fmt=".1f", cmap="Oranges", ax=ax[0], xticklabels=["Normo", "Hiper farm."], yticklabels=["Normo", "Hiper farm."])
255 ax[0].set_title(f"Matriz media do pool - rodada {teste}")
256
257 sns.heatmap(best_cm, annot=True, fmt="d", cmap="Greens", ax=ax[1], xticklabels=["Normo", "Hiper farm."], yticklabels=["Normo", "Hiper farm."])
258 ax[1].set_title(f"Matriz do melhor caso interno - seed {best_seed}")
259
260 mean_importances = np.mean(importances, axis=0)
261 feature_importance = pd.Series(mean_importances, index=feature_names).sort_values(ascending=True)
262 feature_importance.plot(kind="barh", color="teal", ax=ax[2])
263 ax[2].set_title("Importancia media das caracteristicas")
264
265 plt.tight_layout()
266 plt.show()
267 plt.close(fig)
268
269 blind_results = np.array(blind_results)
270 print("\n" + "=" * 80)
271 print("Resumo das validacoes cegas")
272 print(f"Acuracia cega normotensos: {np.mean(blind_results[:, 0]) * 100:.2f}%")
273 print(f"Acuracia cega da classe hipertensa em intervencao farmacologica: {np.mean(blind_results[:, 1]) * 100:.2f}%")
274 print(f"Acuracia cega global: {np.mean(blind_results) * 100:.2f}%")
275 print("=" * 80)

```

Listing D.1: Código-fonte da validação cega por amostras retidas.