



UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM  
INSTITUTO DE COMPUTAÇÃO- ICOMP  
PROGRAMA PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI

# Estratégias de aprendizado para detecção de falhas aprimoradas por kernels quânticos

Mauro Queiroz Nooblath Neto

Manaus - AM  
Agosto de 2025

Mauro Queiroz Nooblath Neto

# Estratégias de aprendizado para detecção de falhas aprimoradas por kernels quânticos

Dissertação de mestrado submetida à avaliação, como requisito geral, para a obtenção do título de Mestre em Informática no Programa de Pós-Graduação em Informática, Instituto de Computação.

Orientadora

Rosiane de Freitas-Rodrigues, Profa. Ph.D.

Universidade Federal do Amazonas - UFAM

Instituto de Computação- IComp

Manaus - AM

Agosto de 2025

Ficha Catalográfica

Elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

---

N816e      Nooblath Neto, Mauro Queiroz  
Estratégias de aprendizado para detecção de falhas aprimoradas por  
kernels quânticos / Mauro Queiroz Nooblath Neto. - 2025.  
97 f. : il., color. ; 31 cm.

Orientador(a): Rosiane de Freitas Rodrigues.  
Dissertação (mestrado) - Universidade Federal do Amazonas, Programa  
de Pós-Graduação em Informática, Manaus, 2025.

1. Aprendizado de máquina. 2. Computação quântica. 3. Dataset  
desbalanceado . 4. Função de kernel. 5. Redução de dimensionalidade . I.  
Rodrigues, Rosiane de Freitas. II. Universidade Federal do Amazonas.  
Programa de Pós-Graduação em Informática. III. Título

---



Ministério da Educação  
Universidade Federal do Amazonas  
Coordenação do Programa de Pós-Graduação em Informática

## FOLHA DE APROVAÇÃO

### "ESTRATÉGIA HÍBRIDA DE APRENDIZADO QUÂNTICO PARA DETECÇÃO DE FALHAS EM MANUTENÇÃO PREDITIVA"

**MAURO QUEIROZ NOOBLATH NETO**

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Profa. Dra. Rosiane de Freitas-Rodrigues (Orientadora PPGI), PRESIDENTE  
Instituto de Computação - Universidade Federal do Amazonas (UFAM), Brasil

Documento assinado digitalmente  
**gov.br** EULANDA MIRANDA DOS SANTOS  
Data: 23/08/2025 11:47:34-0300  
Verifique em <https://validar.iti.gov.br>

Profa. Dra. Eulanda Miranda dos Santos (PPGI) - MEMBRO INTERNO  
Instituto de Computação - Universidade Federal do Amazonas (UFAM), Brasil

Documento assinado digitalmente  
**gov.br** HIDEMBERGUE ORDOZGOITH DA FROTA  
Data: 23/08/2025 08:54:29-0300  
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Hidembergue Ordozgoith da Frota - MEMBRO EXTERNO  
Dep. de Física dos Materiais, ICE - Universidade Federal do Amazonas (UFAM), Brasil

Prof. Dr. Askery Alexandre Canabarro - MEMBRO EXTERNO  
Technology Innovation Institute Abu Dhabi, Emirados Árabes Unidos e Universidade Federal de Alagoas  
(UFAL), Brasil

MANAUS, 08 de agosto de 2025.





*Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy.*

Richard Feynman

# Estratégias de aprendizado para detecção de falhas aprimoradas por kernels quânticos

Autor: Mauro Queiroz Nooblath Neto

Orientador: Rosiane de Freitas-Rodrigues, Profa. Ph.D.

## Resumo

Nesta dissertação, é explorado o uso de funções de *kernel* quânticas em modelos híbridos clássico-quânticos voltados à detecção de falhas em turbinas eólicas. Foram utilizados circuitos quânticos parametrizados para mapear dados em espaços de Hilbert de alta dimensionalidade. As arquiteturas dos circuitos foram: *ZZFeatureMap*, *RealAmplitudes* e *EfficientSU2* com quatro estratégias de emaranhamento: *Linear*, *Full*, *Circular* e *Shift-Circular-Alternate*, comparando-se a quatro *kernels* clássicos: *Linear*, *Polynomial*, *Radial Based Function* e *Sigmoid* em um algoritmo de Máquinas de vetores Suporte. O conjunto de dados, com 54 atributos adquiridos por sensores, foi submetido a um algoritmo de Análise de componentes principais para reduzir sua dimensionalidade para 4, 8 e 16 componentes, considerando a variância cumulativa dos dados. O modelo *RealAmplitudes* com emaranhamento *Full* e 16 componentes superou o *kernel Radial Based Function* em métricas padrão de aprendizado de máquina. Análises adicionais com curvas ROC-AUC e matrizes de confusão indicaram ausência de *overfitting*, reforçando o potencial dos *kernels* quânticos em aplicações industriais.

*Palavras-chave:* aprendizado de máquina, computação quântica, dataset desbalanceado, função de *kernel*, manutenção preditiva, redução de dimensionalidade.

# Estratégias de aprendizado para detecção de falhas aprimoradas por kernels quânticos

Autor: Mauro Queiroz Noobla Neto

Orientador: Rosiane de Freitas-Rodrigues, Profa. Ph.D.

## Abstract

This master's thesis investigates the use of quantum kernel functions in hybrid classical-quantum models for fault detection in wind turbines. Parameterized quantum circuits were used to map input data into high-dimensional Hilbert spaces. The quantum circuit architectures analyzed include ZZFeatureMap, RealAmplitudes, and EfficientSU2, each implemented with four entanglement strategies: Linear, Full, Circular, and Shift-Circular-Alternate. These were compared with classical kernels—Linear, Polynomial, Radial Basis Function, and Sigmoid—within a Support Vector Machine framework. The dataset, comprising 54 features collected by turbine sensors, was reduced via Principal Component Analysis to 4, 8, and 16 components based on cumulative variance. The RealAmplitudes circuit with Full entanglement and 16 components outperformed the Radial Based Function kernel in standard machine learning metrics. ROC-AUC curves and confusion matrices showed no overfitting, reinforcing the potential of quantum kernels in industrial fault detection.

*Keywords:* machine learning, quantum computing, imbalanced dataset, kernel methods, predictive maintenance, dimensionality reduction.

---

# SUMÁRIO

Lista de ilustrações . . . . .	9
Lista de Tabelas . . . . .	10
1       INTRODUÇÃO . . . . .	11
1.1 Problemática e questão de pesquisa . . . . .	15
1.2 Objetivos . . . . .	16
1.3 Organização da Dissertação . . . . .	17
2       APRENDIZAGEM QUÂNTICA DE MÁQUINA . . . . .	19
2.1 Computação quântica . . . . .	19
2.1.1 Manipulação de <i>qubits</i> . . . . .	20
2.1.2 Computadores quânticos reais e simulados . . . . .	28
2.2 Aprendizagem de máquina . . . . .	32
2.2.1 Modelos de aprendizagem de máquina convencional . . . . .	33
2.2.2 Modelos de aprendizagem quântica de máquina . . . . .	36
2.2.3 Circuitos quânticos variacionais . . . . .	37
2.2.4 Modelos de QML híbridos . . . . .	39
3       PROBLEMA DE DETECÇÃO DE FALHAS EM MANUTEN- ÇÃO PREDITIVA . . . . .	41

3.1	Abordagens modernas de manutenção de equipamentos industriais	41
3.2	Computação quântica aplicada em estratégias de detecção de falha	43
4	METODOLOGIA . . . . .	45
4.1	Caso de Estudo . . . . .	49
4.2	Procedimentos para execução da estratégia . . . . .	50
4.2.1	Entrada dos Dados . . . . .	51
4.2.2	Tratamento do desbalanceamento do conjunto de dados . . . . .	52
4.2.3	Preparação dos Modelos para compor a estratégia híbrida . . . . .	53
4.2.4	Execução da estratégia híbrida . . . . .	56
4.3	Procedimento de análise . . . . .	56
5	RESULTADOS E DISCUSSÕES . . . . .	58
5.1	Análise da Performance dos Modelos Clássico e Quântico . . . . .	59
5.1.1	Análise dos Modelos de Classificação . . . . .	59
5.2	Resultados da abordagem híbrida clássico-quântica integrada aos regressores . . . . .	86
6	CONSIDERAÇÕES FINAIS . . . . .	88
6.1	Trabalhos Futuros . . . . .	90
	Referências . . . . .	93

---

## LISTA DE ILUSTRAÇÕES

Figura 1 – Esfera de Bloch com um vetor $ \psi\rangle$ apontando para um estado de um <i>qubit</i> . . . . .	23
Figura 2 – Modelo de algoritmo quântico baseado em circuito lógico. . .	25
Figura 3 – Estatística dos estados colapsados após 1024 medidas realizadas no circuito quântico da Figura 2. Cada estado com aproximadamente o mesmo número de ocorrências. . . . .	28
Figura 4 – Esquema de conexão entre os 127 <i>qubits</i> de um dispositivo quântico supercondutor. . . . .	31
Figura 5 – Algoritmo quântico adaptado para as portas nativas do dispositivo quântico. . . . .	31
Figura 6 – Fronteira de decisão resultante do treinamento do classificador SVM, evidenciando os vetores de suporte que definem a margem máxima entre as classes representadas pelas cores azul e verde. . . . .	34
Figura 7 – Conjunto de dados artificialmente construído com classes de difícil separação linear. . . . .	36
Figura 8 – Circuito com Expressibilidade e Capacidade de Emaranhamento.	38

Figura 9 – As etapas de um sistema de Manutenção Preditiva como exemplo para um determinado equipamento da indústria 4.0. . . .	42
Figura 10 – Framework com a estratégia híbrida de detecção de falhas com aprendizado quântico adotada. . . . .	48
Figura 11 – Variância Cumulativa dos dados . . . . .	52
Figura 12 – Modelo Híbrido de Aprendizagem Quântica . . . . .	54
Figura 13 – Arquiteturas de kernels quânticos e emaranhamentos investigadas no trabalho. . . . .	55
Figura 14 – Matrizes de Confusão SVM com 16 Componentes Principais .	63
Figura 15 – Curvas RoC-AuC com 16 Componentes Principais . . . . .	66
Figura 16 – Número de vetores de suporte utilizados para separar cada classe das demais na estratégia <i>One vs. Rest</i> para 16 componentes principais . . . . .	68
Figura 17 – Matrizes de Confusão SVM 8 Componentes Principais . . . .	71
Figura 18 – Curvas RoC-AuC 8 Componentes Principais . . . . .	74
Figura 19 – Quantidade de vetores suporte utilizadas para tentar separar cada classe das outras na estratégia <i>One Vs. Rest</i> para 8 componentes principais. . . . .	76
Figura 20 – Matrizes de Confusão SVM com 4 Componentes Principais . .	78
Figura 21 – Curvas RoC-AuC com 4 Componentes Principais . . . . .	80
Figura 22 – Quantidade de vetores suporte utilizadas para tentar separar cada classe das outras na estratégia <i>One Vs. Rest</i> para 4 componentes principais. . . . .	82



---

## LISTA DE TABELAS

Tabela 1 – Arquiteturas reais de Computação Quântica. . . . .	28
Tabela 2 – Abordagens em QML para Processamento de Dados Clássicos e Quânticos. . . . .	37
Tabela 3 – Distribuição de tipos de falhas do aerogerador e suas causas. .	50
Tabela 4 – Resultados das métricas para diferentes tipos de kernel treina- dos e testados com a base de teste do conjunto A. . . . .	61
Tabela 5 – Métricas de desempenho para diferentes números de <i>features</i> . O MAE é comparado com um limite aceitável. . . . .	83
Tabela 6 – Resultados de desempenho dos diferentes kernels usando o Conjunto A. . . . .	85
Tabela 7 – Resultados de desempenho dos diferentes kernels. . . . .	86

---

## LISTA DE ABREVIATURAS E SIGLAS

**CNN** *Convolution Neural Network*

**GPU** *Graphical Processing Unit*

**NISQ** *Noisy-Intermediate Scale Quantum*

**PCA** *Principal Component Analysis*

**PQC** *Parameterized Quantum Circuit*

**QC** *Quantum Computing*

**QML** *Quantum Machine Learning*

**QSVM** *Quantum Support Vector Machine*

**QuBit** *Quantum Bit*

**SVM** *Support Vector Machine*

**TPU** *Tensor Processing Unit*

# 1

---

## INTRODUÇÃO

A Computação Quântica (QC, do inglês *Quantum Computing*) constitui um campo multidisciplinar que integra os fundamentos da mecânica quântica aos princípios da ciência da computação. Essa área tem se destacado pelo potencial de resolver determinadas classes de problemas com maior eficiência em comparação aos métodos computacionais clássicos, destacando-se em tarefas que exigem representações mais adequadas, bem como maior capacidade de armazenamento e processamento de informações (Nielsen et al., 2010). Esse ganho de eficiência está relacionado aos computadores quânticos utilizarem sistemas que obedecem às propriedades da mecânica quântica como unidades de informação, esses sistemas são conhecidos como *QuBit* (do inglês, *quantum bits*). Portanto, tais sistemas conseguem assumir estados de superposição e emaranhamento permitindo aproveitá-los para desenvolver novas abordagens algorítmicas. Um exemplo notável é o algoritmo quântico desenvolvido por Peter Shor (Shor, 1996) para fatoração de números inteiros e o de Lov. Grover (Grover, 1996) para busca em listas não estruturadas, onde, teoricamente, exemplificam reduções significativas de complexidade computacional

em relação às abordagens convencionais.

Apesar dessas vantagens teoricamente alcançadas, a implementação prática de algoritmos em hardwares quânticos reais ainda enfrenta diversas limitações, sobretudo devido ao fenômeno da decoerência quântica e ao acúmulo de erros durante a execução. Isso ocorre porque os sistemas quânticos são extremamente sensíveis às interações com o ambiente ao qual estão submetidos (Park et al., 2024), exigindo, assim, códigos corretores de erros mais robustos ou novas arquiteturas de chips quânticos.

Nos últimos anos, avanços na engenharia de hardware têm buscado mitigar o acúmulo de erros e decoerência quântica. Por exemplo, o chip *Willow* (GoogleQ, 2024), desenvolvido pela Google, apresentou uma redução exponencial no acúmulo de erros à medida que o número de *qubits* aumentava. Além disso, a Microsoft (Microsoft, 2025) propôs uma nova abordagem de implementação baseada em férmions de Majorana<sup>1</sup>, possibilitando a construção de *qubits* mais confiáveis e escaláveis. Esses avanços no hardware mencionados anteriormente, têm sido impulsionados tanto pelo aprimoramento de códigos de correção de erros quanto por pesquisas voltadas ao desenvolvimento de novas arquiteturas de hardware, com o objetivo de aumentar o tempo de coerência dos *qubits* e, conseqüentemente, a viabilidade de sistemas quânticos em larga escala.

No entanto, os algoritmos desenvolvidos para computação quântica ainda se concentram na utilização de chips com número reduzido de *qubits* e com tolerância a ruídos limitada. Esses chips são conhecidos como dispositivos quânticos ruidosos de escala intermediária *NISQ* (do inglês, *Noisy Intermediate-Scale*

---

<sup>1</sup> Férmions de Majorana: Férmions neutros idênticos à sua antipartícula, propostos por Ettore Majorana. Embora não observados como partículas fundamentais, quasipartículas com propriedades de Majorana foram detectadas em sistemas de matéria condensada (e.g., supercondutores topológicos), sendo candidatas a *qubits* topológicos em computação quântica (Aguado; Kouwenhoven, 2020).

*Quantum*) e, geralmente são aplicados em tarefas específicas com a finalidade de encontrar alguma vantagem computacional (Preskill, 2018). Pesquisas desenvolvidas nos últimos anos apresentaram resultados promissores ao executar algoritmos quânticos em processadores NISQ por meio da implementação de algoritmos baseados em técnicas como Simulated Annealing (Dwave, 2024) e algoritmos estruturados na forma de Circuitos Quânticos Parametrizados (*PQC*, do inglês *Parameterized Quantum Circuits*<sup>2</sup>) (Horvat et al., 2022), compostos por um número restrito de operações e camadas (Horvat et al., 2022). Assim, há um esforço contínuo na proposição de novos algoritmos quânticos capazes de explorar possíveis ganhos de desempenho em relação aos métodos clássicos, muitos dos quais são validados por meio de simuladores que simulam o comportamento de dispositivos quânticos com tolerância a ruídos (Biamonte et al., 2017).

Nesse contexto, várias aplicações são desenvolvidas na área da computação quântica. Dentre elas se destaca a Aprendizagem de Máquina Quântica — *QML* (do inglês, *Quantum Machine Learning*). Essa aplicação investiga a utilização de algoritmos quânticos para otimizar modelos clássicos de aprendizado de máquina. Diversos estudos, tanto teóricos quanto experimentais, têm demonstrado o potencial da *QML* em diferentes setores da economia, indicando que o uso de algoritmos quânticos em tarefas específicas pode melhorar significativamente o desempenho de modelos clássicos. Por exemplo, o trabalho prático de Naguleswaran (Naguleswaran, 2024) investiga funções de *kernel* calculadas por algoritmos quânticos que podem ser empregados para potencializar algoritmos como Máquina de Vetores Suporte - *SVM* (do inglês, *Support Vector Machine*) e redes neurais convolucionais *CNN* (do inglês, *Convolutional Neural Networks*),

---

<sup>2</sup> Essa abordagem é conhecida como computação quântica universal. Ela é mais comum para implementar algoritmos quânticos conforme discutido no Capítulo 2.

explorando condições nas quais uma vantagem quântica é alcançada, especialmente por meio de *kernels* projetados de forma dependente dos dados. Outro estudo que vale à pena ser ressaltado é o estudo teórico desenvolvido por Schuld (Schuld, 2021) que apresenta uma formulação matemática que evidencia as vantagens do mapeamento de dados em espaços de alta dimensionalidade por meio de algoritmos quânticos variacionais, o que favorece a construção de superfícies de generalização e melhora o desempenho de algoritmos como as Máquinas de Vetores de Suporte Quântico — QSVM (do inglês, *Quantum Support Vector Machine*).

Algoritmos quânticos podem oferecer vantagens quando aplicados a problemas de aprendizado de máquina, especialmente em cenários caracterizados por conjuntos de dados com elevado grau de desbalanceamento e alta dimensionalidade. Um exemplo típico de aplicação com essas características é a detecção de falhas em sistemas e equipamentos industriais, conforme apontado nos estudos de Abidi et al. (Abidi et al., 2022) e Zeguendry et al. (Zeguendry et al., 2023). Essas pesquisas evidenciam que a implementação adequada de estratégias de detecção de falhas, aliada à disponibilidade de recursos para a aquisição de dados, contribui significativamente para a adoção de práticas de manutenção preditiva, reduzindo custos operacionais e evitando paradas inesperadas.

No setor aeronáutico, por exemplo, a detecção de falhas é empregada para o monitoramento contínuo das condições de componentes críticos das aeronaves (Chowdhury et al., 2023). No setor de manufatura, essa técnica tem sido utilizada para o acompanhamento de equipamentos como máquinas de Controle Numérico por Computador e robôs industriais, possibilitando intervenções antecipadas, conforme discutido por (Gupta et al., 2023), que destacam a integração

com sistemas baseados em Internet das Coisas. Em usinas de geração de energia, algoritmos de detecção de falhas vêm sendo aplicados ao monitoramento de turbinas, geradores e outros componentes essenciais, promovendo melhorias na gestão da manutenção, como evidenciado por (Salehi, 2023).

## 1.1 Problemática e questão de pesquisa

O desenvolvimento de estratégias de detecção de falhas pode trazer diversas vantagens para vários setores industriais. No entanto, ainda enfrentam alguns desafios devido às limitações e a dificuldade de manipular conjunto de dados com múltiplas colunas ou variáveis, *features*, conforme também abordado nos trabalhos de Abidi et al. (Abidi et al., 2022) e Zeguendry et al. (Zeguendry et al., 2023). Além disso, ao trabalhar com conjunto de dados com alto grau de desbalanceamento e dimensionalidade como é característico dos problemas de detecção de falhas, é necessário realizar um pré-processamento criterioso dos dados para evitar que o modelo enfrente dificuldades para generalizar a identificação de novas falhas ao utilizar os modelos convencionais de ML.

De acordo com Schuld (Schuld, 2021), algoritmos quânticos possuem a característica de atuar como funções de *kernel* capazes de mapear dados para espaços vetoriais de alta dimensionalidade. Esse mapeamento possibilita a identificação de superfícies de generalização com menor custo computacional, superando limitações observadas em *kernels* clássicos. Nesse contexto, o uso de algoritmos quânticos como funções de *kernel* em modelos de Máquinas de Vetores de Suporte (SVM) representa uma alternativa promissora, ao incorporar paradigmas de processamento baseados nos princípios da mecânica quântica. Essa abordagem pode ser particularmente vantajosa na detecção de falhas em sistemas industriais compostos por múltiplos componentes, onde a falha pode

estar associada a alterações sutis em variáveis específicas. Diante disso, surge a seguinte questão de pesquisa:

- **Como o desempenho de uma estratégia híbrida quântico-clássica, composto por um classificador baseado em Máquinas de Vetores Suporte com um *kernel* calculado por um circuito quântico parametrizado se comporta para identificar e classificar falhas com base em um conjunto de dados com alta dimensionalidade e desbalanceamento?**

Diante do exposto, a investigação proposta busca explorar o potencial dos *kernels* quânticos como alternativa para superar os desafios impostos por dados complexos na detecção de falhas. Para investigar essa abordagem, estabelecem-se os seguintes objetivos:

## 1.2 Objetivos

O objetivo geral desta dissertação de mestrado consiste em propor uma estratégia de detecção de falhas utilizando um algoritmo de máquinas de vetores suporte aprimorado por uma função de *kernel* quântica. A partir disso, serão analisados possíveis cenários de sua utilização, com intuito de discutir as vantagens de detecção de falhas usando um modelo híbrido quântico-clássico. Durante o desenvolvimento desse estudo, emergirão questões críticas relacionadas à aplicabilidade e eficácia dessas estratégias: Quais são os limites das abordagens atuais de detecção de falhas? Como validar a eficácia dos modelos de classificação quântica propostos na detecção de falhas? Essas indagações levam à orientar a formulação dos objetivos específicos da pesquisa, que incluem:

1. **Desenvolver modelos de Máquinas de Vetores Suporte compostos por funções de *kernel* quânticas e clássicas;**



2. **Desenvolver regressores para prever valores de cada feature em um determinado lag de tempo e os modelos QSVM implementados classificar se há falha ou não;**
3. **Avaliar e comparar o desempenho dos modelos em relação às métricas obtidas após a realização dos testes, considerando tanto a precisão dos resultados quanto a eficiência do processamento para cada abordagem;**
4. **Analisar as implicações práticas e os possíveis cenários de utilidade dos modelos.**

### 1.3 Organização da Dissertação

A presente dissertação encontra-se dividida em 6 capítulos, listados a seguir:

- No Capítulo 1 são estabelecidos o contexto da pesquisa, a motivação do estudo, a definição do problema, os objetivos gerais e específicos.
- No Capítulo 2 são explorados os fundamentos teóricos que sustentam a pesquisa. Aborda-se a computação quântica, incluindo seus princípios básicos, modelos de computação e algoritmos relevantes. Em seguida, discute-se a aprendizagem de máquina quântica, destacando como os algoritmos quânticos podem ser aplicados a problemas de aprendizado de máquina, com ênfase em classificadores quânticos e suas vantagens potenciais sobre métodos clássicos.
- No Capítulo 3 é apresentada uma revisão abrangente da literatura relacionada à manutenção preditiva. São discutidas as técnicas tradicionais

utilizadas bem como os desafios enfrentados, como a complexidade dos sistemas e a necessidade de detecção precoce de falhas. Além disso, explora-se como a computação quântica pode contribuir para superar essas limitações, oferecendo novas abordagens para o diagnóstico e prognóstico de falhas.

- No Capítulo 4 detalha-se a abordagem metodológica adotada na pesquisa. Descreve-se o processo de coleta e pré-processamento dos dados operacionais das turbinas eólicas, a seleção das variáveis relevantes e a construção dos modelos de aprendizado de máquina. São apresentados tanto o modelo clássico quanto o modelo quântico, incluindo suas arquiteturas, algoritmos utilizados e parâmetros de treinamento. Além disso, discute-se as estratégias de validação e as métricas de desempenho empregadas para avaliar os modelos.
- No Capítulo 5 são apresentados os resultados obtidos a partir da aplicação dos modelos desenvolvidos. Compara-se o desempenho dos modelos clássico e quântico em termos de acurácia, precisão, *recall*, *F1-score*. Analisa-se a eficácia de cada modelo na detecção de falhas em diferentes componentes da turbina eólica, considerando cenários com e sem redução de dimensionalidade. Os resultados são discutidos baseados nas hipóteses formuladas e das contribuições potenciais para a área de manutenção preditiva.
- No Capítulo 6 serão apresentadas as considerações finais dessa dissertação e os trabalhos futuros.

## 2

---

## APRENDIZAGEM QUÂNTICA DE MÁQUINA

**E**ste capítulo apresenta a fundamentação teórica necessária para uma compreensão aprofundada do trabalho proposto: a Seção 2.1 explora os principais conceitos de computação quântica que são necessários para o entendimento dessa pesquisa, os tipos de arquiteturas de computação quântica e os desafios relacionados à sua aplicação no contexto deste estudo, enfatizando algumas vantagens do uso de simuladores de sistemas quânticos para executar algoritmos quânticos. Por último, a Seção 2.2 revisa os modelos de aprendizagem de máquina, englobando abordagens clássicas, quânticas e híbridas.

### 2.1 Computação quântica

Os Computadores quânticos utilizam *qubits* para processar informações. Ao contrário dos computadores clássicos, que seguem a física newtoniana, os *qubits* podem assumir uma superposição de estados e se emaranharem entre si, o

que amplia a capacidade de processamento em busca de soluções ótimas para determinados problemas (Nielsen et al., 2010). No modelo de circuitos quânticos, os *qubits* são inicializados no estado computacional  $|0\rangle$ , e uma sequência de operações unitárias, que são operações que preservam a métrica do espaço vetorial e portanto garantem que a mudança no estado também seja um estado quântico, faz o sistema evoluir para um estado final, permitindo a extração da solução do problema. As operações em *qubits* são implementadas no nível de hardware por pulsos de controle, como micro-ondas para *qubits* supercondutores e pulsos de laser para íons aprisionados, sendo a precisão desses pulsos essencial para manipular os estados quânticos. Essas operações são representadas por matrizes unitárias  $U$  que atuam sobre o estado de um ou mais qubits, descrito por um vetor de estado em um espaço de Hilbert. Tais matrizes correspondem a portas lógicas quânticas e a composição sequencial dessas operações, dadas pelo produto de matrizes  $U_k, \dots, U_1$ , define um circuito quântico.

Os detalhes matemáticos dessa manipulação, que pode ser entendida como mudanças no estado quântico que descreve o *qubit* através dessas operações unitárias, serão abordados na subseção 2.1.1, enquanto a subseção 2.1.2 explorará os diferentes ambientes de programação de hardwares reais de computação quântica e simuladores quânticos.

### 2.1.1 Manipulação de *qubits*

O formalismo matemático para o processamento de informação quântica fundamenta-se na representação de estados por vetores unitários e operações por transformações unitárias em um espaço de Hilbert complexo. Em outras palavras, o processamento da informação quântica é formalmente descrito de acordo com a álgebra linear sobre espaços de Hilbert complexos. Um estado de um *qubit* é

definido por um vetor unitário escrito numa determinada base vetorial,  $(|0\rangle)|1\rangle$ , onde suas amplitudes correspondem à números complexos. A soma do módulo quadrático deve ser 1 (vetor normalizado), devido à natureza probabilística da mecânica quântica garantindo que o vetor esteja normalizado (Nielsen et al., 2010).

A diferença fundamental entre bits clássicos e *qubits* reside em seus espaços vetoriais: enquanto o primeiro opera em um espaço discreto binário 0,1, tipicamente implementado por níveis de tensão em circuitos digitais, o *qubit*, como sistema quântico de dois níveis, habita um espaço de Hilbert bidimensional complexo (Mermin, 2007). Essa natureza quântica permite a existência de superposições coerentes de estados, em que a diferença de fase entre os estados base permanece constante, conforme descrito pela notação de Dirac na Equação 2.1.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \text{ como } |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ e } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow \quad (2.1)$$

$$|\psi\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

A notação de Dirac (bra-ket) fornece uma representação abstrata e independente de base para estados quânticos, onde vetores de estado (kets,  $|\cdot\rangle$ ) e seus duais (bras,  $\langle\cdot|$ ) permitem expressar operações e produtos internos de forma concisa, sem necessidade de representação matricial explícita, embora nas aplicações esta representação será utilizada. A notação  $|\cdot\rangle$ , chamada de **ket**, representa vetores coluna, enquanto  $\langle\cdot|$ , chamada de **bra**, representa vetores linha que pertencem a espaços de Hilbert diferentes. O **bra**  $\langle\cdot|$  é obtido a partir do **ket**  $|\cdot\rangle$  por transposição conjugada, indicada por  $\dagger$ , ou seja,  $|\cdot\rangle^\dagger = \langle\cdot|$ .

Para **estados quânticos normalizados**, a relação  $|\alpha|^2 + |\beta|^2 = 1$  deve ser satisfeita, com  $\alpha, \beta \in \mathbb{C}$ , cujos módulos ao quadrado representam as **amplitudes de probabilidade** (Griffiths et al., 2011). É importante destacar que, como  $\alpha$  e  $\beta$  pertencem ao conjunto dos números complexos, o espaço  $\mathbb{C}^2$  possui quatro graus de liberdade — as partes real e imaginária de cada coeficiente.

Um exemplo de estado superposto assumido por um sistema quântico é mostrado na Equação 2.2, onde  $\alpha = \beta = \frac{1}{\sqrt{2}}$ . Nesse caso, a probabilidade de o estado superposto colapsar para o estado  $|0\rangle$  ou  $|1\rangle$  após uma medida é dada pelo **módulo quadrático** dos coeficientes  $\alpha$  e  $\beta$ , isto é,  $|\alpha|^2 = |\beta|^2 = \frac{1}{2}$ .

$$|\psi\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad (2.2)$$

Os estados quânticos pertencem ao espaço vetorial de Hilbert, que pode ser finito ou infinito-dimensional, dependendo do sistema (Griffiths et al., 2011). Os *qubits*, por serem sistemas quânticos de dois níveis, pertencem a um espaço vetorial complexo de dimensão finita  $\mathbb{C}^2$ . Seus estados puros podem ser representados geometricamente na esfera de Bloch por meio de um vetor  $|\psi\rangle$  que aponta para tal estado, como mostrado na Figura 1 (Mermin, 2007).

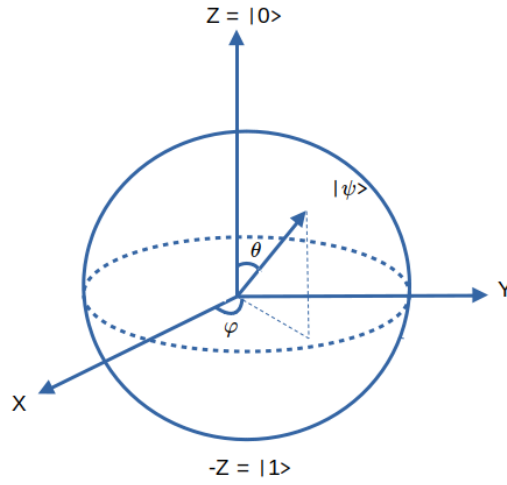


Figura 1 – Esfera de Bloch com um vetor  $|\psi\rangle$  apontando para um estado de um *qubit*.

Note que na Figura 1 o estado quântico  $|\psi\rangle$  pode ser representado por um vetor que aponta para um ponto na superfície da esfera. Portanto, uma alternativa para representar um QuBit é por meio de uma parametrização que utiliza os ângulos  $\theta$  (ângulo medido entre o eixo Z e o vetor, correspondente ao eixo polar) e  $\phi$  (ângulo medido no plano xy, correspondente ao eixo equatorial) da esfera. Essa parametrização é expressa pela equação 2.3.

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \quad (2.3)$$

Observe que se a fase  $\phi$  for múltiplos de  $0, 2\pi, \dots, 2n\pi$  para  $n \in \mathbb{Z}$  e  $\theta = \pi/2$ , o estado  $|\psi\rangle$  assumirá a forma representada na equação 2.3. O ângulo  $\theta$ , correspondente ao eixo polar da esfera de Bloch, é medido entre o eixo Z e o vetor de estado  $|\psi\rangle$ . O ângulo  $\phi$ , eixo equatorial da esfera de Bloch é medido no plano xy. Estes dois parâmetros são utilizados para descrever a posição de um

estado quântico na Esfera de Bloch (Nielsen et al., 2010).

Dado um estado quântico  $|\psi\rangle$  de um QuBit e uma determinada operação neste estado, representada por  $\mathcal{O}$ , afirmar-se que quando um operador atuar em  $|\psi\rangle$  este estado sofrerá uma transformação linear ou rotação na esfera de Bloch. Matematicamente, isto pode ser representado pela equação 2.4.

$$\mathcal{O}|\psi\rangle = |\psi'\rangle \quad (2.4)$$

Quando um sistema quântico é composto por  $n$  qubits, o estado total do sistema pode ser representado por um produto tensorial na forma  $|\psi\rangle^{\otimes n}$ . Esse sistema pertence a um espaço vetorial de dimensão  $2^n$ , e a relação entre os qubits é expressa por meio de produtos tensoriais. Assim, tem-se:  $|\psi\rangle^{\otimes n} = |\psi_n\rangle \otimes \dots \otimes |\psi_1\rangle \otimes |\psi_0\rangle = |\psi_n \dots \psi_1 \psi_0\rangle$  (IBMQ, 2023).

Um operador é simplesmente uma matriz que ao ser aplicado ao vetor de estado  $|\psi\rangle$ , faz com que a esfera de Bloch rotacione, fazendo-o apontar para um outro estado quântico. A partir disto, suponha que se tenha um operador genérico  $\mathcal{O} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ , ao aplicá-lo em  $|\psi\rangle$ , resultará na Equação 2.5.

$$\mathcal{O}|\psi\rangle = \begin{bmatrix} a & b \\ c & d \end{bmatrix} * \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} (\alpha a + \beta b) \\ (\alpha c + \beta d) \end{bmatrix} \quad (2.5)$$

Fazendo  $(\alpha a + \beta b) = \alpha'$  e  $(\alpha c + \beta d) = \beta'$  e definindo  $|\psi'\rangle = \begin{bmatrix} \alpha' \\ \beta' \end{bmatrix}$  se tem a relação representada pela Equação 2.4.

Isto é necessário para entender como os algoritmos quânticos universais costumam funcionar. Estes são representados por circuitos, semelhante ao mostrado na Figura 2.



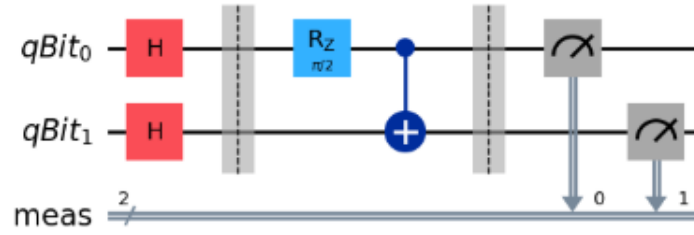


Figura 2 – Modelo de algoritmo quântico baseado em circuito lógico.

Observe que o **sistema quântico** inicial deste circuito possui dois QuBits ( $\text{qBit}_0$  e  $\text{qBit}_1$ ) ambos inicializados no estado  $|0\rangle$ . A primeira parte do circuito prepara o estado, a segunda parte processa o estado e a terceira mede o estado. Os **operadores** ao serem aplicados no estado inicial fará com que este sofra rotações na esfera de Bloch. A porta lógica **Hadamard - H** é uma das principais operações para se obter vantagem quântica, pois ela faz com que o sistema inicial entre em superposição. Inicialmente se tem a seguinte configuração.

$$|\psi\rangle = |0\rangle \otimes |0\rangle \quad (2.6)$$

Após isso aplica-se as **portas Hadamards** em cada QuBit, logo:

$$H|\psi\rangle = H|0\rangle \otimes H|0\rangle \quad (2.7)$$

A porta lógica H é equivalente ao seguinte operador  $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ . Ao aplicar essa matriz no estado  $|0\rangle$  resultará em um estado superposto representado pela equação 2.8:

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}. \quad (2.8)$$

Portanto na primeira parte do circuito, obtem-se o estado representado pela equação 2.9.

$$\begin{aligned} |\psi\rangle &= \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ &= \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2} \end{aligned} \quad (2.9)$$

Observe que cada estado possui a probabilidade de  $|\frac{1}{2}|^2$  para ser medido. Ao aplicar a porta H cada QuBit foi rotacionado em  $\frac{\pi}{2}$  em torno do eixo Y. (IBMQ, 2023)

Em seguida, é aplicada uma **porta de rotação parametrizada em torno do eixo x da esfera e Bloch** no *qubit* 0. Esta operação rotaciona o estado do *qubit* 0 por um ângulo definido, variando o estado conforme o parâmetro de rotação  $\theta$ . Por fim, é aplicada uma **porta CNOT** (do inglês: *Controlled-NOT*), com o *qubit* 0 funcionando como o *qubit* de controle e o *qubit* 1 como o *qubit* alvo. Isso significa que, se o *qubit* de controle estiver no estado  $|1\rangle$ , a operação *Not* será aplicada no *qubit* 1, trocando seu estado de  $|0\rangle$  para  $|1\rangle$ , ou vice-versa. Na equação 2.10 é mostrado os cálculos da evolução dos *qubits* desde o estado de mais baixa energia até o estado resultante<sup>1</sup>.

$$\begin{aligned} |\psi\rangle &= |0\rangle_1 \otimes |0\rangle_0 \\ |\psi\rangle &= H|0\rangle_1 \otimes H|0\rangle_0 = \left( \frac{|0\rangle_1 + |1\rangle_1}{\sqrt{2}} \right) \otimes \left( \frac{|0\rangle_0 + |1\rangle_0}{\sqrt{2}} \right) \\ &= \left( \frac{|0\rangle_1 + |1\rangle_1}{\sqrt{2}} \right) \otimes R_z\left(\frac{\pi}{2}\right) \left( \frac{|0\rangle_0 + |1\rangle_0}{\sqrt{2}} \right) \end{aligned} \quad (2.10)$$

Dado que  $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$  e  $R_z\left(\frac{\theta}{2}\right) = \begin{pmatrix} e^{-i\frac{\pi}{4}} & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$

$$\Rightarrow |\psi\rangle = \frac{1}{\sqrt{2}} (|0\rangle_1 + |1\rangle_1) \otimes \frac{e^{-i\pi/4}|0\rangle + e^{i\pi/4}|1\rangle}{\sqrt{2}}$$

<sup>1</sup> No apêndice 1 são mostradas as matrizes das portas e os efeitos delas nos estados dos *qubits*

É importante sanar que embora todas as portas rotacione o QuBit na esfera de Bloch, o nome das portas RX, RY e RZ também são chamadas de **portas de rotação**. As portas que representam os operadores nos algoritmos quânticos são derivadas das matrizes de Pauli que são  $\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ ,  $\sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$  e  $\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$  e a matriz identidade  $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ . No caso das portas de Rotação a equação 2.11 indica o efeito de tais portas.

$$R_i = e^{-i\frac{\theta}{2}\sigma_i} \text{ tal que } i = x, y, z \quad (2.11)$$

Por isso que, a partir desta relação, a matriz  $R_z$ , por exemplo, pode ser definida como  $R_z = e^{i\frac{\theta}{2}\sigma_z}$ . Esta relação pode ser calculada utilizando a identidade denotada pela equação 2.12.

$$\begin{aligned} R_z(\theta) &= \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Z \\ &= \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \end{aligned} \quad (2.12)$$

A última etapa do circuito é a **medição** do estado. Por exemplo, ao medir 1024 vezes o circuito representado na Figura 2, todos os estados armazenados no sistema  $|\psi\rangle$  têm uma probabilidade aproximadamente igual de  $P(|00\rangle) = P(|01\rangle) = P(|10\rangle) = P(|11\rangle) = \left|\frac{1-i}{2\sqrt{2}}\right|^2$  de colapsar para um desses estados após a medição.

O histograma representado na Figura 3 mostra a distribuição de probabilidade obtida para 1024 medidas.

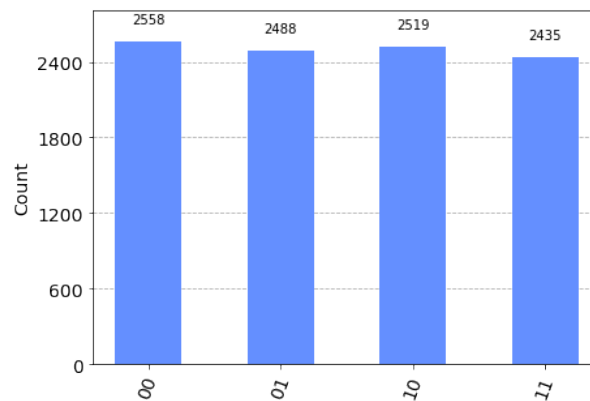


Figura 3 – Estatística dos estados colapsados após 1024 medidas realizadas no circuito quântico da Figura 2. Cada estado com aproximadamente o mesmo número de ocorrências.

### 2.1.2 Computadores quânticos reais e simulados

Em 1982, Richard Feynman (Feynman, 1982) introduziu a ideia de utilizar sistemas quânticos como unidades de informação, estabelecendo as bases teóricas para a computação quântica. Desde então, avanços significativos têm sido alcançados na construção de dispositivos quânticos reais, com *qubits* implementados em diversos chips físicos com diferentes arquiteturas. A Tabela 1 mostra os principais tipos de *qubits* utilizados em computadores quânticos baseados em diferentes abordagens físicas.

Empresa	Tipo de <i>qubit</i>	Descrição
IBM, IQM e Google	<i>qubits</i> Supercondutores	Utiliza circuitos supercondutores e operam em temperaturas criogênicas para reduzir ruídos e decoerência.
Xanadu <sup>2</sup>	<i>qubits</i> Fotônicos	Utilizam fótons como <i>qubits</i> , manipulados por circuitos ópticos.
IonQ <sup>3</sup> e Quantinuum	<i>qubits</i> Iônicos	Utiliza íons aprisionados, confinados por meio de campos eletromagnéticos.
QuERA <sup>4</sup> e Pasqal	<i>qubits</i> de Átomos Neutros	Utiliza átomos neutros controlados por lasers e armadilhas ópticas, permitindo alta escalabilidade e precisão.

Tabela 1 – Arquiteturas reais de Computação Quântica.

A diversidade de abordagens físicas para implementação de *qubits* reflete o caráter exploratório atual da computação quântica. Os *qubits* supercondutores, adotados por líderes como IBM (IBMQ, 2023) e Google (Google Quantum, 2023), destacam-se pela integração com tecnologias de fabricação de semicondutores convencionais, porém exigem infraestrutura criogênica complexa abaixo de 15 mK para manter a coerência quântica e o estado de supercondução. Já os *qubits* fotônicos da Xanadu (Xanadu, 2023) operam em temperatura ambiente e permitem interconexões via fibra óptica, porém enfrentam desafios na detecção eficiente de fótons individuais.

Na categoria de íons aprisionados, soluções como as da IonQ (IonQ, 2025) alcançam tempos de coerência excepcionais (tempo durante o qual as características quânticas do sistema se mantêm) e portas lógicas com alta fidelidade com um valor maior que 99.9%, embora necessitem de sistemas de vácuo ultra-alto e controle de lasers de precisão. Por fim, os átomos neutros adotados pelas empresas QuERA (QuERA, 2024) e Pasqal (Pasqal, 2025) emergem como plataforma promissora para escalabilidade tridimensional, onde arranjos ópticos permitem reconfiguração dinâmica de *qubits* durante a computação.

Esta fragmentação tecnológica evidencia a ausência de um hardware quântico dominante, direcionando investimentos para aplicações específicas: supercondutores para integração em data centers, fótons para redes quânticas, e átomos neutros para simulação de materiais.

Utilizar hardwares quânticos atualmente é acessível via nuvem (Amazon Braket, 2023; IBMQ, 2023), todavia a mitigação do ruído intrínseco permanece um desafio crítico, comprometendo a precisão dos resultados. O ruído intrínseco decorre da sensibilidade dos *qubits* a fatores externos, como temperatura e

vibrações, além de interações indesejadas que aumentam com a escalabilidade do sistema. Para viabilizar aplicações práticas, é essencial o aprimoramento de códigos corretores de erros quânticos. Avanços recentes incluem o chip Willow da Google (GoogleQ, 2024), que demonstra maior resistência a ruídos por meio de técnicas aprimoradas de correção de erros (PAN et al., 2023). Além disso, essa empresa tem apresentado avanços significativos no desenvolvimento de códigos corretores de erros, conforme indicado em (IQM, 2024), e projeta alcançar a era da *fault tolerance* nos próximos cinco anos.

Circuitos quânticos podem ser implementados em diversos frameworks que permitem sua execução em dispositivos quânticos reais (Amazon Braket, 2023). Cada operação sobre *qubits* é realizada por pulsos eletromagnéticos, caracterizados por frequência e amplitude, que induzem a evolução do estado quântico. A execução em hardware real exige a transpilação (Nielsen et al., 2010), processo que adapta o circuito à arquitetura do dispositivo. Esse processo envolve o mapeamento de *qubits* lógicos para *qubits* físicos, a tradução de portas quânticas de alto nível para operações nativas do dispositivo real e a otimização do circuito para minimizar ruídos e erros.

Geralmente a ligação entre os *qubits* no dispositivo quântico real é representada através de um tipo de Grafo chamado *Chimera Graph*. A Figura 4, mostra a ligação entre os 127 *qubits* de um dispositivo quântico (IBMQ, 2023).

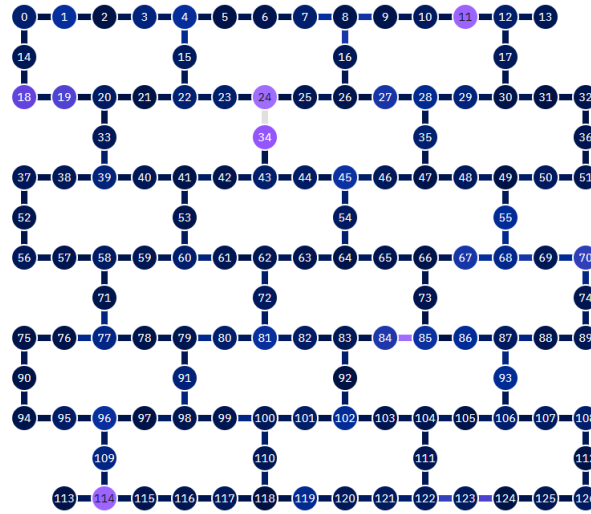


Figura 4 – Esquema de conexão entre os 127 *qubits* de um dispositivo quântico supercondutor.

O processo de execução de um circuito quântico em um dispositivo quântico real, passará pelas etapas descritas acima. Por exemplo, o circuito representado na Figura 2 ao ser executado nesse dispositivo passará pela etapa de mapeamento dos *qubits* de acordo com o grafo da Figura 4. Após isso, as portas serão traduzidas para as portas nativas desse dispositivo que, nesse caso, são: ECR, Identidade, Rotação em torno do eixo Z, SX e X. Por último, a etapa de otimização que cuidará do número de operações do circuito ao gerar o circuito da Figura 4. Portanto, o circuito da Figura 2 vai ser transpilado para um circuito equivalente representado na Figura 5 (IBMQ, 2023).

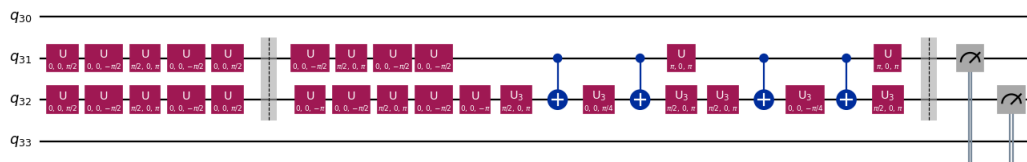


Figura 5 – Algoritmo quântico adaptado para as portas nativas do dispositivo quântico.

Embora a execução de algoritmos quânticos em hardware real seja crucial para investigar o comportamento dos circuitos sob ruído e avaliar seu desempenho, os dispositivos atuais ainda não contam com códigos de correção de erros plenamente desenvolvidos (Park et al., 2024). Essa limitação compromete a consistência dos resultados e eleva significativamente os custos computacionais. Enquanto a implementação de mecanismos mais eficazes de correção de erros permanece um desafio prático, simuladores quânticos que utilizam GPUGPUs (do inglês, *Graphical Processing Unit*) e TPUTPUs (do inglês, *Tensor Processing Unit*) continuam sendo alternativas viáveis para benchmarking e otimização de algoritmos, permitindo sua execução em cenários com ou sem ruído.

## 2.2 Aprendizagem de máquina

**Aprendizagem de Máquina - ML (Machine Learning)** é definida como um campo de estudo que capacita sistemas a adquirirem conhecimento empírico por si próprios, ao extrair padrões de dados brutos (Goodfellow et al., 2016). Existem duas categorias principais de aprendizagem: **(i) Aprendizagem Supervisionada** que ocorre quando um modelo aprende a relação entre uma variável  $Y$  e uma instância  $\vec{X}$  e **(ii) Não Supervisionada** que ocorre quando o modelo "aprende" um determinado tipo de padrão entre as variáveis contidas em uma base de dados  $\vec{X}$  (Petrucione et al., 2018).

Nesta pesquisa, foram utilizados dois algoritmos de aprendizagem supervisionada que compõem a estratégia de detecção de falhas. Portanto as subseções seguintes farão uma revisão dos fundamentos teóricos dos modelos de **Regressão** e modelos quânticos de máquinas de vetores suporte - QSVC (do inglês, *Quantum Support Vector Classification*). O algoritmo QSVC será discutido na última subseção por se tratar de um algoritmo híbrido composto por um algoritmo



clássico de otimização e um algoritmo quântico que desempenhará o papel de função de kernel.

### 2.2.1 Modelos de aprendizagem de máquina convencional

Os modelos de aprendizagem de máquina utilizados no desenvolvimento dessa dissertação são: regressores e os modelos de SVM.

#### 1. Regressores

Em ML, a Regressão é um tipo de algoritmo de aprendizado supervisionado que calcula a relação entre os tipos de variáveis através de um ajuste dos parâmetros de uma equação de acordo com os dados observados. Quando há apenas uma característica (do inglês, *feature*) independente, a regressão é simples, e quando há mais de uma *feature*, é conhecida como regressão múltipla. Da mesma forma, quando há apenas uma variável dependente, é considerada regressão univariada, quando se tem mais de uma variável dependente, é considerada regressão multivariada (Goodfellow et al., 2016).

Seja a seguinte regressão representada pela equação  $\hat{y} = X\vec{w}$  que possui um vetor  $\vec{X}$  de dimensão  $m \times n$  e um vetor de pesos ajustáveis  $\vec{w}$  de dimensão  $n \times 1$ , onde  $n$  é a quantidade de características independentes e  $m$  é a quantidade de instâncias do conjunto de dados. Ao calcular os valores estimados  $\hat{y}$  para todas as  $m$  instâncias utilizando os pesos  $w$ , é necessário avaliar o quanto o estimador está errando. Essa avaliação é feita por meio de uma função de perda, que pode ser representada pelo **erro médio quadrático** (MSE, do inglês *mean squared error*), conforme mostrado na Equação 2.13 (Goodfellow et al., 2016).

$$e = \frac{1}{m} \sum_i^m (\hat{y}_i(w) - y_i)^2 \quad (2.13)$$

O objetivo da regressão é minimizar o erro  $e$  utilizando um algoritmo de otimização. Isso significa que, ao minimizar o erro médio, será estimado o vetor de coeficientes  $w$ , que, ao ser multiplicado pelos valores das variáveis independentes  $X$ , fornecerá a estimativa do valor da variável dependente. Em geral, esse processo de ajuste de pesos é chamado de treinamento do modelo de aprendizagem de máquina.

## 2. Máquinas de vetores suporte

SVM é uma técnica robusta que pode ser aplicada em problemas de classificação que tem como base encontrar um hiperplano que separe todas as características pertencentes ao banco de dados da melhor forma num espaço  $n$ -dimensional (Santos et al., 2002). A Figura 6 mostra a abordagem que o algoritmo SVM utiliza para um banco de dados com duas dimensões, onde  $x_1$  e  $x_2$  são duas features de uma base de dados genérica (Chollet, 2021).

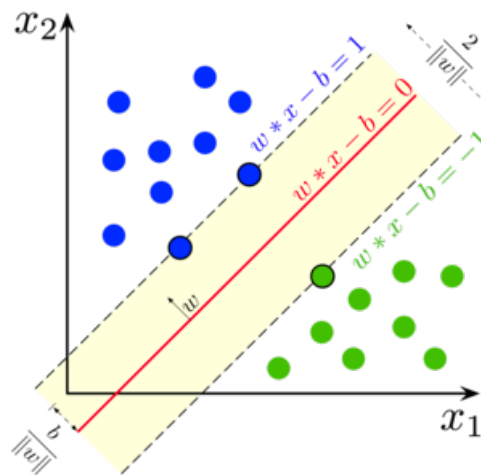


Figura 6 – Fronteira de decisão resultante do treinamento do classificador SVM, evidenciando os vetores de suporte que definem a margem máxima entre as classes representadas pelas cores azul e verde.

Em um problema de classificação em duas dimensões, com dois grupos (verde e azul), o objetivo é encontrar uma reta que maximize a distância entre esses grupos. O vetor  $\vec{w}$  é perpendicular a essa reta e possui uma norma  $\|\vec{w}\|$ . Para maximizar a separação entre os grupos, é equivalente a maximizar  $\|\vec{w}\|$  ou, de forma equivalente, minimizar a distância entre os dois hiperplanos de margens, dada por  $\frac{2}{\|\vec{w}\|}$ , sujeito à restrição 2.14.

$$y_i(w^T * x_i + b) \geq 1, \forall 0 \leq i \leq n. \quad (2.14)$$

Onde  $y_i$  é a classe que a instância  $x_i$  pertence,  $w$  é o vetor normal a reta, e  $b$  é o viés.

Uma das principais vantagens desse método é que seu processo de treinamento não se baseia no método do gradiente descendente<sup>5</sup>. Por isso, ele não corre o risco de ficar preso a mínimos locais, o que é um problema comum em métodos não convexos. O problema que o SVM busca resolver é um problema de otimização convexa, o que garante que a solução encontrada será globalmente ótima. O SVM depende da distribuição dos dados para encontrar uma reta ou hiperplano ótimo que separe as classes de um conjunto de dados de forma adequada (Chollet, 2021). A Figura 7 ilustra a distribuição de um conjunto de dados de difícil separabilidade, encontrar uma superfície que separe tais classes de forma eficiente é um problema complexo, portanto o SVM utiliza um recurso adicional chamado função de kernel.

<sup>5</sup> Método de otimização utilizado para encontrar um mínimo local de uma função.

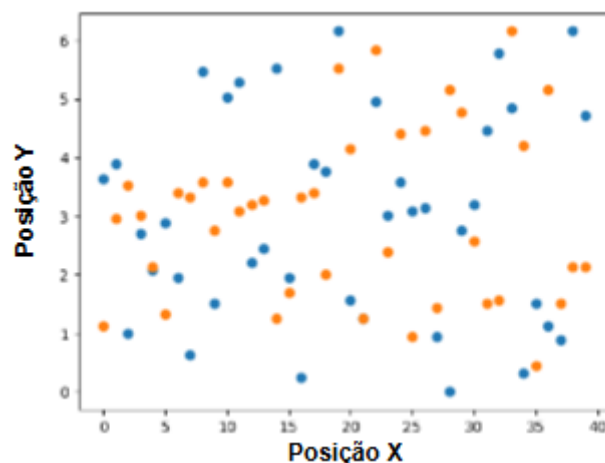


Figura 7 – Conjunto de dados artificialmente construído com classes de difícil separação linear.

As funções de Kernel, por sua vez, permitem ao SVM lidar com problemas não lineares, mapeando os dados de entrada para um espaço de características de maior dimensionalidade, (do inglês, *feature map*), onde é mais provável encontrar um hiperplano de separação. Diversos tipos de funções de kernel são utilizados no SVM para realizar tal mapeamento, como as funções de kernel polinomiais, as funções de kernel de base radial e as funções de kernel sigmóides, permitindo que o SVM encontre hiperplanos de separação quando os dados não são linearmente separáveis no espaço original (Chollet, 2021). Assim, a função de kernel é um recurso essencial do SVM para lidar com problemas de classificação não lineares, permitindo que o algoritmo encontre uma superfície de generalização ótima em um espaço de características de maior dimensionalidade.

### 2.2.2 Modelos de aprendizagem quântica de máquina

Os princípios fundamentais da QC têm o potencial de aprimorar o desempenho dos modelos de ML em vários casos, uma vez que dados podem ser preparados e

processados de forma muito mais eficiente do que nos algoritmos convencionais. Existem quatro formas distintas de combinar técnicas de QC e ML, que se distinguem pelo tipo de dado do problema e o tipo de hardware empregado para processar o algoritmo sendo ou quântico ou clássico (Dunjko; TAYLOR; BRIEGEL, 2016). Essas abordagens e suas descrições estão ilustradas na Tabela 2.

Tipo de Dado		Tipo de Modelo		
		C	Q	
		Processamento de Dados Clássicos utilizando modelos de ML inspirados em quântica	Processamento de Dados Clássicos utilizando modelos de QML	
	C	Processamento de dados Quânticos utilizando algoritmos de ML	Processamento de Dados Quânticos utilizando modelos de QML	
	Q			

Tabela 2 – Abordagens em QML para Processamento de Dados Clássicos e Quânticos.

Vale ressaltar que a estratégia de detecção de falhas desenvolvida nesta pesquisa enquadra-se na categoria de processamento de dados clássicos utilizando algoritmos de QML conforme mostrado na Tabela 2. Esses algoritmos são construídos com base em circuitos quânticos parametrizados - PQC. A seguir, serão discutidos os fundamentos essenciais desses algoritmos.

### 2.2.3 Circuitos quânticos variacionais

Os Circuitos Quânticos Variacionais também conhecidos como Circuitos Quânticos Parametrizados, são compostos por uma sequência de portas quânticas unitárias, algumas das quais dependem de parâmetros contínuos que podem ser ajustados durante o processo de treinamento de um modelo de ML. Esses tipos de circuitos são fundamentais para o trabalho com QML. Dentre as propriedades

fundamentais para circuitos quânticos variacionais, Sim et al. [2019] propuseram duas métricas importantes para avaliação: **expressabilidade**, que mede a capacidade do circuito de explorar o espaço de estados quânticos, e **capacidade de emaranhamento**, que avalia o grau de correlação quântica entre os *qubits* no circuito. O circuito mostrado na Figura 8 é um exemplo de um circuito que possui expressibilidade e capacidade de emaranhamento.

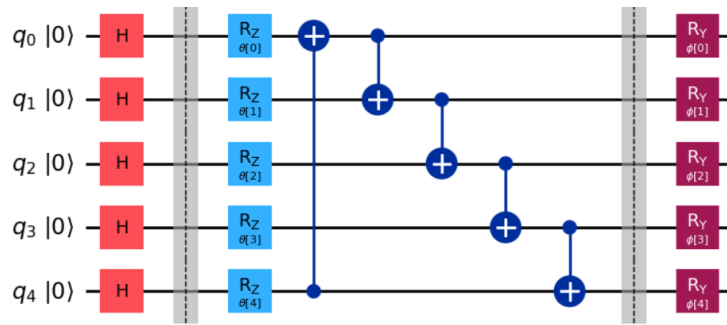


Figura 8 – Circuito com Expressibilidade e Capacidade de Emaranhamento.

O circuito descrito possui uma camada com portas Hadamard (H), rotação em Z ( $R_z$ ), porta CNOT ( $C_x$ ) e rotação em Y ( $R_y$ ). As portas Hadamard aplicam superposição aos *qubits*, rotacionando suas esferas de Bloch em torno do eixo X por  $\frac{\pi}{2}$ . As portas  $R_z$ , parametrizadas por um vetor de ângulos  $\vec{\Theta}$ , giram o estado dos *qubits* em torno do eixo Z. A configuração circular de portas CNOT entre os *qubits*  $[q_4, q_0]$ ,  $[q_0, q_1]$ ,  $[q_1, q_2]$ ,  $[q_2, q_3]$ ,  $[q_3, q_4]$  emaranha os estados dos *qubits*. Finalmente, as portas  $R_y$  aplicam rotações ao redor do eixo Y, gerando mais estados emaranhados e preenchendo a esfera de Bloch.

De acordo com Hubregtsen et al.[2020], os PQCs, utilizados como modelos para classificação em tarefas de ML, apresentam uma forte correlação entre sua capacidade de generalização e sua expressibilidade. Em contrapartida, existe uma correlação fraca entre a capacidade de generalização e a capacidade de

emaranhamento. Isso sugere que circuitos com maior expressibilidade podem ser bons candidatos para tarefas de classificação, pois são capazes de explorar uma ampla gama de estados quânticos. É também importante notar que circuitos altamente expressíveis podem enfrentar problemas como os *barren plateaus*, que dificultam a otimização devido à diminuição do gradiente da função de custo.

Outros fatores como o tamanho do conjunto de treinamento, a arquitetura do circuito e o tipo de algoritmo de otimização são cruciais para o desempenho das tarefas de classificação em QML. A escolha da codificação dos dados também desempenha um papel fundamental, pois, assim como em algoritmos de ML convencionais, essa representação numérica afeta significativamente a performance dos modelos de QML. Esse processo, denominado vetorização (do inglês, *Embedding*), pode ser feito por métodos como codificação de base, amplitude e ângulo (LaRose et al., 2020), e impacta diretamente nas métricas de classificação como Acurácia, Precisão, Recall e F1-Score (RATH; DATE, 2024).

#### 2.2.4 Modelos de QML híbridos

Os modelos híbridos de QML combinam métodos clássicos e quânticos para aproveitar as vantagens de ambos os paradigmas, visando melhorar a eficiência computacional e a capacidade de aprendizado em tarefas complexas como classificação, regressão e otimização. Essa abordagem se justifica pela habilidade dos modelos quânticos de explorar espaços de Hilbert de alta dimensionalidade, permitindo representar e separar padrões que seriam difíceis de distinguir com métodos clássicos.

Um exemplo dessa combinação é o modelo híbrido QSVM para classificação (QSVC), que une a computação clássica e quântica para aprimorar a capacidade de classificação. O SVM clássico pode enfrentar algumas dificulda-

des para encontrar um hiperplano que separe bem as classes disponíveis em um conjunto de dados e mesmo utilizando um função de kernel clássica esse mapeamento pode ter um alto custo computacional. Por outro lado, o QSVM supera essas limitações ao mapear os dados clássicos para um espaço quântico de dimensão superior por meio de uma função kernel implementada através de um PQC, o que proporciona uma separação mais eficaz entre as classes. Essa abordagem utiliza princípios da mecânica quântica para otimizar o mapeamento dos vetores de características, como demonstrado por Schuld [2021], que realizaram a analogia entre a estrutura dos kernels e as medições em algoritmos quânticos (Petrucione et al., 2018).



## 3

---

# PROBLEMA DE DETECÇÃO DE FALHAS EM MANUTENÇÃO PREDITIVA

**N**este capítulo, serão explorados os conceitos relacionados ao problema de detecção de falhas em manutenção preditiva. Com o objetivo de contextualizar sua relevância, apresentar uma revisão das estratégias já implementadas e discutir como os avanços em computação quântica podem contribuir para esse campo, particularmente no desenvolvimento de soluções mais eficientes e robustas.

### 3.1 Abordagens modernas de manutenção de equipamentos industriais

A evolução das práticas de manutenção industrial reflete uma transição significativa das abordagens tradicionais para métodos mais avançados, impulsionados por tecnologias emergentes. Entre essas abordagens, destaca-se a manutenção

preditiva, que utiliza análise de dados em tempo real para prever falhas com base nas condições operacionais atuais e históricas dos equipamentos (GILABERT et al., 2017). Essa estratégia reduz tanto o tempo de inatividade planejado quanto o não planejado, prolonga a vida útil dos ativos e melhora a eficiência operacional.

A Figura 9 mostra as etapas adotadas em um processo de manutenção preditiva, além disso, ajuda a entender onde a estratégia híbrida quântico-clássica de detecção de falhas desenvolvida nessa pesquisa pode atuar para melhorar a performance desse processo. Note na Figura que após um pré-processamento e análise dos dados coletados a partir de um sistema de IoT, conforme mostrado nos blocos amarelo, rosa e laranja é feita a detecção de falhas através de um algoritmo de aprendizagem de máquina, ilustrada no bloco verde.

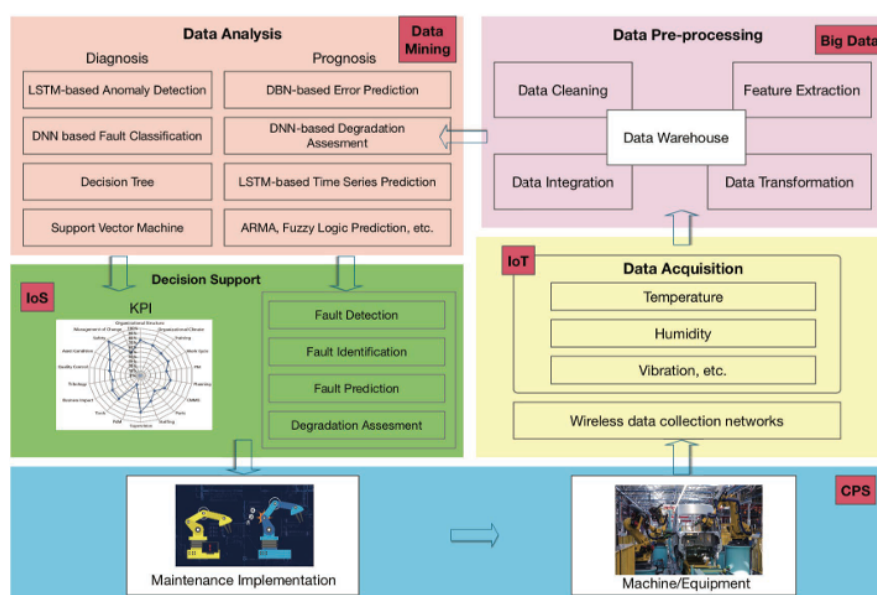


Figura 9 – As etapas de um sistema de Manutenção Preditiva como exemplo para um determinado equipamento da indústria 4.0.

A Manutenção Preditiva tem se beneficiado das tecnologias da Indústria

4.0, como Big Data, Internet das Coisas e Inteligência Artificial, que possibilitam decisões mais precisas por meio da análise avançada de dados, como mostra a Figura 9. A integração de informações coletadas por sensores embarcados em equipamentos — como vibração, temperatura e velocidade — permite a construção de modelos preditivos capazes de identificar padrões operacionais, antecipar falhas e indicar o momento ideal para intervenções. Essa abordagem torna a manutenção mais inteligente, pois substitui o modelo reativo por ações programadas com base em previsões confiáveis. No entanto, a adoção dessas tecnologias enfrenta desafios importantes, como a fusão de dados oriundos de múltiplas fontes, especialmente diante do crescente volume de informações geradas em tempo real por sistemas de monitoramento industrial (FENG; LI, 2022; RAN et al., 2019).

### 3.2 Computação quântica aplicada em estratégias de detecção de falha

A computação quântica representa uma fronteira emergente no processamento de dados complexos, com potencial significativo para aprimorar estratégias de detecção de falhas em sistemas industriais. Esta, por sua vez, visa antecipar falhas em equipamentos a partir da análise de dados provenientes de sensores e sistemas de monitoramento, permitindo intervenções planejadas e evitando paralisações não programadas. No entanto, a elevada dimensionalidade e o desbalanceamento entre as classes de Falha e Não Falha desse conjunto de dados impõem desafios computacionais substanciais.

Modelos de ML são amplamente utilizados em estratégias de detecção de falha para tarefas de classificação, identificando padrões que precedem certos tipos de falhas, como falhas mecânicas e elétricas. Contudo, quando aplicados a

grandes volumes de dados multivariados, esses modelos enfrentam limitações quanto ao tempo de processamento e à capacidade de generalização. Algoritmos quânticos, como os circuitos variacionais e os classificadores quânticos como *Quantum Neural Networks* e QSVM, têm demonstrado capacidade de lidar com essas limitações ao explorar espaços de alta dimensionalidade de forma mais eficiente. A codificação de dados clássicos em estados quânticos permite explorar correlações não triviais entre variáveis, aprimorando a sensibilidade à detecção de anomalias (Ajagekar, 2020).

## 4

---

## METODOLOGIA

A presente pesquisa é de natureza aplicada, análise qualitativa, quantitativa e abordagem descritiva, de modo a extrair informações relevantes sobre a aplicação de QML no desenvolvimento de estratégias híbridas quântico-clássicas para aplicá-las em detecção de falhas. O estudo foi conduzido por meio de uma abordagem exploratória, objetivando a extração de informações relevantes acerca da produção atual no âmbito do tema. Esse procedimento se mostra de grande relevância, uma vez que destaca a sua importância para a comunidade científica e sua utilidade no enfrentamento de desafios práticos ainda não resolvidos.

A estratégia híbrida quântico-clássica aplicada em detecção de falhas foi desenvolvida utilizando ferramentas disponíveis em bibliotecas específicas para a criação de modelos de QML, implementados na linguagem de programação *Python* utilizando as bibliotecas comuns de ML e computação quântica dentre elas se destacam *Scikit-Learn*, *Pandas* e *Qiskit*, executada no simulador quântico chamado *kuatomu*, que contém bons recursos computacionais para simular a quantidade de qubits necessária para a execução dessa estratégia (LAQCC, 2025).

Além disso, o seu desenvolvimento foi fundamentado através de uma revisão da literatura que incluiu estudos relevantes em diversas áreas. Primeiramente, foram considerados trabalhos sobre QML, com ênfase em modelos de classificação, com contribuições de autores como Schuld [2021], Wittek [2014], Jerbi et al. [2023], Mitarai et al. (2018) e Emmanoulopoulos et al. (2022). Em seguida, foram considerados trabalhos sobre ML convencional, com ênfase em modelos de regressão conforme apresentado por Medsker et al. (2001), Shaheen et al. (2023) e Dangut et al. (2023). Por fim, a literatura sobre **Deteccção de Falhas** foi analisada, com destaque para os estudos de Tschärke et al. (2023), Vilela et al. (2023), Abidi et al. (2022), Sakhnenko et al. (2021), .

Após a análise dessas literaturas, foram identificados os desafios para responder a questão de pesquisa dessa dissertação de mestrado que foi: *Como o desempenho de uma estratégia híbrida quântico-clássica, composta por um classificador baseado em Máquinas de Vetores de Suporte (SVM) com kernel calculado por um circuito quântico parametrizado, se comporta na identificação e classificação de falhas em conjuntos de dados com alta dimensionalidade e desbalanceamento?*. Assim, como objetivo geral foi proposta uma estratégia de deteção de falhas utilizando um algoritmo de máquinas de vetores suporte aprimorado por uma função de kernel quântica e a partir disso, analisar os possíveis cenários de sua utilização, com intuito de discutir as vantagens de deteção de falhas usando um modelo híbrido quântico-clássico. Com o intuito de facilitar o entendimento dos procedimentos metodológicos adotados para solucionar cada objetivo específico do Capítulo 1, seguem abaixo suas relações com os métodos adotados, respectivamente:

1. **Desenvolver modelos de SVM compostos por funções de kernel quânticas e clássicas para classificação.** Para isso, foram implementados modelos

de SVM utilizando tanto funções de kernel clássicas quanto quânticas.

2. **Desenvolver regressores para previsão de valores de cada *feature* em um determinado intervalo de tempo.** Foram criados modelos de regressão independentes para cada *feature* do conjunto de dados e foi estabelecido um intervalo de tempo que será melhor detalhado nas próximas seções.
3. **Avaliar e comparar o desempenho dos modelos quanto às métricas de ML frequentemente utilizadas e a eficiência computacional.** A análise comparativa foi conduzida por meio de métricas padronizadas, como *precision*, *recall*, *accuracy*, *F1-score*, avaliação das matrizes de confusão, curvas ROC-AUC e para a eficiência computacional foi feita uma comparação entre as métricas alcançadas e a quantidade de vetores suporte que cada configuração de modelo utilizou.
4. **Analisar as implicações práticas e os possíveis cenários de aplicação dos modelos desenvolvidos.** Foi realizada uma análise considerando as vantagens e limitações de cada abordagem, com foco em cenários industriais reais onde essa estratégia poderia ser aplicada de forma eficaz.

Os procedimentos metodológicos adotados nesse estudo foram divididos em duas etapas principais. A primeira etapa concentrou-se na implementação de um modelo quântico de QML para classificação, *Quantum Support Vector Machine*, com o objetivo de identificar a ocorrência ou não de falhas, a segunda etapa concentrou-se em um desenvolvimento mais simples de modelos de regressão para realizar um experimento de manutenção preditiva, como ilustrado na Figura 10. Vale ressaltar que a etapa de classificação realizada pelo modelo quântico

demandou um tempo de análise mais aprofundado, por ter sido o aspecto central dessa pesquisa.

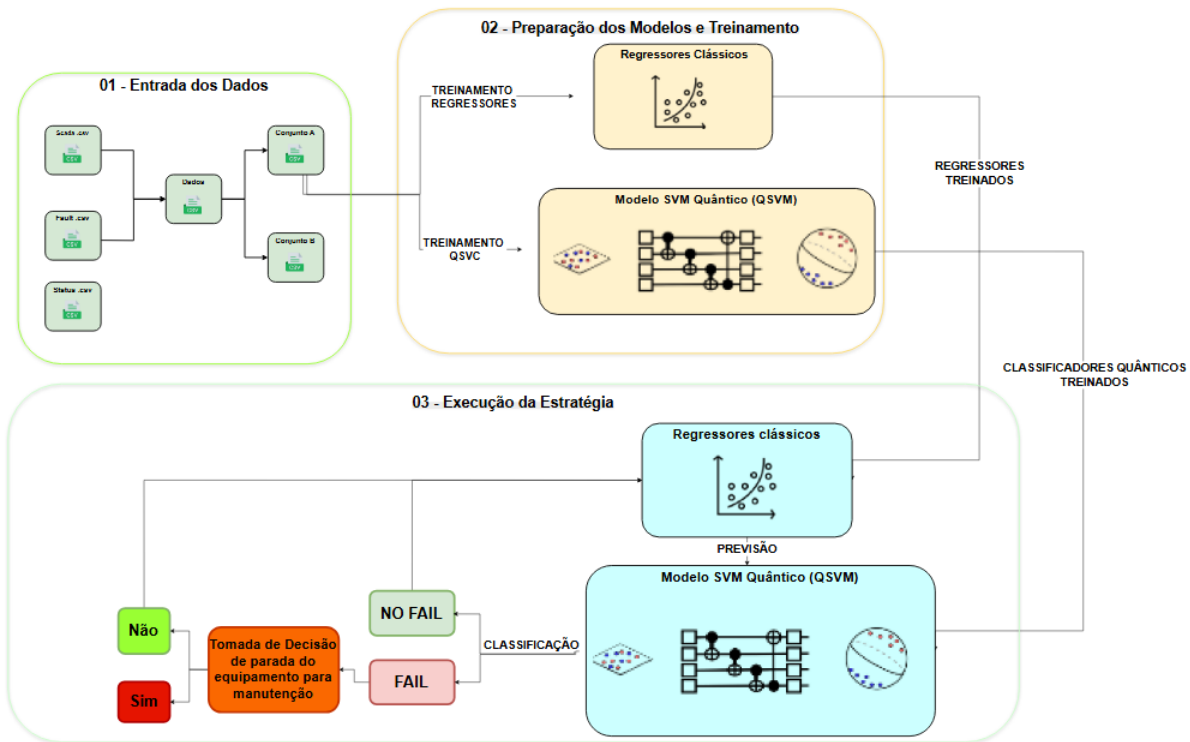


Figura 10 – Framework com a estratégia híbrida de detecção de falhas com aprendizado quântico adotada.

As seções seguintes apresentarão maiores detalhes da metodologia para melhor compreensão. Na Seção 4.1 será apresentado o caso de estudo dessa pesquisa. Na seção 4.2 serão descritos os procedimentos de pré-processamento dos dados para executar a estratégia. E, por fim, na seção 4.3 será feita uma discussão dos métodos usados para avaliar os resultados alcançados pelos modelos desenvolvidos.



## 4.1 Caso de Estudo

A estratégia híbrida de detecção de falhas implementada foi aplicada nos dados de um sistema de um aerogerador, cuja operação ininterrupta e eficiente é crucial para a geração de energia limpa. Esses equipamentos operam em ambientes submetidos a ventos intensos, variações bruscas de temperatura, elevados níveis de umidade, presença de agentes corrosivos e turbulências atmosféricas, o que contribui para o desgaste progressivo dos componentes, aumentando significativamente o risco de danos que podem comprometer a estabilidade do fornecimento energético. O sistema de detecção de falhas em aerogeradores envolve a coleta de dados sensoriais das principais partes da turbina, tais como gerador, eixo principal, sistema de transmissão e pás, cujo agrupamento em séries temporais permite inferir o comportamento do sistema ao longo do tempo e identificar padrões que antecedem possíveis falhas através de modelos computacionais (Vilela et al., 2023).

O conjunto de dados utilizados encontra-se na plataforma **Kaggle**<sup>1</sup> e foi adequado para o treinamento da estratégia desenvolvida nessa dissertação de mestrado. Destaca-se que esse conjunto de dados já foi utilizado em outros trabalhos que utilizaram modelos convencionais para executar algoritmos de ML que identificam falhas.

Esse conjunto de dados contém três arquivos com as seguintes descrições:

- O arquivo **Scada.csv** contém uma série temporal de medições realizadas entre maio de 2014 e maio de 2015, com intervalos regulares de 10 minutos.

Esses dados foram coletados por um sistema SCADA (do inglês, Supervi-

---

<sup>1</sup> <<https://www.kaggle.com/code/yohanesnuwara/iiot-wind-turbine-analytics/notebook#2.-Time-series-analysis>>

sory Control and Data Acquisition) e compreendem diversas *features* que caracterizam o funcionamento operacional do aerogerador

- O arquivo **Fault.csv** registra os períodos em que foram identificadas falhas no aerogerador, categorizadas conforme os tipos apresentados na Tabela 3;
- O arquivo **Status.csv** fornece informações sobre o status operacional da turbina ao longo do tempo, permitindo relacionar os eventos de falha aos diferentes modos de operação.

Tipo de Falha	Quantidade	Porcentagem	Causa
Falha no Sistema de Resfriamento a Ar (AF)	254 casos	45,90%	Problemas relacionados ao sistema de refrigeração do aerogerador.
Falha de Alimentação (FF)	254 casos	45,90%	Problemas na fonte de energia do aerogerador, como falhas na conexão elétrica ou nos circuitos internos.
Falha de Excitação (EF)	174 casos	31,50%	Incapacidade do gerador de produzir eletricidade de maneira eficiente, resultando em perda de geração ou operação instável.
Falha de Geração (GF)	43 casos	7,80%	Problemas na conversão de energia mecânica em elétrica, como defeitos no gerador.
Outras Falhas (MF)	82 casos	14,80%	Englobam falhas mecânicas e outras falhas não especificadas.

Tabela 3 – Distribuição de tipos de falhas do aerogerador e suas causas.

## 4.2 Procedimentos para execução da estratégia

A estratégia de detecção de falhas caracteriza pela Entrada dos Dados, Preparação dos Modelos e Execução da Estratégia composta pelo modelo quântico e clássico, conforme o framework exposto na Figura 10. Cada uma dessas etapas serão detalhadas nas próximas subseções.

### 4.2.1 Entrada dos Dados

Conforme mencionado anteriormente, o arquivo *Scada.csv* está organizado como uma série temporal e o arquivo *Fault.csv* contém os registros dos instantes em que ocorreram falhas específicas. Esses dois arquivos foram integrados e a série temporal foi mantida de tal forma que se associou os registros de falha do *Fault.csv* às respectivas instâncias do arquivo *Scada.csv*. As instâncias de tempo que não apresentaram ocorrência de falhas foram rotuladas como "Não Falha". Após essa integração, foi realizada a etapa de limpeza dos dados, com a remoção de valores ausentes (NaN), resultando em um conjunto final com dimensão de 49134 instâncias e 54 *features*.

Primeiramente, testou-se a capacidade do simulador quântico *kuatomu* para treinar o classificador desenvolvido com todas as *features*. Entretanto, devido às restrições do simulador (que não permite simular sistemas com mais de 16 qubits), optou-se pela redução de dimensionalidade dos dados por meio da Análise de Componentes Principais *PCA* (do inglês, *Principal Component Analysis*). Para isso, gerou-se uma curva de variância cumulativa do conjunto de dados, representada no gráfico da Figura 11. Observa-se que o eixo horizontal indica o número de componentes principais, e o eixo vertical, a variância acumulada. À medida que o número de componentes aumenta, a variância cresce de forma que, a partir de 4 componentes, atinge-se acima de 78% de variância cumulativa. Assim, reduziu-se a dimensionalidade para 16, 8 e 4 componentes principais, preservando aproximadamente 93%, 91% e 78% da variância dos dados, respectivamente.

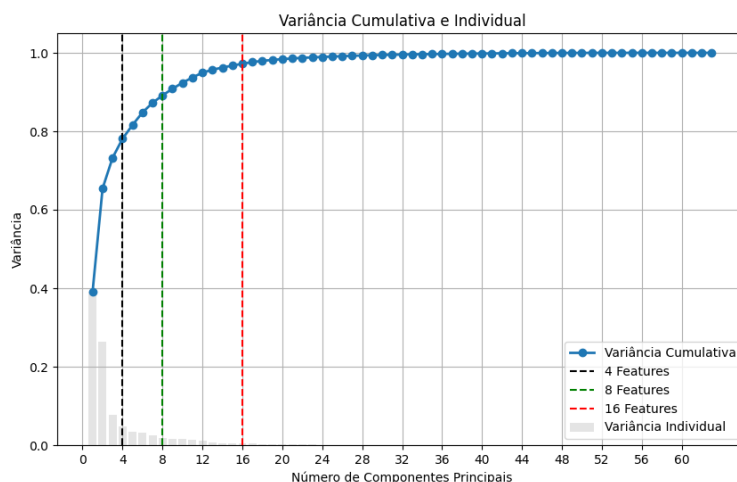


Figura 11 – Variância Cumulativa dos dados

As instâncias desses últimos conjuntos de dados com a dimensionalidade reduzida foram divididas em dois subconjuntos: o Conjunto A, com 29.001 instâncias, destinado ao treinamento e teste dos algoritmos que comporão a estratégia híbrida, e o Conjunto B, que foi reservado para o teste.

#### 4.2.2 Tratamento do desbalanceamento do conjunto de dados

Para tratar o desbalanceamento dos dados no conjunto A, aplicou-se *undersampling*, reduzindo a quantidade de amostras da classe majoritária para minimizar viés e melhorar a identificação das classes de falha. Além disso, foi atribuída uma **distribuição de pesos** com base na frequência de cada categoria, garantindo maior importância às classes minoritárias e equilibrando a contribuição de todas as classes no treinamento. Por último, antes do treinamento, aplicou-se a

padronização dos dados com *StandardScaler*, transformando-os para uma distribuição de **média zero** e **desvio-padrão unitário** garantindo que todas as *features* do aerogerador estejam na mesma escala para um treinamento mais eficiente e consistente.

#### 4.2.3 Preparação dos Modelos para compor a estratégia híbrida

Os classificadores foram inicialmente treinados com o conjunto de dados A. A principal diferença entre a arquitetura do SVC clássico e o SVC quântico está na função de kernel utilizada. Nos modelos clássicos, foram empregadas funções de kernel convencionais: **Base Radial**, **Linear**, **Polinomial** e **Sigmóide**, disponíveis na biblioteca *Scikit-learn*. E, no QSVC, foram utilizadas funções de kernel quânticas, que foram calculadas através dos PQC implementados. Para isso, foram testados três tipos de PQC: **EfficientSU2**, **Real Amplitude** e **ZZFeatureMap**. Além disso, cada PQC foi avaliado com quatro diferentes estratégias de emaranhamento: Linear, Completa, Circular e Deslocada Circular Alternada. O tipo de emaranhamento introduz diferentes tipos de correlações entre os estados dos qubits, ampliando o espaço de soluções e tornando o algoritmo mais eficiente e robusto em comparação com simulações clássicas.

As Figuras 12 e 13 ilustram a estrutura de funcionamento dos kernels quânticos utilizados.

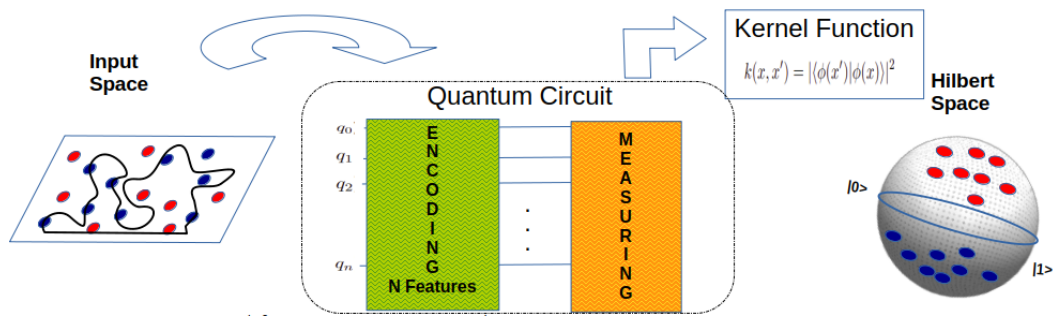


Figura 12 – Modelo Híbrido de Aprendizagem Quântica

Os codificadores das  $N$  *features* do conjunto de dados são os PQC: EfficientSU2, Real Amplitudes e ZZFeatureMap ilustrados na Figura 13, que também apresenta as estratégias de emaranhamento. Esses circuitos realizam o mapeamento dos dados para um espaço de Hilbert de alta dimensionalidade (dimensão  $2^n$  para  $n$  qubits) (Petrucione et al., 2018). O classificador é híbrido, com otimização clássica dos parâmetros e o PQC atuando como *Feature Map*. A abordagem explora propriedades quânticas para representar as *features* em um espaço exponencialmente mais rico ( $2^n$  dimensões) com apenas  $n$  qubits, o que pode facilitar a identificação de superfícies de decisão com melhor capacidade de generalização em problemas complexos.

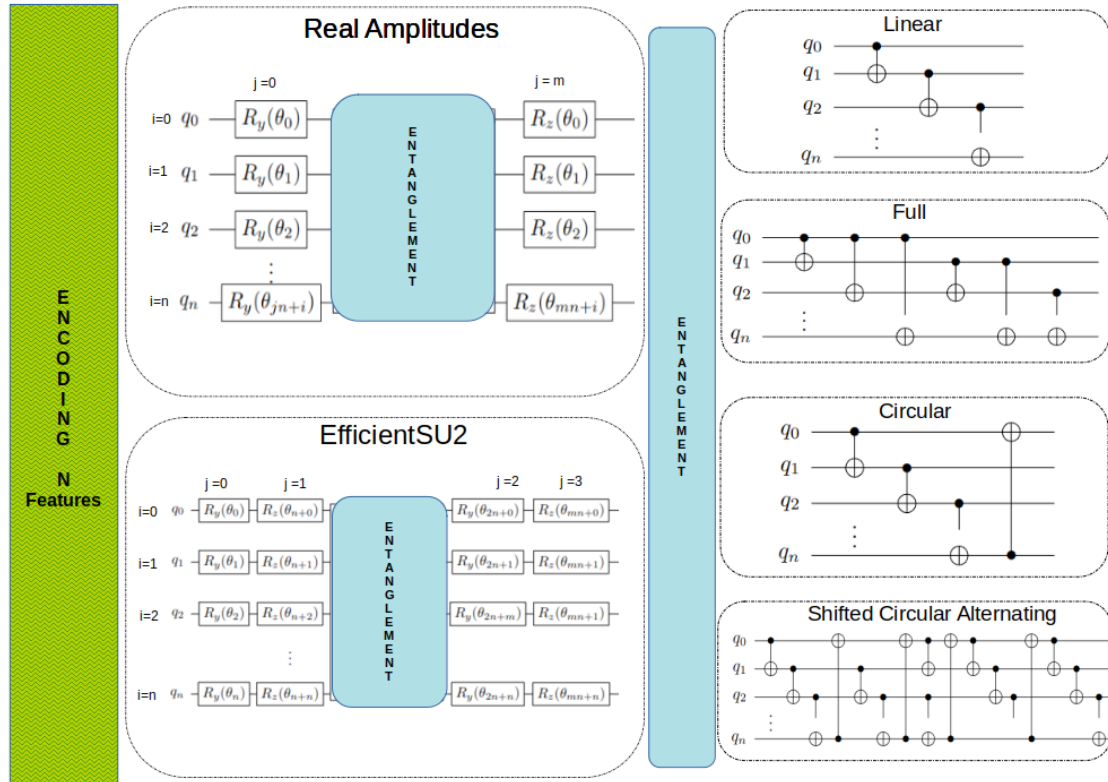


Figura 13 – Arquiteturas de kernels quânticos e emaranhamentos investigadas no trabalho.

A quantidade de qubits na codificação varia conforme o circuito escolhido: o **Real Amplitudes** utiliza  $\frac{N}{2}$  qubits para codificar  $N$  variáveis, o **EfficientSU2** emprega  $\frac{N}{4}$  qubits, enquanto o **ZZFeatureMap** emprega  $N$  qubits para  $N$  variáveis. Assim, os dois primeiros circuitos reduzem a quantidade de qubits no espaço de busca, o que pode facilitar a otimização ao exigir menos qubits para representar o problema. Além disso, o emaranhamento nos circuitos quânticos introduz correlações entre os estados dos qubits que carregam a informação das instâncias do conjunto de dados do aerogerador, ampliando o espaço de soluções e tornando o algoritmo mais eficiente e robusto em comparação com simulações

clássicas. Por fim, o treinamento dos modelos QSVC segue o mesmo princípio do SVC clássico, pois o kernel quântico apenas mapeia os dados para um espaço de maior complexidade e o processo de otimização continua sendo clássico.

A previsão das *features* foi realizada com os regressores treinados e testados, usando a biblioteca scikit-learn nos conjuntos de dados com 16, 8 e 4 componentes principais, dentro de um intervalo de tempo de 20 períodos que é o equivalente à 3 horas e 20 minutos, uma vez que cada instância do conjunto de dados possui um intervalo de tempo de 10 minutos.

#### 4.2.4 Execução da estratégia híbrida

Após realizado o treinamento e teste dos modelos quânticos separadamente, aqueles que alcançaram as melhores métricas foram integrados para formar a estratégia híbrida de detecção de falhas. Os regressores treinados e testados foram alimentados com os dados do Conjunto B e suas previsões foram utilizadas como entradas para os classificadores de tal modo que se identificasse futuras falhas no tempo de 3h20min.

### 4.3 Procedimento de análise

O estudo da qualidade dos resultados foi realizada utilizando métricas convencionais de Aprendizado de Máquina. Para os modelos de classificação, foram consideradas as seguintes métricas: Acurácia, que mede a proporção de previsões corretas; Recall, que avalia a capacidade do modelo de identificar corretamente todas as instâncias de falha, minimizando os falsos negativos; Precisão, que indica a proporção de falhas corretamente identificadas entre todas as instâncias classificadas como falhas, sendo essencial para reduzir a ocorrência de falsos positivos; e F1-Score, que calcula a média harmônica entre precisão e recall.

Além dessas métricas, foi feita uma análise das matrizes de confusão para



verificar o desempenho detalhado dos modelos em cada classe. Para compreender a complexidade dos classificadores, também foi avaliada a quantidade de vetores de suporte utilizada por cada configuração. É importante destacar que um modelo mais complexo, com maior número de vetores de suporte, não é necessariamente inferior. Essa complexidade reflete a dificuldade do modelo em encontrar uma superfície de generalização adequada. Assim, modelos com mais vetores de suporte podem, inclusive, apresentar melhor desempenho que aqueles com menos vetores e métricas inferiores.

Adicionalmente, foi realizada uma análise aprofundada por meio da Curva ROC, que compara a taxa de verdadeiros positivos com a de falsos positivos, e da AUC, que quantifica a capacidade do modelo de distinguir entre as classes de falha e não falha. A AUC foi comparada entre os conjuntos de treino e teste para garantir a consistência dos resultados.

Para os modelos de regressão, foram utilizadas as métricas MAE (Erro Absoluto Médio), MSE (Erro Quadrático Médio), RMSE (Raiz do Erro Quadrático Médio) e R (Coeficiente de Determinação), que mede a proporção da variância explicada pelo modelo.

## 5

---

# RESULTADOS E DISCUSSÕES

Neste capítulo, os resultados alcançados serão discutidos. É importante ressaltar que o principal objetivo desta dissertação de mestrado é avaliar o desempenho da estratégia híbrida quântico-clássica em detectar falhas utilizando um conjunto de dados desbalanceado e com alta dimensionalidade.

Na Seção 5.1, são apresentados os resultados obtidos da performance individual dos modelos híbridos quântico-clássico na detecção de falhas e na Seção 5.2 serão apresentados os resultados de um experimento que foi feito com os modelos híbridos quântico-clássico que obtiveram melhor performance para detecção de falhas aplicados em valores previstos por regressores.

## 5.1 Análise da Performance dos Modelos Clássico e Quântico

Nesta seção é apresentada a análise comparativa realizada entre os modelos SVM compostos por kernels quânticos e clássicos aplicados à detecção de falhas de um aerogerador. A comparação visa identificar as vantagens e limitações de cada abordagem, contribuindo para a escolha de estratégias mais eficazes de detecção de falhas.

### 5.1.1 Análise dos Modelos de Classificação

Os modelos de classificação SVM com kernels quânticos e clássicos foram treinados individualmente utilizando a amostra de dados do **Conjunto A**, conforme descrito nas Seções 4.1 e 4.2.1. Nesse contexto, foi avaliada a performance de **12 modelos SVM com kernels quânticos e 4 modelos SVM com kernels clássicos**, com o objetivo de realizar uma **análise comparativa do desempenho dos modelos de SVM com esses diferentes tipos de kernels**. Essa avaliação foi realizada por meio das **métricas padrão de aprendizagem de máquina: Acurácia, Recall, Precisão e F1-Score**. Além disso, foram utilizadas as **CMs e as Curvas RoC-AuC** para investigar se houve *overfitting* e a capacidade da identificação de falhas específicas, considerando o conjunto dados resultante após as técnicas aplicadas para reduzir o alto grau de desbalanceamento do conjunto de treinamento. É importante destacar que o problema em questão é **multiclasse** e, portanto, as curvas ROC-AUC foram calculadas utilizando a estratégia **One-Vs-Rest (OvR)**. Nessa abordagem, uma curva ROC é gerada para cada classe

tratando-a como positiva enquanto as demais são combinadas como classe negativa. Além disso, a **área sob a curva (AUC)** foi calculada usando a **média macro (Macro-Averaged AUC)**, que corresponde à média aritmética das AUCs individuais de cada classe e a quantidade de vetores suporte foi registrada para verificar a complexidade do modelo.

A Tabela 4 apresenta as métricas alcançadas pelos modelos de SVM para diferentes tipos de kernels quânticos e clássicos. Os dados demonstraram variações significativas no desempenho que dependeu do: kernel utilizado, do tipo de emaranhamento aplicado e do número de componentes principais do algoritmo de SVM.

Tabela 4 – Resultados das métricas para diferentes tipos de kernel treinados e testados com a base de teste do conjunto A.

Kernel	Emaranhamento	Comp. Principais	Nº de Vetores	Suporte	Acurácia	Precisão	Recall	F1-Score
ZZFeatureMap	Linear	8	778		62.80%	59.22%	62.80%	56.63%
		4	320		68.00%	68.54%	68.00%	67.58%
	Full	8	880		59.20%	52.77%	59.20%	50.71%
		4	328		36.40%	41.57%	36.40%	38.34%
	Circular	8	803		61.60%	57.80%	61.60%	54.85%
		4	315		41.60%	44.98%	41.60%	41.37%
	SCA	8	803		61.60%	57.80%	61.60%	54.85%
		4	315		41.60%	44.98%	41.60%	41.37%
Real Amplitudes	Linear	16	290		75.60%	78.66%	75.60%	75.63%
		8	224		63.60%	64.94%	63.60%	62.89%
		4	297		22.80%	39.86%	22.80%	27.18%
	Full	16	307		80.40%	80.67%	80.40%	79.96%
		8	213		76.80%	77.41%	76.80%	76.09%
		4	297		22.80%	39.86%	22.80%	27.18%
	Circular	16	292		75.60%	78.10%	75.60%	75.25%
		8	220		62.80%	67.53%	62.80%	63.54%
		4	297		22.80%	39.86%	22.80%	27.18%
	SCA	16	292		75.60%	78.10%	75.60%	75.25%
		8	220		62.80%	67.53%	62.80%	63.54%
		4	297		22.80%	39.86%	22.80%	27.18%
EfficientSU2	Linear	16	229		66.40%	68.11%	66.40%	66.62%
		8	321		32.00%	40.87%	32.00%	35.28%
		4	285		24.00%	36.67%	24.00%	26.77%
	Full	16	236		66.00%	71.20%	66.00%	67.61%
		8	321		32.00%	40.87%	32.00%	35.28%
		4	285		24.00%	36.67%	24.00%	26.77%
	Circular	16	235		65.60%	68.82%	65.60%	66.72%
		8	321		32.00%	40.87%	32.00%	35.28%
		4	285		24.00%	36.67%	24.00%	26.77%
	SCA	16	235		65.60%	68.82%	65.60%	66.72%
		8	321		32.00%	40.87%	32.00%	35.28%
		4	285		24.00%	36.67%	24.00%	26.77%
Classical Kernel: Linear	-	16	198		66.80%	72.40%	68.80%	68.34%
		8	256		61.60%	68.29%	61.60%	60.44%
		4	235		65.20%	65.65%	65.20%	62.15%
Classical Kernel: Polinomial	-	16	184		75.20%	77.53%	75.20%	74.20%
		8	190		74.00%	75.92%	74.00%	73.21%
		4	232		63.20%	66.58%	63.20%	63.25%
Classical Kernel: Rbf	-	16	164		76.00%	78.95%	76.00%	75.85%
		8	166		68.8%	70.52%	68.80%	68.94%
		4	201		42.00%	44.80%	42.00%	42.35%
Classical Kernel: Sigmoid	-	16	152		43.20%	56.45%	43.20%	48.23%
		8	155		37.60%	54.35%	37.60%	43.92%
		4	159		36.40%	53.74%	36.40%	42.68%

## Modelos Clássicos e Quânticos de Classificação com 16 PCs

Os resultados dos modelos com 16 componentes principais, apresentados na Tabela 4, indicam que o modelo SVM com *kernel Real Amplitudes* e emaranhamento *Full* obteve o melhor desempenho entre as abordagens testadas. Esse modelo alcançou uma acurácia de 80.40%, uma precisão de 80.67%, um *recall* de 80.40% e um F1-Score de 79.96%. Esses valores refletem um bom equilíbrio entre as métricas, sugerindo que o modelo possui uma capacidade eficaz de classificação e generalização.

Por outro lado, o modelo SVM com *kernel EfficientSU2* e emaranhamento *Full* apresentou o pior desempenho para o conjunto de 16 *features*, com uma acurácia de 66.00%, uma precisão de 71.20%, um *recall* de 66.00% e um F1-Score de 67.61%. Esses resultados indicam que essa configuração foi menos eficiente em comparação ao modelo baseado no *kernel Real Amplitudes* com emaranhamento *Full*.

Dentre os modelos clássicos, o SVM com *kernel* Polinomial alcançou uma acurácia de 75.20%, com precisão de 77.53%, *recall* de 75.20% e F1-Score de 74.20%. Esses resultados indicam um desempenho sólido e equilibrado, embora ligeiramente inferior ao modelo SVM com *kernel Real Amplitudes*. Já o modelo com *kernel* RBF obteve uma acurácia de 76.00%, precisão de 78.95%, *recall* de 76.00% e um F1-Score de 75.85%, destacando-se como um dos modelos clássicos mais eficazes. Em contraste, o *kernel* Sigmoid apresentou o pior desempenho, com uma acurácia de apenas 43.20% e métricas de precisão, *recall* e F1-Score significativamente inferiores.

Além das métricas tradicionais de Aprendizagem de Máquina, as análises das matrizes de confusão dos modelos são essenciais, especialmente em cenários onde o conjunto de dados apresenta um alto grau de desbalanceamento entre as

classes. A avaliação dessas matrizes permite uma compreensão mais detalhada dos erros de classificação, auxiliando na identificação de possíveis ajustes para melhorar o desempenho do modelo.

As matrizes de confusão referente aos modelos de SVM com kernels quânticos com 16 Componentes Principais estão mostradas na Figura 14.

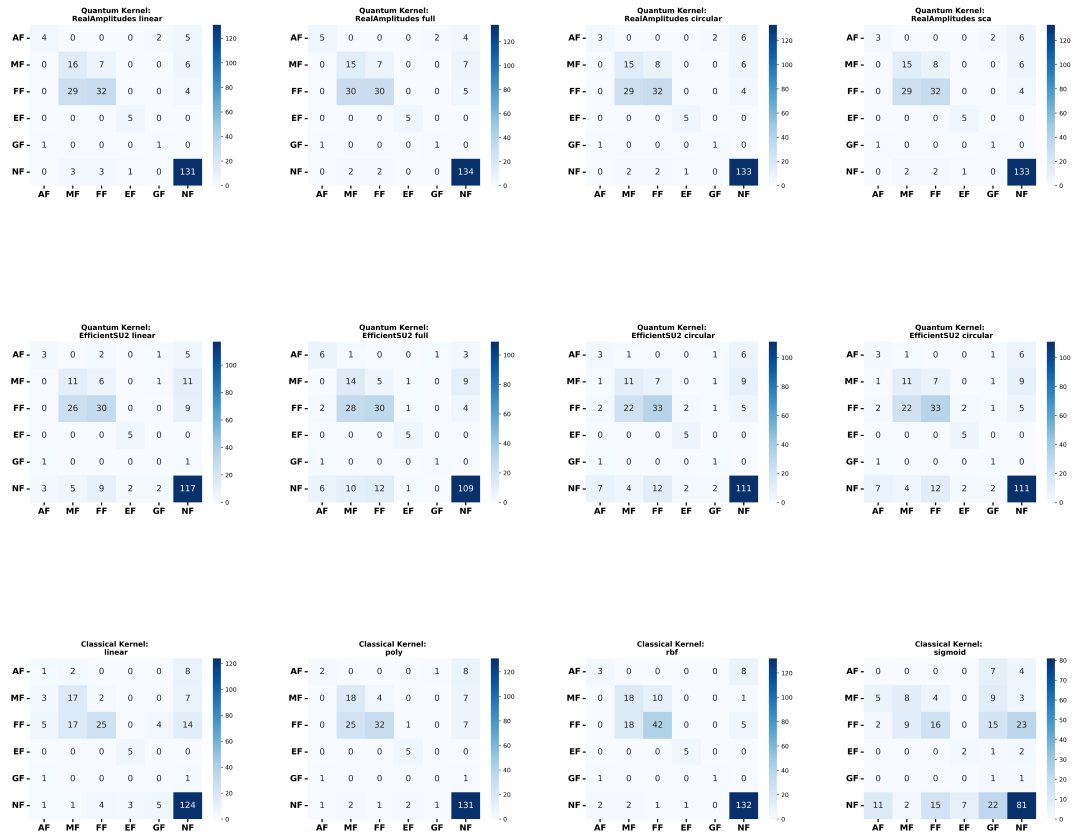


Figura 14 – Matrizes de Confusão SVM com 16 Componentes Principais

Observando as matrizes de confusão, dentre os modelos do tipo Real Amplitudes o que alcançou maiores métricas foi com emaranhamento quântico Full. Ele conseguiu identificar corretamente 5 tipos de Falha AF, 15 tipos de Falha MF, 30 tipos de Falha FF, 5 tipos de Falha EF, 1 tipo de Falha GF e 134 tipos de

Não Falha. Apesar de ter alcançado as melhores métricas, conforme apresentado na Tabela 5, esse modelo cometeu erros ao classificar quatro instâncias que não correspondiam a falhas como sendo falhas. Além disso, os modelos com emaranhamento Linear, Circular e SCA demonstraram um desempenho ligeiramente superior na identificação correta de casos de falha do tipo FF.

Embora o modelo SVM com kernel que utiliza emaranhamento Full tenha apresentado os melhores resultados, é fundamental destacar que esse tipo de emaranhamento é altamente custoso para a implementação em computadores quânticos reais. Isso ocorre devido à complexidade associada à implementação de portas de controle, como as portas CNOT, que exigem um alto nível de fidelidade e aumentam significativamente os recursos computacionais necessários. Portanto, aqui é necessária uma análise mais aprofundada e que está fora do escopo deste trabalho, para verificar a viabilidade de implementação em dispositivos quânticos reais.

Por outro lado, em relação ao modelo SVM com kernel quântico EfficientSU2, note que nesse caso, dentro dos casos de Falhas os modelos apresentaram resultados semelhantes entre os tipos de emaranhamento. Porém, o que conseguiu identificar mais tipos de falhas corretamente foi o modelo SVM com o kernel também tendo emaranhamento Full. O modelo conseguiu identificar corretamente 6 Falhas AF, 14 Falhas MF, 30 Falhas FF, 5 Falhas EF, 1 Falha GF. Além disso identificou corretamente 109 tipos de Não Falha, Apesar disso, ele identificou incorretamente 29 tipos de casos de Não Falha como Falha. Além disso, vale à pena destacar que apesar desse modelo ter identificado corretamente mais casos de Falha. Os modelos com emaranhamento Circular e SCA conseguiram identificar corretamente mais tipos de Falha FF e uma quantidade



maior de casos de Não Falha.

Por fim, se tratando de execução em dispositivos quânticos reais, seria interessante, em trabalhos futuros, testar um algoritmo quântico como o EfficientSU2, esses circuitos utilizaram somente 4 qubits e alcançaram resultados interessantes para 16 componentes principais. Uma análise de viabilidade pode ser estudada para implementação em dispositivos da era NISQ.

Em contrapartida, dentre os tipos de kernel clássicos o que conseguiu identificar um maior número de Falhas foi RBF. Esse modelo conseguiu identificar corretamente 3 tipos de Falha AF, 18 Falhas MF, 42 Falhas FF, 5 Falhas EF, 1 Falha GF. Além disso, identificou corretamente 132 tipos de Falhas 132, mas identificou incorretamente 6 casos de Não Falha como Falha.

As curvas RoC-AuC referente aos modelos de SVM com kernel quântico utilizando 16 componentes principais são mostradas na Figura 15.

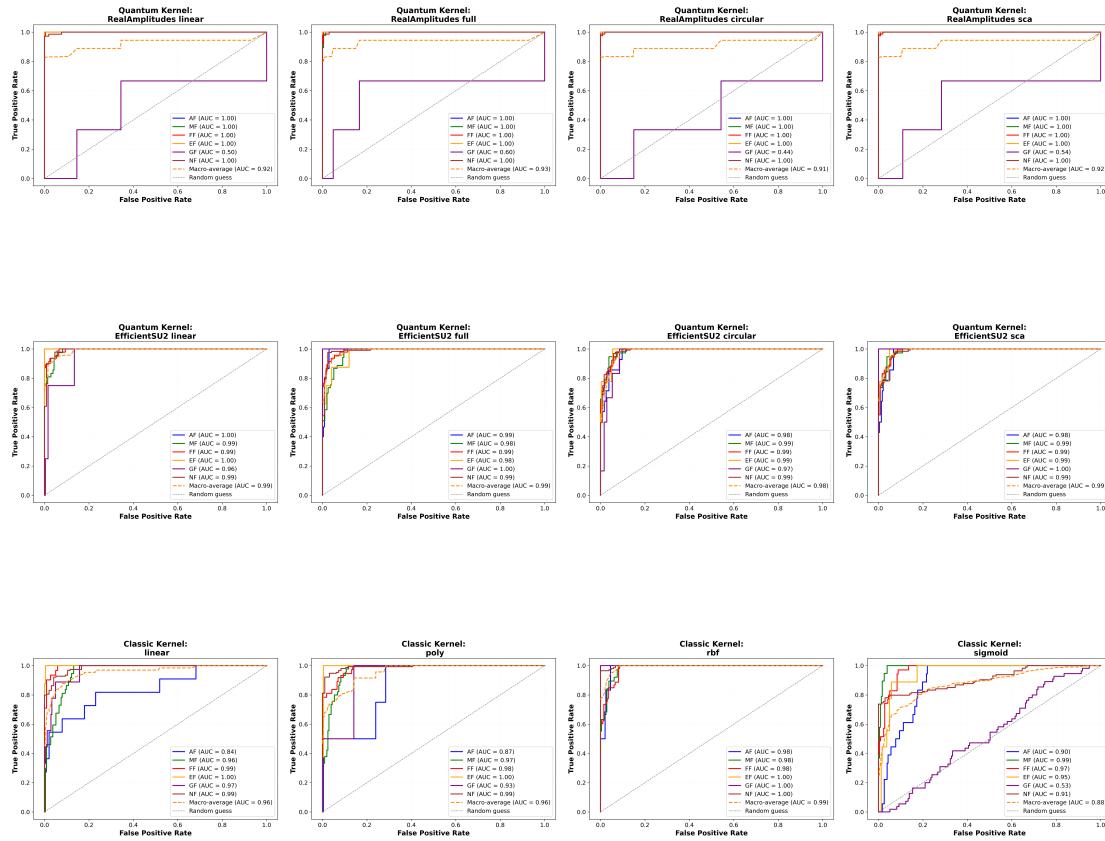


Figura 15 – Curvas RoC-AuC com 16 Componentes Principais

No caso do Real Amplitudes com emaranhamento Full, em geral apresentou um bom desempenho. Somente na generalização da classe GF que o AUC ficou aproximadamente 0.50. Já o modelo que apresentou o melhor resultado, em geral, para todas as classes obteve um excelente desempenho.

Por outro lado, embora as curvas ROC-AUC do modelo SVM com kernel EfficientSU2 tenham apresentado um ótimo desempenho durante o treinamento, as métricas obtidas ao testar o modelo com novos tipos de instâncias foram relativamente baixas. Isso indica que o modelo não conseguiu generalizar bem para novos dados. Esse *trade-off* é indicativo de overfitting, o que pode estar

associado ao conjunto de dados ou a baixa complexidade do modelo, visto que o EfficientSU2 é um circuito que reduz a dimensão do espaço de features a quarta parte.

Dentre os kernels clássicos, de modo geral, demonstraram um bom desempenho durante o treinamento e, com base nas métricas obtidas, indicaram uma capacidade satisfatória de generalização para o conjunto com 16 features. No entanto, o kernel sigmoid, embora tenha apresentado valores elevados de AUC para determinadas classes no treinamento, não obteve um desempenho satisfatório na distinção entre os diferentes tipos de falhas presentes nas instâncias do conjunto de testes.

Por fim, a análise da complexidade dos modelos na tarefa de generalização é ilustrada na Figura 16, que apresenta a quantidade de vetores de suporte utilizados na abordagem One vs. Rest para cada classe de falha. A média desses valores, consolidada na Tabela 3, fornece uma estimativa do esforço computacional associado à separabilidade de cada classe. Valores mais elevados indicam maior esforço do modelo para identificar fronteiras decisórias, possivelmente devido à maior sobreposição entre as classes ou variabilidade interna. Essa análise permite avaliar a complexidade intrínseca do problema e contribui para decisões sobre escalabilidade e otimização do sistema.

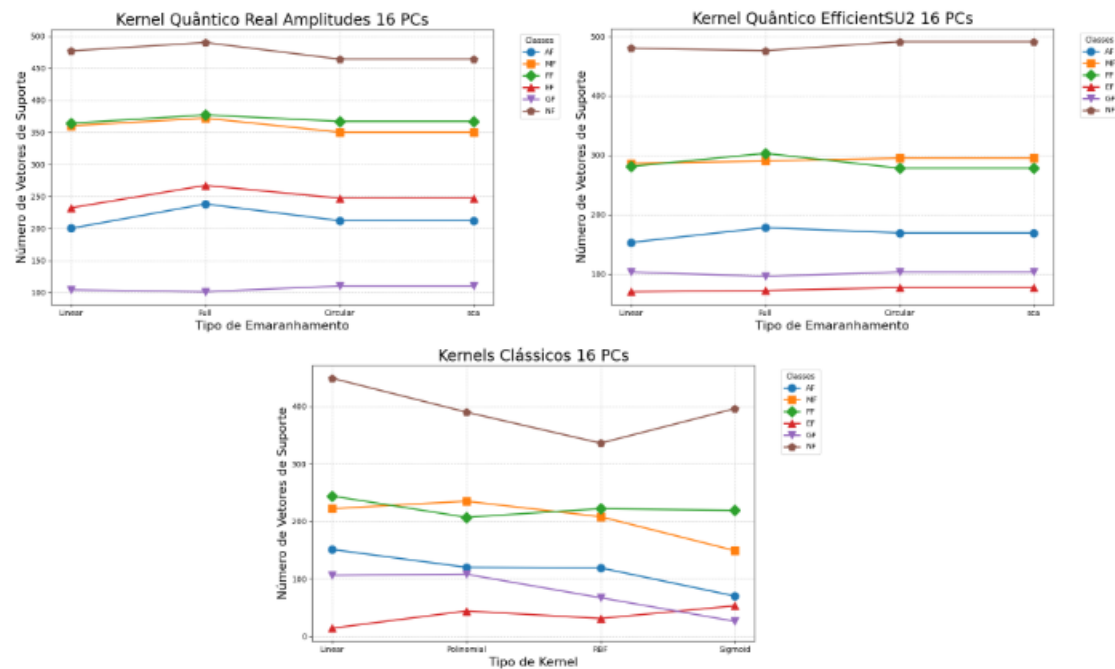


Figura 16 – Número de vetores de suporte utilizados para separar cada classe das demais na estratégia *One vs. Rest* para 16 componentes principais

O modelo SVC com kernel Real Amplitudes e emaranhamento Full, utilizando 16 componentes principais, apresentou o maior desempenho entre todos os modelos testados, com uma média de aproximadamente 307 vetores de suporte. A Figura 16 revela que a separação da classe NF (Não Falha) das demais exigiu a maior quantidade de vetores, o que pode ser atribuído ao fato de essa ser a classe majoritária, mesmo após a aplicação de técnicas de undersampling. Esse comportamento sugere que, apesar da redução do desequilíbrio, a dominância da classe NF ainda impõe maior esforço ao modelo para estabelecer uma superfície de decisão adequada, refletindo uma possível dificuldade em distinguir instâncias marginais dessa classe em relação às classes minoritárias.

O segundo melhor desempenho foi observado no modelo clássico com

kernel RBF, também com 16 componentes principais, que utilizou em média 164 vetores de suporte — praticamente metade do número utilizado pelo modelo quântico. Ainda assim, esse modelo também apresentou maior número de vetores de suporte na separação da classe NF, reforçando a hipótese de que a estrutura do conjunto de dados, especialmente o desequilíbrio residual, impõe uma complexidade adicional à modelagem dessa classe.

Essa diferença no número de vetores de suporte entre os dois modelos evidencia uma troca entre complexidade e desempenho: embora o modelo baseado em Real Amplitudes tenha exigido maior capacidade de generalização (indicada pelo número mais alto de vetores), isso resultou em um desempenho superior. Isso mostra que, a maior complexidade não deve ser interpretada negativamente, mas sim como uma resposta necessária às características do problema.

### **Modelos clássicos e quânticos de classificação com 8 componentes principais**

Os resultados dos modelos com 8 features apresentados na Tabela 6 mostram que o modelo SVM com kernel Real Amplitudes e emaranhamento Full se destacou como o melhor modelo entre os SVMs treinados com 8 features. Esse modelo alcançou uma acurácia de 76.80%, uma precisão de 77.41%, recall de 76.80% e F1-Score de 76.09%. Por outro lado, o modelo SVM com kernel EfficientSU2 e emaranhamento Full apresentou um desempenho inferior, evidenciado por métricas mais baixas. Esse resultado pode ser atribuído à estrutura do modelo, que reduz a dimensionalidade do espaço de features para um quarto do original. Com apenas dois qubits para representar as oito componentes principais, há uma perda substancial de capacidade representacional, comprometendo a separação eficiente entre as classes, tais métricas foram: Acurácia de 32.00%,

precisão de 40.87%, recall de 32.00% e F1-Score de 35.28%.

Além disso, observa-se que as diferentes estratégias de emaranhamento não impactaram significativamente os resultados desse modelo. Isso se deve ao fato de que, com apenas dois qubits disponíveis, as variações no padrão de emaranhamento não geram correlações expressivas entre os estados quânticos, limitando seu efeito na capacidade de generalização do modelo.

O modelo SVM com kernel ZZFeatureMap e emaranhamento Linear para oito componentes principais obteve uma acurácia de 62.80%, precisão de 59.22%, recall de 62.80% e F1-Score de 56.63%. Esse modelo apresentou o melhor desempenho entre os SVMs treinados com oito componentes principais. No entanto, de forma geral, o ZZFeatureMap não alcançou métricas expressivas nesse cenário, o que é um aspecto relevante a ser destacado, dado que esse método mantém a quantidade de qubits igual ao número de componentes principais do conjunto de dados.

Dentre os modelos clássicos, o Kernel Polinomial obteve uma acurácia de 74.00%, com precisão de 75.92%, recall de 74.00% e F1-Score de 73.21%, oferecendo um bom desempenho, semelhante ao Real Amplitudes com emaranhamento Full. O Kernel Rbf, por sua vez, obteve uma acurácia de 68.80%, com precisão de 70.52%, recall de 68.80% e F1-Score de 68.94%, destacando-se como um dos melhores modelos clássicos e mostrando um bom equilíbrio entre precisão e recall.

Por outro lado, o Kernel Sigmoid apresentou, novamente, o pior desempenho, com uma acurácia de 37.60%, precisão de 54.35%, Recall de 37.60% e F1-Score de 43.92%. Este modelo, portanto, não foi eficaz na tarefa de classificação para os dados analisados.

Em síntese, o kernel Real Amplitudes e estratégia de emaranhamento Full com 8 features se mostrou superior, tanto entre os modelos quânticos quanto clássicos, enquanto os modelos clássicos com kernels Polinomial também apresentou um bom desempenho. Por fim, o ZZFeatureMap e o Kernel Sigmoid ficaram abaixo das expectativas, com resultados inferiores aos demais.

As matrizes de confusão referente aos modelos de SVM com kernels quânticos com 8 Componentes Principais estão mostradas na Figura 17.



Figura 17 – Matrizes de Confusão SVM 8 Componentes Principais

As matrizes de confusão mostraram que dentre os modelos SVM que utilizaram o kernel ZZFeatureMap. O emaranhamento Circular foi o que alcançou os melhores resultados, identificando corretamente 16 tipos de Falhas somente. Em geral, esses modelos foram os que apresentaram os mais baixos desempenhos para 8 Features em comparação com os outros tipos de kernel.

Por outro lado, as matrizes de confusão dos modelos com kernel Real Amplitudes, mostraram que o kernel com emaranhamento Full novamente se destacou diante dos demais identificando corretamente 2 tipos de casos de Falhas AF, 18 Falhas MF, 36 Falhas FF, 5 Falhas EF e 0 tipos de Falha GF. Além disso identificou corretamente 131 casos de Não Falha. Em geral esse modelo se destacou diante dos demais com 8 componentes principais. Outro ponto relevante é que os outros modelos conseguiram identificar pelo menos um caso de falha do tipo GF enquanto o que utilizou emaranhamento Full não conseguiu.

Em contrapartida, as matrizes de confusão dos modelos SVM com kernel EfficientSU2 não demonstraram variação de performance em relação ao tipo de emaranhamento. Como explicado nos parágrafos anteriores, esse algoritmo quântico utiliza uma quantidade de qubits que é a quarta parte da quantidade de componentes principais, como mostrado na Figura 13. Como são 8 componentes, esse algoritmo quântico só terá 2 qubits e ao variar a estratégia de emaranhamento, as correlações geradas pelo emaranhamento não apresentarão variações, pois o circuito continuará sendo o mesmo. Além disso, existiram muitos erros nas identificações das classes como mostradas nas matrizes de confusão da Figura 17 e nas métricas da Tabela 5. Isso pode ser justificado por terem muito poucos estados dos sistemas quânticos, ao ponto do algoritmo não ser capaz de encontrar uma superfície de generalização para o problema que é complexo.



Por fim, o modelo clássico que apresentou melhores desempenhos para identificar tipos de Falhas foi o que utilizou kernel Polinomial e kernel RBF. Conseguiram identificar corretamente 56 e 55 tipos de Falha respectivamente. O modelo com kernel Polinomial identificou corretamente 3 tipos de falhas AF, 17 falhas MF, 31 falhas FF, 5 falhas EF e 0 falhas GF e conseguiu identificar corretamente 129 casos de não falha, classificando incorretamente 9 tipos de casos de não falha como falha. O modelo com kernel RBF identificou corretamente 4 tipos de falhas AF, 14 falhas MF, 32 falhas FF, 5 falhas EF e conseguiu identificar corretamente 117 casos de não falha, classificando incorretamente 21 casos de não falha como falha.

As curvas RoC-AuC referente aos modelos de SVM com 8 componentes principais são mostradas na Figura 18.

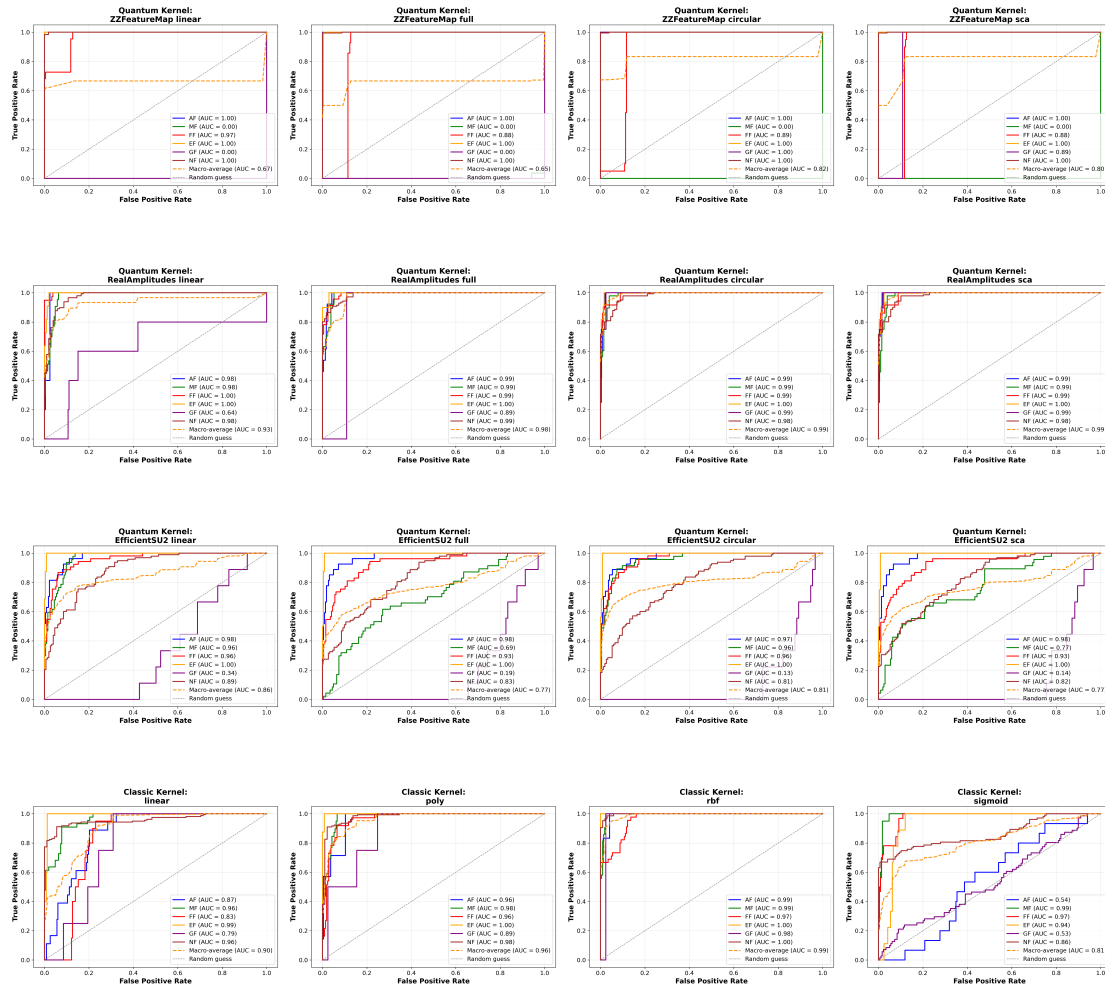


Figura 18 – Curvas RoC-AuC 8 Componentes Principais

Os modelos baseados em *ZZFeatureMap* apresentaram desempenho significativamente variável. A configuração *ZZFeatureMap + Full* atingiu um desempenho quase ideal, com valores de AUC próximos a 1 para todas as classes, enquanto as versões *Linear* e *Circular* exibiram comportamento mais instável, possivelmente devido à maneira como os dados foram projetados no espaço quântico.

Na segunda linha, os kernels baseados em Real Amplitudes mostraram maior robustez, especialmente nas configurações *Full* e *Circular*, onde a separabilidade das classes foi maximizada. No entanto, a versão *Linear* apresentou menor desempenho, sugerindo que o uso de um espaço de características mais complexo melhora a capacidade discriminativa do modelo.

Os kernels *EfficientSU2* demonstraram desempenho intermediário entre as abordagens anteriores, com variações significativas entre as métricas de similaridade. A versão *Full* obteve métricas mais altas, enquanto as versões *Linear* e *SCA* tiveram desempenho inferior em algumas classes, indicando que a profundidade do circuito pode impactar a efetividade do aprendizado.

Os modelos SVM com kernels clássicos *RBF* e *Polynomial* mostraram desempenho competitivo, enquanto o kernel Sigmoid apresentou resultados mais inconsistentes. Em alguns cenários, os kernels quânticos superaram os métodos clássicos, evidenciando o potencial das abordagens híbridas para problemas de classificação complexos.

Por fim, a análise da complexidade dos modelos para 8 componentes principais na tarefa de generalização é ilustrada na Figura 19, que apresenta a quantidade de vetores de suporte utilizados na abordagem One vs. Rest para cada classe de falha.

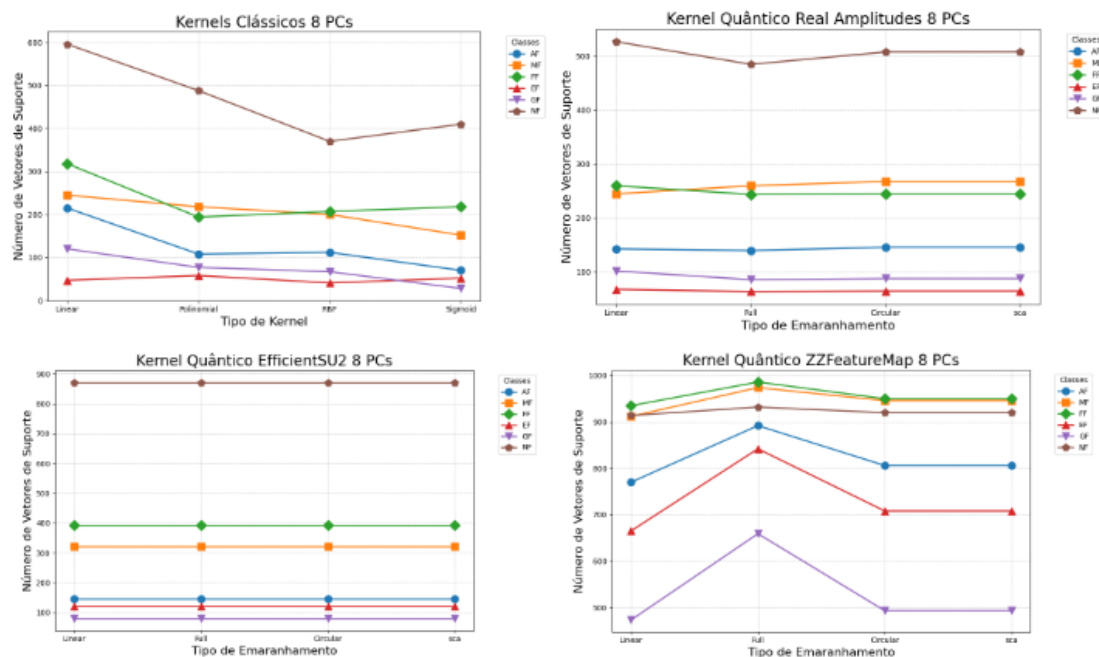


Figura 19 – Quantidade de vetores suporte utilizadas para tentar separar cada classe das outras na estratégia One Vs. Rest para 8 componentes principais.

O modelo SVC com kernel Real Amplitudes e emaranhamento Full utilizando 8 componentes principais, apresentou novamente o maior desempenho, com uma média de aproximadamente 213 vetores suporte. A Figura 19 mostra também que a separação da classe NF das demais exigiu maior quantidade de vetores suporte e mostra também que a quantidade de vetores suporte dos kernels clássicos em relação ao kernel quântico real amplitudes foi em média igual. Note também que o kernel quântico EfficientSU2 necessitou de muitos vetores suporte e não conseguiu obter uma superfície que generalizasse bem, semelhante ao ZZFeatureMap.

#### Modelos Clássicos e Quânticos de Classificação com 4 Componentes Princi-

## país

Os resultados dos modelos com 4 features apresentados na Tabela 6 mostram que o modelo SVM com kernel ZZFeatureMap com emaranhamento Linear se destacou como o melhor modelo entre os SVMs treinados com 4 componentes principais. Esse modelo alcançou uma acurácia de 68.00%, uma precisão de 68.54%, recall de 68.00% e F1-Score de 67.58%. Apesar disso, ao utilizar 4 componentes principais os modelos SVMs com kernels quânticos e clássicos, em geral, não conseguiram alcançar ótimos resultados.

Por outro lado, os modelos SVM com kernel EfficientSU2 e RealAmplitudes apresentaram um desempenho significativamente inferior, evidenciado por métricas mais baixas. Esse resultado pode ser atribuído à estrutura do modelo, que reduz a dimensionalidade do espaço de features para um quarto e a metade do original, respectivamente. Com apenas um qubit e dois qubits para representar as quatro componentes principais, há uma perda substancial de capacidade representacional, comprometendo a separação eficiente entre as classes. Como consequência, esses modelos atingiram uma acurácia em torno de 23.00%, precisão de 38.00%, recall de 23.00% e F1-Score de 30.00%. Além disso, observa-se que as diferentes estratégias de emaranhamento não impactaram significativamente os resultados desses modelos. Devido ao mesmo fato já discutido nos parágrafos anteriores

Dentre os modelos SVM com kernels clássicos, o Linear, obteve uma acurácia de 65.20%, com precisão de 65.65%, recall de 65.20% e F1-Score de 62.15%, oferecendo um desempenho um pouco menor do que o modelo SVM com o kernel quântico ZZFeatureMap Linear. Por outro lado, o Kernel Sigmoid apresentou, novamente, o pior desempenho, com uma acurácia de 36.40%,

precisão de 53.74%, Recall de 36.40% e F1-Score de 42.68%. Este modelo, portanto, não foi eficaz na tarefa de classificação para os dados analisados.

As matrizes de confusão referente aos modelos de SVM com 4 componentes principais são mostradas na Figura 20.



Figura 20 – Matrizes de Confusão SVM com 4 Componentes Principais

De acordo com as matrizes de confusão acima, o modelo com kernel ZZFeatureMap com emaranhamento Linear foi o que conseguiu identificar mais

casos de falha, totalizando 47 falhas. Dentre essas 47, o modelo identificou corretamente 1 falha do tipo AF, 11 falhas do tipo MF, 34 falhas do tipo FF, 1 falha do tipo EF e 0 falha do tipo GF. Além disso, esse modelo conseguiu identificar corretamente 123 casos de não falha e classificou incorretamente 15 casos de não falha como falha. É importante destacar que nesse caso o tipo de estratégia de emaranhamento gerou grandes divergências entre os modelos. Em contrapartida, analisando as matrizes de confusão referente dos modelos de SVM com kernel quântico Real Amplitudes, observa-se que elas apresentaram o mesmo resultado. Isso pelo mesmo motivo já discutido, como esse tipo de kernel reduz a quantidade de qubits para a metade gerou-se um sistema quântico possuindo somente dois qubits. Portanto o tipo de emaranhamento, nesse caso, não influenciou para gerar divergências entre os resultados e em geral o desempenho desse modelo não apresentou bons resultados. Conseguindo identificar corretamente apenas 1 tipo de falha AF, 9 falhas MF, 1 falha FF, 0 falhas EF e GF. Além disso, esses modelos identificaram corretamente 46 casos de não falha e classificaram incorretamente 92 casos de não falha como falha.

Por outro lado, as matrizes de confusão referente aos modelos de SVM com kernel quântico EfficientSU2 também se repetiu para esse tipo de kernel. Observe que todos os modelos alcançaram os mesmos resultados insatisfatórios, identificando corretamente somente 4 tipos de falha FF, 1 falha EF. Além disso, identificaram corretamente 55 tipos de casos de não falha e erraram ao identificar 83 casos de não falha como falha.

Por último, ao analisar as matrizes de confusão referente aos modelos de SVM com kernels clássicos, nota-se que o modelo que se destacou em identificar mais tipos de falha foi o kernel linear. Identificou corretamente 16 tipos de falha

MF, 23 falhas FF. Além disso, identificou corretamente 124 casos de não falha e errou classificando 24 casos de não falha como falha.

As curvas RoC-AuC referente aos modelos de SVM 4 componentes principais são mostradas na Figura 21.

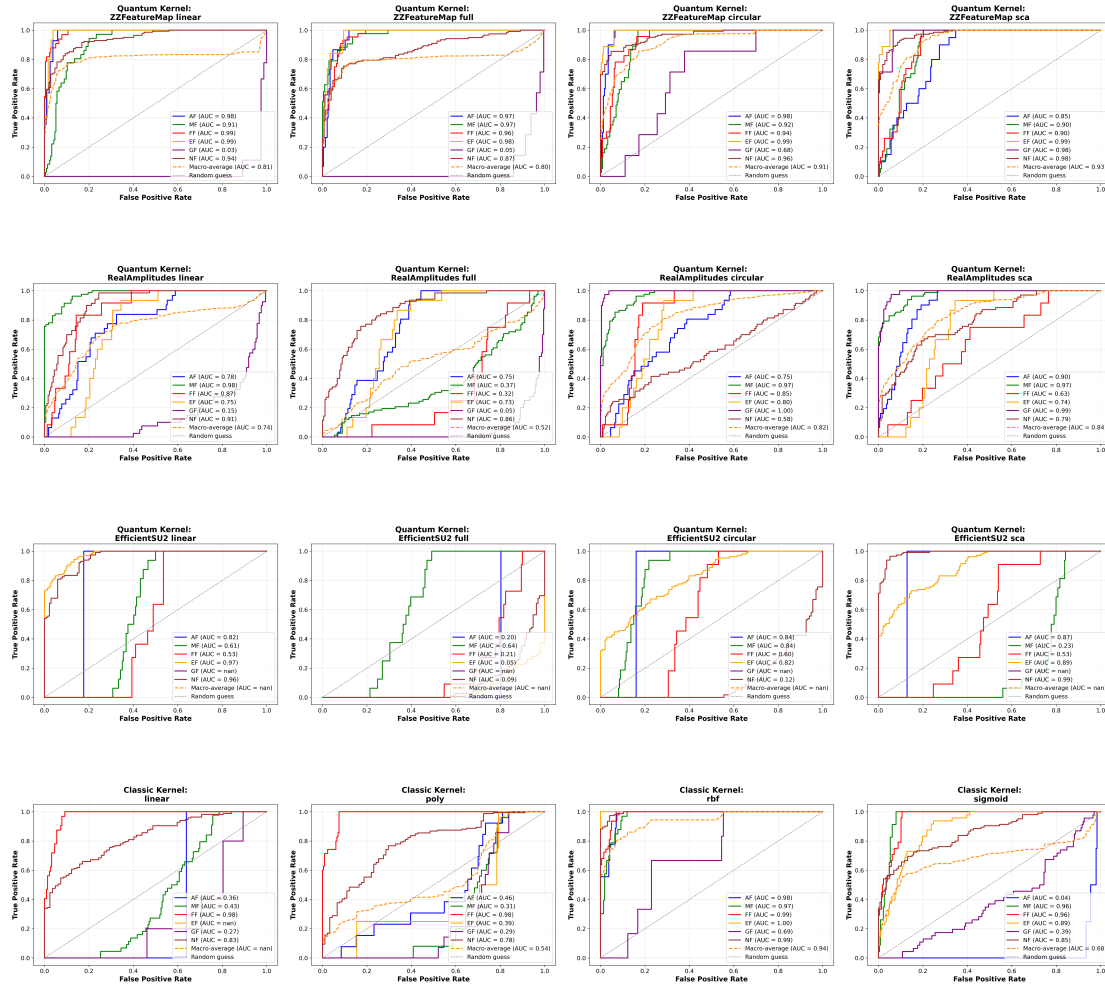


Figura 21 – Curvas RoC-AuC com 4 Componentes Principais

Dentre os kernels quânticos *ZZFeatureMap*, observa-se que a configuração *ZZFeatureMap + Full* apresentou os melhores resultados, com valores de AUC ele-



vados e uma separabilidade bem definida. Os emaranhamentos *Linear* e *Circular* demonstraram maior variabilidade no desempenho, enquanto o kernel *ZZFeatureMap* + *SCA* também apresentou uma performance competitiva, sugerindo uma representação eficaz dos dados.

Para os kernels baseados em *Real Amplitudes*, o kernel *Real Amplitudes* + *Full* obteve um desempenho consistente, indicando que essa abordagem permite uma melhor separação das classes. No entanto, as versões *Linear* e *Circular* demonstraram maior sensibilidade à escolha da métrica de similaridade, enquanto o *Real Amplitudes* + *SCA* obteve resultados intermediários.

Já os kernels baseados em *EfficientSU2* apresentaram um desempenho mais instável. A configuração *EfficientSU2* + *Full* foi a única que demonstrou uma curva ROC bem definida, enquanto as versões *Linear* e *Circular* tiveram uma performance inferior, possivelmente devido à complexidade do circuito e à escolha da métrica de similaridade. O kernel *EfficientSU2* + *SCA* apresentou a maior variabilidade, indicando dificuldades na generalização dos dados.

Em comparação com os modelos clássicos, o kernel *RBF* se destacou como a abordagem mais robusta, confirmando sua eficácia em problemas de separabilidade não-linear. O kernel *Linear* teve um desempenho aceitável, mas inferior às melhores configurações quânticas. Já o kernel *Polynomial* apresentou variações significativas, sugerindo dificuldades de ajuste, e, por último, o kernel *Sigmoid* teve um dos piores desempenhos, reforçando sua menor aplicabilidade para esse problema.

Por fim, a análise da complexidade dos modelos para 4 componentes principais na tarefa de generalização é ilustrada na Figura 22, que apresenta a quantidade de vetores de suporte utilizados na abordagem One vs. Rest para

cada classe de falha.

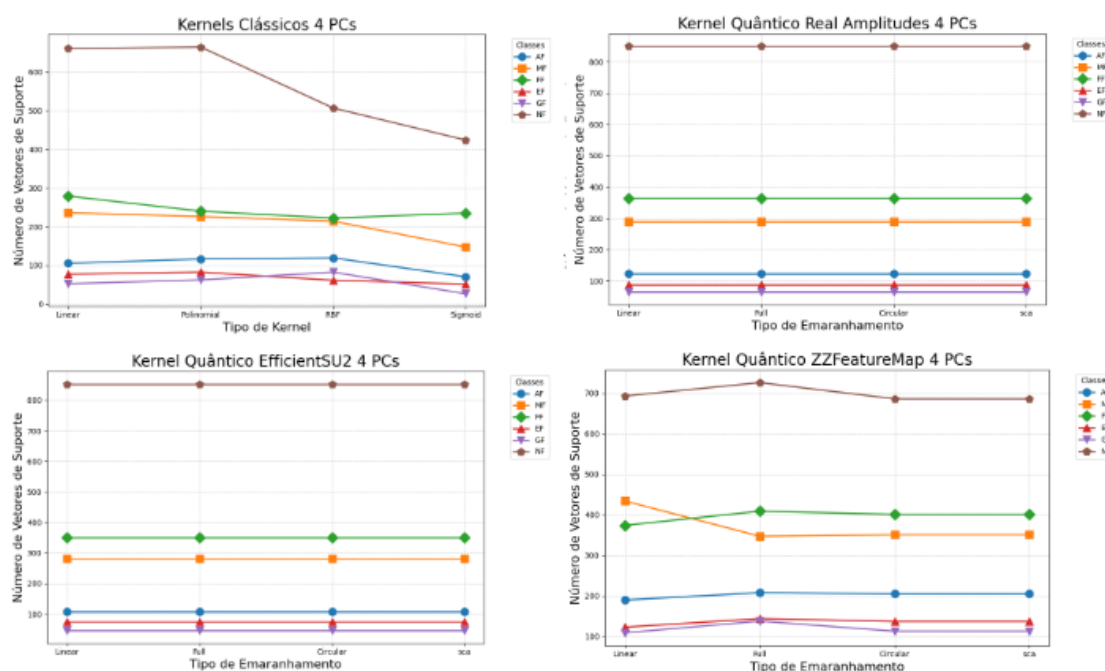


Figura 22 – Quantidade de vetores suporte utilizadas para tentar separar cada classe das outras na estratégia One Vs. Rest para 4 componentes principais.

Os modelos, em geral, não apresentaram bons resultados para 4 componentes principais, todavia o ZZFeatureMap com emaranhamento linear utilizando 4 componentes principais, apresentou relativamente o maior desempenho usando 320 vetores suporte. A Figura 22 mostra novamente que a separação da classe NF das demais exigiu maior quantidade de vetores suporte e mostra também que a quantidade de vetores suporte dos kernels clássicos se assemelhou a utilizada pelo ZZFeatureMap.

### Modelos clássicos de Regressão

Os modelos de regressão também foram treinados com base no **Conjunto**

A. A avaliação desses modelos foi realizada por meio das métricas Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) e Coeficiente de Determinação ( $R^2$ ), conforme descrito na Seção 4.3. A Tabela 5 mostra os valores de tais métricas.

Número de Features	MAE	MSE	$R^2$
4	1.1	6.42	0.37
8	0.89	3.65	0.35
16	0.65	2.23	0.22
$\sigma = 0.4921$			

Tabela 5 – Métricas de desempenho para diferentes números de *features*. O MAE é comparado com um limite aceitável.

Os resultados apresentados na Tabela 6 fornecem uma visão detalhada sobre o desempenho do modelo preditivo para diferentes números de features. Inicialmente, com 4 features, o modelo obteve um MAE de 1.1 e um MSE de 6.42, o que indica que o modelo ainda estava cometendo erros relativamente grandes. Comparado ao desvio padrão dos dados de treino (0.4921), esses valores sugerem que o modelo não está capturando de forma eficiente a variabilidade dos dados. O coeficiente de determinação ( $R^2$ ) também foi de 0.37, refletindo que o modelo foi capaz de explicar 37% da variabilidade dos dados, mas ainda há uma grande parte das variações não modeladas.

Ao adicionar 8 features ao modelo, houve uma melhoria em ambas as métricas de erro. O MAE reduziu para 0.89 e o MSE caiu para 3.65, o que indica que o aumento no número de features contribuiu para uma melhoria nas previsões, embora o erro ainda fosse superior ao desvio padrão. Apesar dessa melhoria, o  $R^2$  caiu para 0.35, indicando que, embora o modelo estivesse acertando melhor suas previsões, sua capacidade de explicar a variabilidade dos dados não aumentou significativamente. Esse comportamento sugere que

o modelo pode não estar aproveitando todo o potencial das features adicionais para melhorar a generalização dos dados.

Com 16 features, o modelo apresentou uma redução ainda maior no MAE, que chegou a 0.65, e o MSE caiu para 2.23. Esse comportamento reforça a ideia de que a adição de features permite uma maior precisão nas previsões, mas, ao mesmo tempo,  $R^2$  continuou a cair, atingindo 0.22. Esse declínio no  $R^2$  sugere que o modelo pode estar se ajustando demais aos dados de treino, um sinal claro de overfitting, onde o modelo se adapta excessivamente às particularidades dos dados, sem melhorar sua capacidade de generalização para novos conjuntos de dados.

Em comparação com o desvio padrão dos dados de treino, que foi de 0.4921, o modelo ainda apresenta um MAE significativamente maior, o que indica que, mesmo com a redução dos erros, o modelo não está alcançando uma precisão ideal. Isso sugere que o modelo ainda pode ter dificuldades em prever de forma exata os valores, dado que o erro médio está bem acima do limite aceitável, representado pelo desvio padrão. Esse fenômeno é comum em modelos que ainda não capturaram toda a complexidade dos dados, sugerindo que ajustes adicionais são necessários.

Por fim, os resultados demonstram que, enquanto o aumento do número de features melhora a precisão do modelo, a queda no  $R^2$  levanta preocupações sobre a sua capacidade de generalização. O modelo pode estar se ajustando excessivamente aos dados de treino, o que limita sua aplicação em novos conjuntos de dados.

A partir disso, dado esses dados previstos ainda no conjunto A, foram utilizados os classificadores já treinados para verificar se eles conseguem uma

boa classificação. A Tabela 7 mostra as métricas de ML alcançadas para os modelos que apresentaram maiores métricas para cada quantidade de componentes principais da Tabela 6.

Kernel	Emaranhamento	Comp. Principais	Acurácia	Precisão	Recall	F1-Score
Real Amplitudes	Full	16	74.11%	22.31%	29.10%	22.82%
Real Amplitudes	Full	8	74.11%	16.49%	20.75%	14.64%
ZZFeatureMap	Linear	4	67.76%	16.46%	19.90%	14.10%
Classical Kernel: Rbf	-	16	70.81%	16.17%	12.12%	13.86%
Classical Kernel: Polynomial	-	8	91.38%	16.21%	15.64%	15.92%
Classical Kernel: Linear	-	4	57.53%	17.01%	14.42%	13.35%

Tabela 6 – Resultados de desempenho dos diferentes kernels usando o Conjunto A.

De acordo com a Tabela 6, os experimentos indicaram que os classificadores apresentaram um desempenho insatisfatório na identificação das classes de falha. Além disso, observa-se indícios de *overfitting* na classificação dos valores previstos pelos regressores.

A baixa qualidade das previsões geradas pelos regressores pode ser atribuída às métricas MAE, MSE e  $R^2$ , que apresentam valores significativamente superiores ao desvio padrão dos dados. Esse comportamento sugere que a presença de *outliers* no conjunto de dados pode ter impactado negativamente a modelagem, tornando necessária a investigação de abordagens alternativas para a implementação dos regressores, a fim de melhorar o desempenho das previsões.

Neste estudo, optou-se por testar modelos mais simples para previsão de séries temporais. No entanto, ao integrar esses modelos aos classificadores para a identificação das classes de falha e não falha, observou-se um decréscimo substancial no desempenho dos classificadores, indicando que essa abordagem pode não ser a mais adequada para o problema em questão.

## 5.2 Resultados da abordagem híbrida clássico-quântica integrada aos regressores

Na etapa de integração da Abordagem Híbrida Clássico-Quântica, os modelos foram combinados para avaliação conjunta. Para essa fase, foi utilizado o **Conjunto B**, que havia sido previamente reservado como conjunto de teste, após o treinamento independente dos modelos, conforme descrito na seção anterior.

Nesse conjunto de teste, os regressores foram empregados para prever os valores futuros das variáveis, e, em seguida, os classificadores foram aplicados para categorizar as previsões em uma das classes possíveis: Falha ou Não Falha.

Kernel	Emaranhamento	Comp. Principais	Acurácia	Precisão	Recall	F1-Score
Real Amplitudes	Full	16	99.66%	55.23%	57.98%	56.17%
Real Amplitudes	Full	8	99.66%	48.51%	48.32%	47.80%
ZZFeatureMap	Linear	4	94.56%	34.52%	30.81%	32.17%
Classical Kernel: Rbf	-	16	97.92%	34.48%	37.53%	33.30%
Classical Kernel: Polynomial	-	8	92.26%	33.20%	69.34%	40.90%
Classical Kernel: Linear	-	4	96.82%	22.23%	36.82%	25.06%

Tabela 7 – Resultados de desempenho dos diferentes kernels.

A inclusão dos regressores resultou em uma queda no desempenho dos modelos, conforme apresentado na Tabela 7. Esse comportamento pode ser atribuído à natureza do conjunto de dados, no qual as falhas geralmente se encontram fora do desvio padrão, caracterizando-se como *outliers*. Consequentemente, os regressores tendem a se ajustar preferencialmente aos casos de não falha, dificultando a correta previsão das falhas.

Embora esses resultados não fossem teoricamente esperados, conclui-se que utilizar a abordagem implementada com esse tipo de regressores não apresentou um desempenho satisfatório na prática. Em cenários como esse, a simples implementação de um regressor dessa forma não é suficiente para

garantir uma modelagem eficaz do comportamento das variáveis desse conjunto de dados. Uma vez que os modelos de classificação foram treinados e testados com as amostras de dados do conjunto A de tal modo que eles conseguiram alcançar resultados satisfatórios.

Por fim, o capítulo 6 desta dissertação tem o intuito de discutir as considerações finais desta dissertação destacando os pontos que foram tratados e sugestões de trabalhos futuros.

## 8

---

## CONSIDERAÇÕES FINAIS

Nesta dissertação, foi desenvolvida uma estratégia híbrida de detecção de falhas de um aerogerador. A metodologia utilizou um modelo híbrido quântico-clássico para detectar falhas. Esse modelo, denominado QSVM, utilizou um circuito quântico para realizar o mapeamento das features em um espaço de maior complexidade, proporcionando, em determinados cenários, um desempenho competitivo ou superior aos dos modelos clássicos.

Dentre os modelos avaliados, destacou-se o SVM com kernel Real Amplitudes e emaranhamento Full, aplicado a 16 componentes principais, alcançando 80.40% de acurácia, 80.67% de precisão, 80.40% de recall e 79.96% de F1-score nos testes com o "conjunto A". Adicionalmente, os regressores foram treinados nesse mesmo conjunto, preservando a estrutura temporal dos dados para garantir previsões consistentes. Outros modelos também demonstraram desempenho relevante, como os SVMs com emaranhamentos Linear e Circular, ambos utilizando 8 componentes principais. O primeiro atingiu 75.60% de acurácia, 78.66% de precisão, 75.60% de recall e 75.63% de F1-score, enquanto o segundo apresen-



tou resultados similares, evidenciando a influência do tipo de emaranhamento na eficácia da classificação.

Entretanto, um dos principais desafios enfrentados foi a limitação do conjunto de dados, o que impediu a construção de um modelo preditivo capaz de fazer previsões com lags superiores a 3 horas e 40 minutos. Tentativas de treinamento para períodos mais longos comprometeriam a qualidade da divisão dos dados, afetando tanto o desempenho dos algoritmos de regressão e classificação quanto a integração da estratégia híbrida para testes com o "conjunto B". Além disso, a disponibilidade de conjuntos de dados adequados para esse tipo de estudo ainda é um obstáculo significativo. Dados de falhas de equipamentos frequentemente apresentam limitações, como baixa quantidade de eventos registrados, falta de padronização nos rótulos e dificuldades na obtenção de séries temporais completas e confiáveis. Isso reforça a necessidade de estratégias robustas para o tratamento e enriquecimento dos dados, garantindo que os modelos sejam capazes de generalizar adequadamente para diferentes cenários operacionais.

Diferentes abordagens para a estratégia híbrida também foram consideradas, como a utilização de redes neurais recorrentes para previsão e redes neurais quânticas para classificação. No entanto, este estudo concentrou-se na investigação de algoritmos quânticos, QSVM, com o intuito de avaliar suas vantagens na resolução de problemas de classificação em espaços de maior complexidade. E, em virtude disso, foram utilizados modelos de previsão baseado em regressão, pois foram mais simples de serem implementados e interpretados. Os resultados obtidos sugerem que modelos híbridos quântico-clássicos podem oferecer ganhos em certos cenários de detecção de falhas. Todavia, vale ressaltar que este

projeto foi desenvolvido utilizando simuladores quânticos e a estratégia não foi testada em computadores quânticos reais, uma vez que o objetivo foi estudar o comportamento da estratégia híbrida em cenários de manutenção preditiva simulando um dispositivo quântico que seja tolerante à falhas feito no simulador. Esses resultados contribuem para abordagens algorítmicas inovadoras que se consiga explorar o potencial da computação quântica aplicada na área de manutenção preditiva.

Este trabalho foi apresentado em 2023 no **1º Encontro Regional de Grupos de Pesquisa em Computação e Informação Quântica - EGPCIQ**, um evento regional ocorrido no Rio de Janeiro que abordou temas relacionados às tecnologias quânticas. Ele também foi aceito para apresentação no evento chamado **BRACIS 2025 (Brazilian Conference on Intelligent Systems)** (Nooblath; Bessa; Freitas, 2025), uma conferência com grande relevância nacional voltado para pesquisas em Inteligência Artificial. A aceitação nesta conferência reforça a relevância científica da proposta e o potencial das soluções quântico-clássicas no contexto da IA aplicada à manutenção preditiva.

## 8.1 Trabalhos Futuros

Com base nos resultados obtidos neste trabalho, observa-se várias possibilidades para investigações futuras na área de manutenção preditiva utilizando modelos híbridos quântico-clássicos. Um caminho promissor consiste na exploração de outras arquiteturas quânticas, como as Redes Neurais Quânticas, a fim de avaliar seu desempenho em comparação aos kernels quânticos empregados no QSVM. Adicionalmente, a investigação de diferentes estratégias de treinamento e otimi-

zação dos circuitos quânticos pode contribuir significativamente para aprimorar a estabilidade e o desempenho dos modelos, além da execução da estratégia em hardware quântico real.

Dado que este estudo teve como foco principal a etapa de classificação por meio de algoritmos quânticos, a etapa de regressão ainda apresenta oportunidades de melhoria, visando aumentar o desempenho global da estratégia. Nesse sentido, recomenda-se explorar técnicas de regularização, bem como revisar o conjunto de variáveis de entrada, com o objetivo de mitigar o overfitting e aumentar a robustez do modelo diante de novos dados. Além disso, a adoção de estratégias como a validação cruzada pode assegurar um desempenho mais consistente e confiável em diferentes partições do conjunto de dados.

A aplicação da estratégia híbrida proposta em diferentes tipos de equipamentos industriais representa outra vertente de investigação promissora, pois possibilita avaliar a adaptabilidade do método em distintos contextos operacionais. Além disso, a análise da escalabilidade da solução em hardwares quânticos reais, considerando limitações relacionadas ao ruído e à profundidade dos circuitos, configura-se como um desafio crucial para avanços futuros.

Por fim, a busca por novos algoritmos híbridos com menor complexidade computacional e maior eficiência na classificação de falhas permanece como uma linha de pesquisa aberta e altamente relevante. A identificação de casos específicos em que algoritmos quânticos superem abordagens clássicas poderá representar um avanço significativo para a área, fomentando novas investigações e contribuindo para o desenvolvimento da Manutenção Preditiva baseada em Computação Quântica

---

## REFERÊNCIAS

Abidi, M. H. et al. Predictive Maintenance Planning for Industry 4 . 0 Using Machine Learning for Sustainable Manufacturing. 2022. 16, 17, 24, 25, 32, 33

Aguado, R.; Kouwenhoven, L. P. Majorana qubits for topological quantum computing. *Physics Today*, v. 73, n. 6, p. 44–50, 06 2020. ISSN 0031-9228. Disponível em: <<https://doi.org/10.1063/PT.3.4499>>. 14, 22, 30

Ajagekar, A. Quantum computing assisted deep learning for fault detection and diagnosis in industrial process systems. *Computers Chemical Engineering*, v. 143, p. 107119, 2020. ISSN 0098-1354. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0098135420308322>>. 62

Amazon Braket. *Amazon Braket*. 2023. Disponível em: <<https://aws.amazon.com/pt/braket/>>. 47, 48

Biamonte, J. et al. Quantum machine learning. *Nature*, Springer Science and Business Media LLC, v. 549, n. 7671, p. 195–202, set. 2017. ISSN 1476-4687. Disponível em: <<http://dx.doi.org/10.1038/nature23474>>. 15, 23, 31

Chollet, F. *Deep learning with Python*. [S.l.]: Simon and Schuster, 2021. 52, 53, 54

Chowdhury et al. A review of aircraft environmental control system simulation and diagnostics. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, v. 0, n. 0, p. 09544100231154441, 2023. 16, 24, 32

Dunjko, V.; TAYLOR, J. M.; BRIEGEL, H. J. Quantum-enhanced machine learning. *Physical Review Letters*, 2016. 55

Dwave. *What is Quantum Annealing?* 2024. Disponível em: <[https://docs.dwavesys.com/docs/latest/c\\_gs\\_2.html](https://docs.dwavesys.com/docs/latest/c_gs_2.html)>. 15, 23, 31

FENG, M.; LI, Y. Predictive maintenance decision making based on reinforcement learning in multistage production systems. *IEEE Access*, v. 10, p. 18910–18921, 2022. 61

Feynman, R. P. Simulating physics with computers. *International Journal of Theoretical Physics*, v. 21, n. 6-7, p. 467–488, 1982. Disponível em: <<https://doi.org/10.1007/BF02650179>>. 46

GILABERT, E. et al. Simulation of predictive maintenance strategies for cost-effectiveness analysis. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, v. 231, p. 2242 – 2250, 2017. 60

Goodfellow, I. et al. *Deep learning*. [S.l.]: MIT press, 2016. 50, 51

Google Quantum. *Google Quantum AI*. 2023. Disponível em: <<https://quantumai.google/>>. 47

GoogleQ. *Google Willow: A New Quantum Chip for the Next Generation of Quantum Computing*. 2024. Accessed: 2024-12-17. Disponível em: <<https://blog.google/technology/research/google-willow-quantum-chip/>>. 14, 22, 30, 48

Griffiths, D. J. et al. *Mecânica Quântica*. .: PEARSON, 2011. 40

Grover, L. K. A fast quantum mechanical algorithm for database search. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. [S.l.: s.n.], 1996. p. 212–219. 13, 21, 29

Gupta, V. et al. Predictive maintenance of baggage handling conveyors using iot. *Computers & Industrial Engineering*, Elsevier, v. 177, p. 109033, 2023. 16, 24, 32

Horvat, S. et al. Universal quantum computation via quantum controlled classical operations. *Journal of Physics A: Mathematical and Theoretical*, IOP Publishing, v. 55, n. 7, p. 075301, jan 2022. Disponível em: <<https://dx.doi.org/10.1088/1751-8121/ac4393>>. 15, 23, 31

Hubregtsen, T. et al. Evaluation of parameterized quantum circuits: on the relation between classification accuracy, expressibility and entangling capability. 2020. 56

IBMQ. *IBM Quantum Computing*. 2023. Disponível em: <<https://www.ibm.com/quantum>>. 42, 47, 48, 49

IBMQ. *Qiskit Textbook*. 2023. Disponível em: <<https://qiskit.org/learn/>>. 44

- IonQ. *Quantum is Now*. 2025. Disponível em: <<https://ionq.com/>>. 47
- IQM. *We build quantum computers for research and supercomputing centers*. 2024. Disponível em: <<https://www.meetiqm.com/technology>>. 48
- LaRose, R. et al. Robust data encodings for quantum classifiers. *Physical Review A*, American Physical Society (APS), v. 102, n. 3, sep 2020. 57
- Mermin, N. D. *Quantum Computer Science: An Introduction*. [S.l.: s.n.], 2007. 39, 40
- Microsoft. Interferometric single-shot parity measurement in inas-al hybrid devices. *Nature*, v. 638, p. 651–655, 2025. Disponível em: <<https://doi.org/10.1038/s41586-024-08445-2>>. 14, 22, 30
- Naguleswaran, S. *Quantum Machine Learning: Quantum Kernel Methods*. 2024. Disponível em: <<https://arxiv.org/abs/2405.01780>>. 15, 23, 31
- Nielsen, M. A. et al. *Quantum computation and quantum information*. [S.l.]: Cambridge University Press, 2010. 676 p. ISBN 9781107002173. 13, 21, 29, 38, 39, 42, 48
- Nooblath, M.; Bessa, J. A.; Freitas, R. d. Mitigating high dimensionality and few training data with a quantum kernel learning approach. In: *Intelligent Systems – 35th Brazilian Conference, BRACIS 2025*. Cham: Springer, 2025. (Lecture Notes in Computer Science, v. 16180). Proceedings, Part II. To be published. 90
- PAN, Z. et al. Understanding the impact of quantum noise on quantum programs. *2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, p. 426–437, 2023. 48
- Park, J. et al. A methodology for comparing and benchmarking quantum devices. In: CHO, D.-J.; KIM, J. (Ed.). *Unconventional Computation and Natural Computation*. Cham: Springer Nature Switzerland, 2024. p. 28–42. ISBN 978-3-031-63742-1. 14, 22, 30, 50
- Pasqal. *We are building (the best) Quantum computers out of neutral atoms*. 2025. Disponível em: <<https://www.pasqal.com/>>. 47
- Petruccione et al. *Supervised learning with quantum computers*. [S.l.]: Springer, 2018. v. 17. 50, 58
- Preskill, J. Quantum computing in the nisq era and beyond. *Quantum*, Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, v. 2, p. 79, ago. 2018. ISSN 2521-327X. Disponível em: <<http://dx.doi.org/10.22331/q-2018-08-06-79>>. 15, 23, 31

QuERA. *Our Quantum Roadmap*. 2024. Disponível em: <<https://www.quera.com/our-quantum-roadmap>>. 47

RAN, Y. et al. A survey of predictive maintenance: Systems, purposes and approaches. *ArXiv*, abs/1912.07383, 2019. 61

Raspberry Pi Foundation. *Raspberry Pi 3 Model B*. 2019. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>>. 63

RATH, M.; DATE, H. Quantum data encoding: a comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy. *EPJ Quantum Technology*, v. 11, p. 72, 2024. Disponível em: <<https://doi.org/10.1140/epjqt/s40507-024-00285-3>>. 57

Salehi, M. Ai-enhanced renewable energy: Revolutionizing monitoring & driving sustainable progress. *Available at SSRN 4416913*, 2023. 17, 25, 33

Santos, E. M. d. et al. Teoria e aplicação de support vector machines à aprendizagem e reconhecimento de objetos baseado na aparência. Universidade Federal de Campina Grande, 2002. 52

Schuld, M. *Supervised quantum machine learning models are kernel methods*. 2021. Disponível em: <<https://arxiv.org/abs/2101.11020>>. 16, 17, 24, 25, 32, 33, 58

Shor, P. Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 1996. 13, 21, 29

Sim, S. et al. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, Wiley, 2019. 56

Xanadu. *Xanadu Photonics*. 2023. Disponível em: <<https://www.xanadu.ai/photonics>>. 47

Zeguendry, A. et al. Quantum machine learning: A review and case studies. 2023. 16, 17, 24, 25, 32, 33