



Universidade Estadual Paulista "Júlio de Mesquita Filho"
Instituto de Biociências, Letras e Ciências Exatas - Campus de São José do Rio
Preto

LUIZ ENRIQUE BURSI

Análise Comparativa entre Algoritmos Clássicos e Quânticos para a
Identificação de Áudios Sintéticos Gerados por Inteligência Artificial

São José do Rio Preto

2025

Luiz Enrique Bursi

Análise Comparativa entre Algoritmos Clássicos e Quânticos para a Identificação de Áudios Sintéticos Gerados por Inteligência Artificial

Monografia apresentada à Universidade Estadual Paulista (UNESP), Instituto de Biociências, Letras e Ciências Exatas, São José do Rio Preto, para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Eng. Rodrigo Capobianco Guido

São José do Rio Preto
2025

B972a Bursi, Luiz Enrique
Análise comparativa entre algoritmos clássicos e quânticos para a
identificação de áudios sintéticos gerados por inteligência artificial /
Luiz Enrique Bursi. -- São José do Rio Preto, 2025
44 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Ciência da
Computação) - Universidade Estadual Paulista (UNESP), Instituto de
Biotecnologia, Letras e Ciências Exatas, São José do Rio Preto
Orientador: Rodrigo Capobianco Guido

1. Processamento de sinais Técnicas digitais. 2. Computação
Quântica. 3. Machine Learning. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

LUIZ ENRIQUE BURSI

**ANÁLISE COMPARATIVA ENTRE ALGORITMOS CLÁSSICOS E QUÂNTICOS
PARA A IDENTIFICAÇÃO DE ÁUDIOS SINTÉTICOS GERADOS POR
INTELIGÊNCIA ARTIFICIAL**

Trabalho de Conclusão de Curso apresentado à Universidade Estadual Paulista (UNESP), Instituto de Biociências, Letras e Ciências Exatas, São José do Rio Preto, para obtenção do título de Bacharel em Ciência da Computação.

Data de defesa: 10/12/2025

BANCA EXAMINADORA

Prof. Dr. Eng. Rodrigo Capobianco Guido
UNESP – Instituto de Biociências, Letras e Ciências Exatas

Profa. Dra. Rogéria Cristiane Gratão de Souza
UNESP – Instituto de Biociências, Letras e Ciências Exatas

Prof. Dr. Arnaldo Candido Junior
UNESP – Instituto de Biociências, Letras e Ciências Exatas

Dedico este trabalho a minha família.

AGRADECIMENTOS

Gostaria de começar agradecendo aos meus pais, Luiz Fernando Bursi e Viviane Roda, por todo amor, suporte e incentivos dado a mim, sem eles nada disso seria possível. Agradeço também à minha irmã Elena Bursi, que diariamente me instiga a ser melhor a fim de servir como bom exemplo.

Quero também agradecer a minha namorada Brenda, ao meu melhor amigo e mais novo irmão Yhan e todos meus amigos, que durante todo esse tempo, principalmente nesse último ano de graduação, me ajudaram e apoiaram momentos tão importante e decisivos, além de experiências, conquistas e ótimas recordações que compartilhamos.

Ao meu Orientador Rodrigo Capobianco Guido, pela dedicação e paciência que foi imprescindível para o desenvolvimento deste trabalho. Além de todo o corpo docente e banca examinadora, por toda experiência e ensinamentos trocados ao longo dos anos.

*“O gênio vê a resposta antes da pergunta.”
(Robert Oppenheimer)*

RESUMO

Nos últimos anos, o escopo de atuação nas sub-áreas da Inteligência Artificial aumentou significativamente. Atualmente, observamos técnicas cada vez mais avançadas e realistas para a geração de conteúdos sintéticos, inclusive áudios e sinais de voz. Tais conteúdos, chamados de AIGC (*Artificial Intelligence Generated Content*), são disponibilizados em muitas ferramentas que contam com modelos *text-to-speech* e com sistemas de clonagem de voz, o que torna cada vez mais difícil distinguir gravações autênticas de produções artificiais e gera preocupação ética e moral quanto ao seu uso. Paralelamente, métodos de *Machine Learning* têm alcançado níveis crescentes de refinamento em suas técnicas clássicas. Nesse sentido, a utilização dos mesmos métodos a partir de uma abordagem quântica nos dados instiga pesquisas e investimentos, conforme a Computação Quântica também se desenvolve. Diante desse panorama, é executada uma análise comparativa entre métodos clássicos de aprendizado de máquina e abordagens quânticas aplicadas à detecção de áudios sintéticos gerados por IA, observando as vantagens e desvantagens de ambos, além dos desafios da simulação quântica em máquinas de arquitetura clássica.

Palavras-Chave: Processamento Digital de Sinais (PDS); Computação Quântica; Aprendizado de Máquina; Classificação de Conteúdo Gerado por IA ; K-Vizinhos Mais Próximos (KNN);

ABSTRACT

In recent years, the scope of activity within the subfields of Artificial Intelligence has increased significantly. Today, we encounter increasingly advanced and realistic techniques for generating synthetic content, including audio and speech signals. Such content, known as AIGC (Artificial Intelligence Generated Content), is available in many tools that incorporate text-to-speech models and voice-cloning systems, making it progressively more difficult to distinguish authentic recordings from artificial productions and raising ethical and moral concerns regarding their use. In parallel, Machine Learning methods have reached a growing level of refinement in their classical techniques. Therefore, applying these same methods through a quantum approach to data has encouraged research and investment, as Quantum Computing continues to evolve. Given this panorama, a comparative analysis is carried out between classical machine learning methods and quantum approaches applied to the detection of AI-generated synthetic audio, examining the advantages and disadvantages of both, as well as the challenges of quantum simulation on classical-architecture machines.

Keywords: Signal Digital Processing (SDP); Quantum Computing; Machine Learning; Classification of AI-Generated Content; K-Nearest Neighbors (KNN);

LISTA DE ILUSTRAÇÕES

Figura 1	Ilustração ZCR	5
Figura 2	Demonstração do Classificador KNN	9
Figura 3	Diagrama do processo metodológico	13
Figura 4	Forma de onda do áudio da classe <i>auth.</i>	14
Figura 5	Forma de onda do áudio da classe <i>fake.</i>	15
Figura 6	Comparação entre os espectrogramas Mel das duas classes analisadas. . .	16
Figura 7	Histograma de acurácia do KNN Clássico.	22
Figura 8	Histograma de acurácia do KNN Quântico.	23
Figura 9	Histograma de acurácia dos métodos KNN Clássico e QKNN.	24
Figura 10	<i>Boxplot</i> comparativo das acurácias dos métodos KNN Clássico e QKNN. .	25
Figura 11	<i>Scatter</i> pareado da acurácia em cada execução dos métodos KNN Clássico e QKNN.	26

LISTA DE TABELAS

Tabela 1 – Hiperparâmetros utilizados no <i>GridSearchCV</i> para o modelo KNN clássico.	17
Tabela 2 – Médias e desvios padrão das métricas para o KNN Clássico	22
Tabela 3 – Valores mínimo e máximo das métricas do KNN Clássico	22
Tabela 4 – Médias e desvios padrão das métricas para o KNN Quântico	23
Tabela 5 – Valores mínimo e máximo das métricas do KNN Quântico	23
Tabela 6 – Comparação das métricas médias entre os classificadores	24

LISTA DE ABREVIATURAS E SIGLAS

IA	Inteligência Artificial
AIGC	<i>Artificial Inteligente Generated Content</i>
QML	<i>Quantum Machine Learning</i>
KNN	<i>K-Nearest Neighbor</i>
QKNN	<i>Quantum K-Nearest Neighbor</i>
ZCR	<i>Zero-Crossing Rate</i>
MFCC	<i>Mel-Frequency Cepstral Coefficients</i>
NISQ	<i>Noisy Intermediate-Scale Quantum</i>
MEMS	<i>Micro-Electromechanical Systems</i>

LISTA DE CÓDIGOS

3.1	Escolha de Hiperparâmetros com <i>GridSearchCV</i>	17
3.2	Aplicação do PCA na Base de dados	18
3.3	Inicialização do dispositivo quântico	19
3.4	Implementação do Kernel Quântico	20
3.5	Implementação da função de distância Hilbert-Schmidt	20

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Justificativa	1
1.2	Objetivos	2
1.2.1	Objetivos Específicos	2
1.3	Organização do trabalho	3
2	REFERENCIAL TEÓRICO	4
2.1	Energia	4
2.2	Taxa de Cruzamento por Zero	5
2.3	Entropia	6
2.4	<i>Mel-Frequency Cepstrum Coefficients</i>	6
2.5	<i>K-Nearest Neighbor</i>	8
2.6	<i>Quantum Kernel</i>	10
2.7	Codificação dos Dados	11
3	METODOLOGIA	13
3.1	Base de dados	13
3.2	Extração de <i>Features</i>	15
3.3	<i>K-Nearest Neighbor</i> Clássico	17
3.3.1	Hiperparâmetros e Pré-processamento	17
3.3.2	Treinamento do modelo	18
3.4	<i>K-Nearest Neighbor</i> Quântico	19
3.4.1	Pré-processamento dos dados	19
3.4.2	Construção do <i>Kernel</i> Quântico	19
4	RESULTADOS	21
4.1	Apresentação do Resultados do KNN Clássico	22
4.2	Apresentação do Resultados do KNN Quântico	23
4.3	Comparação entre os Métodos	24
4.3.1	Visualizações	24
5	CONCLUSÃO	27
5.1	Trabalhos Futuros	27
	Referências	29

1 INTRODUÇÃO

É indiscutível o avanço que a Inteligência Artificial teve nos últimos anos. Até recentemente, falar sobre IA era falar apenas sobre *Machine Learning* ou, em alguns casos, *Deep Learning*. Atualmente, temos um cenário totalmente novo, trazendo novos conceitos e expandindo essa área da computação. Termos como Redes Neurais, IA Generativa e *Large Language Models*, antes pouco discutidos e estudados, estão hoje no dia-a-dia de inúmeras pessoas. Naturalmente, métodos mais clássicos e pioneiros atingem um nível de sofisticação que instiga outras abordagens nas etapas centrais e cruciais do desenvolvimento de modelos.

1.1 JUSTIFICATIVA

A busca por maior eficiência computacional vem de encontro com a necessidade do desenvolvimento de novas capacidades de modelagem. Por isso, na crescente área da Computação Quântica, explorar as propriedades intrinsecamente quânticas surge como uma solução promissora para potencialmente capturar padrões e correlações de maneira mais eficiente do que o paradigma clássico [12].

Outro exemplo, dentre os vários avanços citados, está a capacidade de gerar conteúdos sintéticos cada vez mais realistas [19], abrangendo textos, imagens, vídeos e, de maneira particular, áudios e sinais de voz. Esse tipo de produção, conhecido como *Artificial Intelligence Generated Content* (AIGC), tem se beneficiado de modelos de *text-to-speech* e técnicas de clonagem de voz, que permitem a criação de amostras altamente semelhantes às produzidas por seres humanos.

A detecção de áudios *Deepfake* [19] se tornou um tópico emergente e cada vez mais crítico na comunidade de IA [18], devido a tamanha evolução e popularização das ferramentas e tecnologias voltadas à síntese de conteúdo. Dois pilares motivam o presente trabalho, sendo o primeiro a ameaça à integridade e a importância dos sistemas de detecção de *Deepfake*, e o segundo que aborda a necessidade de explorar o paradigma quântico em comparação com modelos clássicos de *Machine Learning*.

O uso mal-intencionado da tecnologia *Deepfake* representa um grave risco para a segurança social, pois tais atitudes abrem margem para fraudes e desinformação, tornando essencial uma identificação eficiente de áudios sintéticos. Dessa maneira, torna-se importante o desenvolvimento de algoritmos robustos e abrangentes, empregando amostras válidas e confiáveis, de modo em que a identificação de *Deepfake* de áudio possa ser aplicada em contextos reais [19].

Apesar da detecção de AIGC ser um campo ativo, há uma disparidade em relação com o desenvolvimento e aperfeiçoamento das tecnologias *Deepfake*, com amostras cada vez mais realistas e casos graves de desinformação.

Tradicionalmente, os esforços se concentram de maneira predominante em uma classificação

binária: Real vs. Falso [18]. Contudo, há um interesse crescente em ir além, abordando tarefas mais complexas, como localizar intervalos manipulados em áudios parcialmente falsos e reconhecer algoritmos de *Deepfake*, encontrando possíveis fontes responsáveis pela geração do áudio falso.

Porém, o presente trabalho tem como foco delimitador de interesse apenas a questão classificação binária do sinal de voz, na tentativa de identificar áudios autênticos e áudios sintéticos gerados por IA.

A Aprendizagem de Máquina Quântica surge como uma área promissora para superar gargalos de eficiência e modelagem [12], e a motivação para a sua exploração se dá a partir das propriedades relacionadas à Computação Quântica, ciência que abrange Física Quântica e a Ciência da Computação. Suas propriedades são:

1. Redução de escala

A superposição e o emaranhamento dos estados quânticos permite que a escala de armazenamento seja reduzida exponencialmente, dado à natureza de um *qubit* [12, 17].

2. Paralelismo quântico

Uma virtude da Computação Quântica que pode acelerar a execução significativamente, principalmente em cenários de Big Data [17].

1.2 OBJETIVOS

O objetivo geral deste trabalho consiste em realizar um estudo comparativo entre o algoritmo *K-Nearest Neighbor* (KNN) [17, 3] em sua versão clássica e sua versão quântica, aplicados à tarefa de detecção de áudios *Deepfake*. Busca-se analisar o comportamento de ambas as abordagens no contexto da classificação de sinais de voz, considerando suas características computacionais, desempenho e adequação ao problema proposto.

1.2.1 Objetivos Específicos

Com o intuito de atingir o objetivo geral estabelecido, definem-se os seguintes objetivos específicos:

- Avaliar o desempenho dos modelos KNN clássico e quântico por meio de métricas de classificação.
- Comparar a eficiência computacional, as limitações e o comportamento de cada abordagem, e identificar as vantagens e as desvantagens de ambas.
- Analisar a viabilidade e a aplicabilidade de soluções baseadas em Aprendizagem de Máquina Quântica em problemas reais, especialmente no contexto da detecção de áudios sintéticos gerados por Inteligência Artificial.

1.3 ORGANIZAÇÃO DO TRABALHO

Ao longo deste trabalho, os capítulos estão organizados conforme segue:

- No Capítulo 2, apresenta-se uma descrição dos principais conceitos abordados no trabalho, abrangendo desde as técnicas de extração de *features* até os conceitos que envolvem Computação Quântica.
- No Capítulo 3, apresenta-se com detalhes todo o desenvolvimento do trabalho proposto e a metodologia em que os conceitos discutidos no capítulo anterior foram utilizados.
- No Capítulo 4, relatam-se todos os resultados obtidos no trabalho a partir dos testes de reconhecimento que foram realizados.
- No Capítulo 5, apresentam-se as conclusões sobre o trabalho e discussões acerca de possíveis trabalhos futuros.

2 REFERENCIAL TEÓRICO

Neste capítulo, são apresentados os fundamentos teóricos que sustentam as técnicas empregadas neste estudo por meio de uma análise detalhada das metodologias adotadas ao longo do desenvolvimento do trabalho. A revisão está organizada em duas vertentes principais: a primeira abrange os métodos de extração de características, incluindo Energia [6], Taxa de Cruzamento por Zero (*Zero-Crossing Rate*) [7], Entropia [8] e os Coeficientes Cepstrais na Escala Mel (MFCC) [20].

A extração de características (*features*) é uma etapa crucial no processamento de sinais digitais e em sistemas de Reconhecimento de Padrões, servindo como uma metodologia fundamental para abstrair as complexidades do mundo real e melhorar a eficácia dos esquemas de classificação [6].

Na segunda vertente, contempla-se os conceitos relacionados ao algoritmo K-Nearest Neighbor (KNN), considerando tanto sua formulação clássica quanto sua abordagem com conceitos quânticos através da hibridização do algoritmo e adoção de *quantum kernels* [16], além da formulação das medidas de distâncias consideradas para o presente trabalho.

Este trabalho, apesar de se basear na premissa de Yijie Dang et al., porém voltado para processamento de sinais digitais, e basear-se nas problemáticas apresentadas por Jiangyan Yi et al., não possui, mediante ampla busca na literatura, trabalhos correlatos que tenham o mesmo objetivo ou similaridades que viabilizem comparações e discussões mais aprofundadas.

2.1 ENERGIA

A energia do sinal se destaca como um conceito simples e, ao mesmo tempo, altamente relevante quando empregado de forma apropriada como ferramenta de extração de características [6].

Na análise de sinais, a definição de energia do sinal é adotada por convenção para qualquer sinal $x(t)$ (tempo contínuo) ou $x[n]$ (tempo discreto), independentemente de estar diretamente relacionada à energia física [13]. Porém, quando voltado à sinais de fala, a energia de um sinal representa a energia necessária para que o locutor produza som, resumindo-se à sua capacidade de realizar trabalho [6].

Formalmente, para sinais unidimensionais (1D), a energia E é calculada como a soma dos quadrados das amplitudes dos pontos do sinal. Para um sinal $s[\cdot]$ de comprimento M , a energia é definida por:

$$E(s[\cdot]) = \sum_{i=0}^{M-1} (s_i)^2, \quad (1)$$

onde M representa o comprimento de $s[\cdot]$ [6].

É importante ressaltar que existem abordagens alternativas e inovadoras para trabalhar com o conceito de energia do sinal. Nos trabalhos referenciados nesta monografia, são propostos três diferentes métodos para caracterizações distintas do sinal sob análise, chamados de A1, A2 e A3 [6]. No entanto, neste estudo, foi utilizado apenas o método A3.

Diferente dos outros métodos, que medem o nível de energia em janelas de tamanho fixo ou variável, o procedimento do A3 determina os comprimentos proporcionais necessários para atingir níveis de energia predefinidos. Para tal, é definido um nível base crítico de energia C ($0 < C < 100$), e o vetor de *features* $f[\cdot]$ é determinado pela proporção do comprimento do sinal, começando do início, que é coberta pela janela para alcançar $C\%$, $2 \cdot C\%$, $3 \cdot C\%$, e assim sucessivamente, da energia total do sinal [6]. O método descrito é caracterizado como ideal para inspecionar a constância na ação da entidade física responsável por gerar o sinal, quantificando a carga de trabalho parcial realizada pelo locutor à medida que o tempo avança.

2.2 TAXA DE CRUZAMENTO POR ZERO

A Taxa de Cruzamento por Zero (ZCR - *Zero-Crossing Rate*), conceito amplamente reconhecido em Processamento de Sinais Digitais, procura recuperar o comportamento espectral dos sinais [7], sendo também uma das ferramentas mais simples para a extração de informação espectral básica, sem a necessidade de conversão explícita para o domínio da frequência.

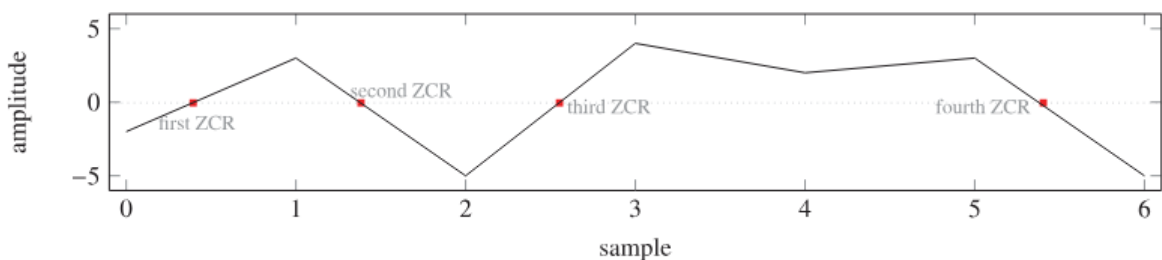


Figura 1 – Ilustração ZCR

Fonte: R.C. Guido (Knowledge-Based Systems 105, 2016, p.248–269).

Em sua forma elementar, a ZCR é definida como o número de vezes que a forma de onda de um sinal cruza a amplitude zero (Figura 1) [7]. Formalmente, para um sinal de tempo discreto unidimensional (1D) $s[\cdot] = \{s_0, s_1, s_2, \dots, s_{M-1}\}$ de comprimento $M > 1$, o número total de cruzamentos por zero é dado por:

$$\text{ZCR}(s[\cdot]) = \frac{1}{2} \sum_{j=0}^{M-2} |\text{sign}(s_j) - \text{sign}(s_{j+1})|, \quad (2)$$

onde a função $\text{sign}(x)$ é definida como 1 se $x \geq 0$ e -1 caso contrário.

Diferentes métodos foram avaliados para este conceito, dentre eles B1, B2 e B3, sendo este último o método selecionado para utilização no presente trabalho.

O B3 é um método similar ao A3, focando na constância da frequência da entidade geradora do sinal [7]. Define-se também um nível base crítico de ZCR C ($0 < C < 100$) e analogamente, determinando o vetor de *features* $f[\cdot]$ coberto pela janela, a partir do início, para atingir $C\%$, $2C\%$, $3C\%$, e assim por diante, do ZCR total do sinal. Este método não utiliza normalizações adicionais, pois a informação já está expressa como uma proporção de comprimento ou área.

2.3 ENTROPIA

A entropia é um conceito fundamental amplamente explorado na literatura científica, que abrange visões distintas e complementares aos conceitos já abordados de energia e ZCR. Enquanto a energia do sinal expressa o potencial para realizar trabalho, e o ZCR se relaciona com o comportamento espectral elementar, a entropia é uma medida da aleatoriedade do sinal [8]. Ela se destaca pela capacidade de revelar características em amostras até mesmo ruidosas, onde outros métodos tendem a falhar.

O primeiro passo envolve determinar a probabilidade p_i de ocorrência do i -ésimo dado distinto (símbolo) em um conjunto de tamanho M que contém K símbolos distintos:

$$p_i = \frac{\alpha_i}{M} \quad . \quad (3)$$

Para o cálculo de entropia [8], temos a seguinte formulação:

$$H = - \sum_{i=0}^{K-1} p_i \cdot \log_{\beta}(p_i) \quad . \quad (4)$$

A base (β) do logaritmo pode ser escolhida de acordo com o contexto. Se $\beta = 2$ for adotada, como neste trabalho, H representa o número médio de bits necessários para representar o sinal digital. Outras bases incluem $\beta \approx e$ (resultando em “nats”) ou $\beta \approx \pi$ (resultando em “slices”). O método C2, por sua vez, busca capturar o sinal em diferentes resoluções. Ele concatena múltiplos subvetores (ξ) obtidos a partir de janelas não sobrepostas [8]. Ademais o sinal é enquadrado usando T janelas não sobrepostas, onde T é definido com base em números primos e a entropia é calculada em cada uma dessas janelas.

Ainda, o vetor de características final $f[\cdot]$ é a concatenação dos subvetores resultantes de cada partição e cada subvetor (ξ_Q) é normalizado separadamente, dividindo seus componentes pelo valor máximo de entropia encontrado naquele subvetor [8].

2.4 MEL-FREQUENCY CEPSTRUM COEFFICIENTS

O *Mel-Frequency Cepstrum Coefficients* (MFCC) constitui uma das representações paramétricas mais difundidas e eficazes no processamento de sinais de fala . Esta técnica fornece uma

representação compacta do sinal acústico [20, 4], cuja superioridade é amplamente reconhecida, especialmente em tarefas de *Speech Recognition*.

O MFCC é obtido a partir de uma transformada cosseno (DCT) aplicada ao logaritmo real do espectro de energia de curto prazo do sinal, após este ter sido mapeado para a escala de frequência mel [20]. A adoção da escala mel é importante, pois tem como objetivo refletir aproximadamente a percepção do ouvido humano [20], além de que a eficácia do MFCC decorre da sua capacidade de representar melhor os aspectos perceptualmente relevantes do espectro de fala de curto prazo.

O cálculo do MFCC envolve uma série de etapas de transformação no domínio da frequência:

1. **Transformada de Fourier (DFT) e Espectro de Potência:** O segmento de fala janelado é transformado para o domínio da frequência usando a Transformada Discreta de Fourier (DFT), resultando no espectro de potência de curto prazo [20].
2. **Deformação (*Warping*) para a Escala Mel:** O espectro de potência $P(f)$ é mapeado para o eixo de frequência mel (M), gerando $P(M)$, utilizando a equação de transformação de Hertz para Mel [20]:

$$M(f) = 2595 \log_{10} \left(1 + \frac{f}{700} \right) . \quad (5)$$

3. **Filtragem em Bandas Críticas:** O espectro deformado é convolucionado com filtros passa-banda triangulares (ou outras formas, como filtros retangulares ou curva de Schroeder) [20]. Em implementações tradicionais, 20 filtros triangulares são utilizados [4]. É importante notar que a sobreposição (overlap) dos filtros adjacentes geralmente resulta em maiores taxas de acerto em comparação com filtros lado a lado (não-*overlapped*) [20]. Além disso, o número e a forma dos filtros são fatores que afetam o desempenho do MFCC.
4. **Transformada Cosseno Discreta (DCT):** Por fim, o MFCC é computado aplicando a DCT nos log-outputs de energia (X_k) de cada filtro [4]:

$$\text{MFCC}(d) = \sum_{k=1}^K X_k \cos \left(\frac{d\pi(2k-1)}{2K} \right), \quad d = 1, \dots, D, \quad (6)$$

onde D é o número total de coeficientes extraídos e K é o número de filtros (ou bandas de frequência) utilizados, e frequentemente $D \ll K$, devido à capacidade de compressão do MFCC.

Portanto, pode-se afirmar que o MFCC é uma *feature* chave em sistemas de reconhecimento de padrões e processamento de fala, possuindo uma base sólida e sendo utilizado em diversos artigos e projetos que envolvem o Processamento Digital de Sinais.

2.5 K-NEAREST NEIGHBOR

Entre os diversos métodos empregados em aprendizado de máquina supervisionado, destaca-se o *K-Nearest Neighbors* (KNN), amplamente utilizado em tarefas de classificação e regressão devido à sua simplicidade conceitual e eficiência prática. Esse algoritmo toma por base a premissa de que amostras com características semelhantes tendem a pertencer à mesma classe ou a apresentar comportamentos próximos, o que o torna uma ferramenta versátil em diferentes domínios de aplicação, como reconhecimento de padrões, processamento de sinais e análise de dados.

O seu princípio operacional se encontra fundamentado em algumas outras premissas intrinsecamente simples [3, 17], como o cálculo da similaridade ou distância entre a amostra de treino e as instâncias do conjunto de treinamento, a seleção dos K vizinhos que possuem a menor distância (ou maior similaridade) e a classificação da amostra de teste de acordo com a classe dominante entre os K vizinhos mais próximos.

Na Figura 2, pode-se visualizar o funcionamento do algoritmo KNN aplicado a um conjunto de dados genérico, considerando duas variáveis preditoras para fins de representação bidimensional. O plano formado por esses atributos é particionado em regiões coloridas que indicam as classes previstas pelo modelo para cada ponto do espaço. Sobre essas regiões são plotadas as amostras reais do conjunto de dados, permitindo uma avaliação direta do desempenho do classificador.

Como principal métrica de distância, temos a Distância Euclidiana, que calcula a distância entre dois pontos em um espaço multidimensional [9]. A Distância Euclidiana representa a menor distância reta entre dois pontos, sendo particularmente útil para avaliar a similaridade entre vetores ou sinais em espaços de alta dimensão. A fórmula da Distância Euclidiana entre dois vetores $\mathbf{x} = (x_1, x_2, \dots, x_n)$ e $\mathbf{y} = (y_1, y_2, \dots, y_n)$ em um espaço n -dimensional é expressa como:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (7)$$

- \mathbf{x} e \mathbf{y} representam os vetores (ou sinais) sendo comparados;
- x_i e y_i são as componentes individuais dos vetores \mathbf{x} e \mathbf{y} ;
- n é o número de *features* do *dataset*.

A Distância de Manhattan se apresenta como uma sólida alternativa usada para a quantificação da similaridade entre os vetores de características. Sua definição, para vetores de n dimensões, como $a = (x_1, x_2, \dots, x_n)$ e $b = (y_1, y_2, \dots, y_n)$, é dada pela soma das diferenças absolutas das coordenadas [9]:

$$D_{\text{MH}}(a, b) = \sum_{i=1}^n |x_i - y_i| \quad (8)$$

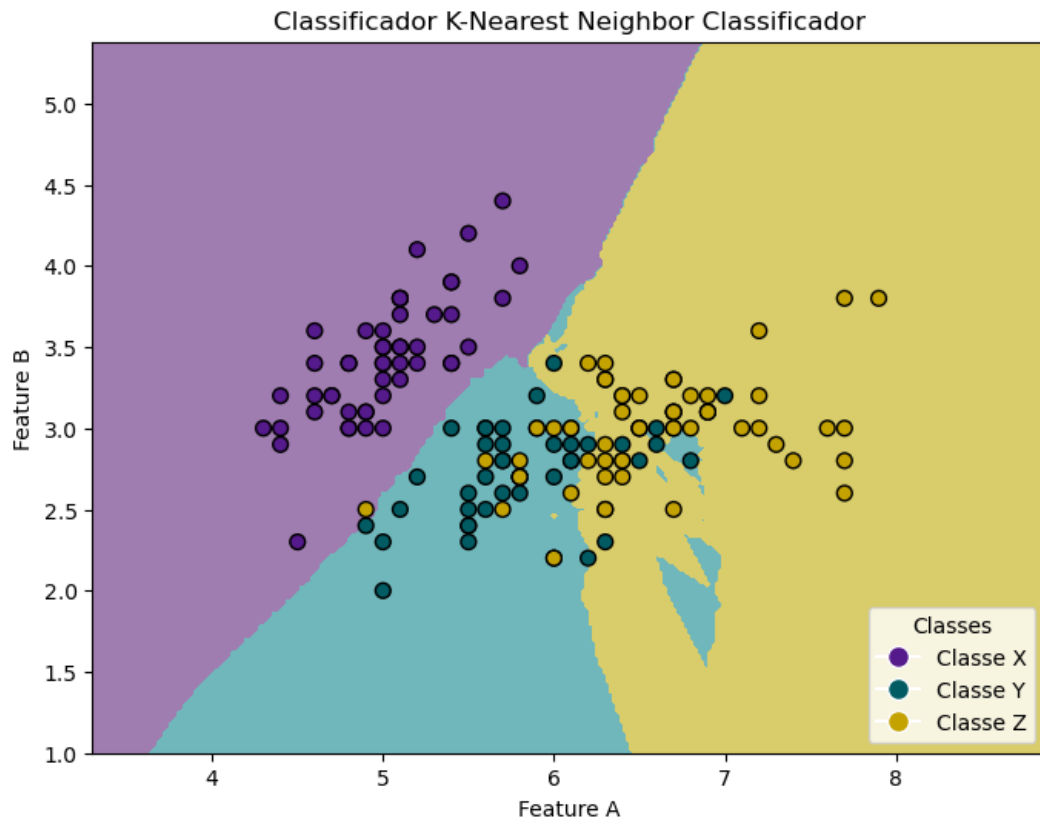


Figura 2 – Demonstração do Classificador KNN
Fonte: Geração própria do autor

Apesar da simplicidade teórica, o KNN possui algumas desvantagens notáveis, especialmente quando está lidando com grandes volumes de dados e características complexas [10, 1]. Portanto, é até mesmo considerado um algoritmo "preguiçoso", dado ao fato de percorrer todo o conjunto de treinamento para a inferência de uma única amostra de teste. Em implementações clássicas, o processo de cálculo de similaridade pode ter complexidade $O(M)$ e o processo de busca ser $O(M \log k)$, onde M é o número de amostras de treinamento [3].

Além do alto custo operacional, o desempenho do KNN se apresenta altamente sensível a alguns problemas relacionados ao conjunto de dados. O conceito de escolher a classe majoritária, naturalmente levanta previsões enviesadas [10]. Ao se deparar com *outliers* (como pontos ruidosos ou distantes) ou *features* irrelevantes, os cálculos de distância e inferências [1] são diretamente prejudicados.

O *K-Nearest Neighbor Quântico* (QKNN) é uma versão quântica do algoritmo clássico KNN, com o objetivo principal de otimizar a eficiência em cenários de *Big Data* [3, 17], pois o QKNN aproveita propriedades quânticas como a superposição e o paralelismo quântico, permitindo o cálculo paralelo de similaridade entre a amostra de teste e todo o conjunto de treinamento, o que, conceitualmente, reduz a complexidade de tempo de linear para constante em comparação com o cálculo clássico [3].

O esquema de classificação envolve a extração de vetores de características em computadores

clássicos; a codificação desses vetores em um estado de superposição quântica [3], e o uso de métricas de distância quânticas, como a derivada do produto interno (fidelidade/distância de cosseno) [17] ou a Distância de *Hamming* [17] para vetores binários. A etapa seguinte acelera o processo de busca pelos k vizinhos mais próximos através do algoritmo quântico de busca de mínimo (como o algoritmo de *Dürr*) [10], após o qual o resultado da classificação, que serve como um bom análogo para a “votação majoritária” do KNN, é determinado por meio de uma única medição quântica.

A medida de dissimilaridade entre dois estados quânticos pode ser obtida pela distância de *Hilbert-Schmidt* [14], uma métrica definida como

$$D_{\text{HS}}(\rho_1, \rho_2) = \sqrt{\text{Tr}[(\rho_1 - \rho_2)^2]}. \quad (9)$$

Para estados puros, isto é, $\rho_i = |\phi_i\rangle\langle\phi_i|$, a expressão simplifica-se para

$$D_{\text{HS}}(|\phi_1\rangle, |\phi_2\rangle) = \sqrt{2(1 - |\langle\phi_1|\phi_2\rangle|^2)}. \quad (10)$$

onde $|\phi(x)\rangle$ é o estado quântico gerado pela codificação *AngleEmbedding*, que será abordada na próxima seção.

Essa métrica representa a distância euclidiana entre as matrizes densidade dos estados quânticos e é frequentemente utilizada para quantificar a similaridade entre estados puros preparados por circuitos de codificação (*embeddings*) em algoritmos quânticos. Sua formulação foi discutida e descrita como uma medida válida de distinção entre estados [14], ainda que não seja contrativa sob operações completamente positivas e traço-preservantes.

2.6 QUANTUM KERNEL

Noisy Intermediate-Scale Quantum (NISQ) é um termo introduzido por John Preskill [16] descreve a próxima geração de dispositivos quânticos compostos por até algumas centenas de *qubits*, capazes de realizar experimentos que escapam à simulação clássica direta, porém ainda fortemente limitados por ruído e por erros nos gates quânticos. É defendido que estes dispositivos devem ser encarados como marcos tecnológicos intermediários: úteis para exploração científica e para validação experimental de algoritmos, mas não como soluções imediatas para problemas práticos em larga escala.

Computadores quânticos têm um potencial natural para simular sistemas quânticos altamente entrelaçados [16], tarefas intratáveis para computadores clássicos, o que abre a chamada *entanglement frontier*. A capacidade de preparar e manipular estados muito entrelaçados pode permitir avanços na química quântica, na física da matéria condensada e na compreensão de dinâmicas quânticas complexas, áreas em que a simulação tradicional esbarra em limitações de escalabilidade. Entretanto, destaca-se que a observação e o controle de sistemas quânticos causam perturbações fundamentais, e para processar informação quântica é necessário isolar os

qubits do ambiente enquanto, simultaneamente, se garante controle e leitura confiáveis.

Esse conflito técnico explica por que o ruído decorrente da decoerência e dos erros de operação, permanece como o principal obstáculo atual, limitando tanto o tamanho dos dispositivos quanto a profundidade dos circuitos executáveis sem correção de erros. O número de *qubits* é apenas um dos parâmetros determinantes; entretanto, a qualidade dos mesmos, a taxa de erro por *gate*, a fidelidade de medição, o tempo de operação e a conectividade entre eles são igualmente relevantes para o poder computacional efetivo dos dispositivos NISQ.

Na prática, dispositivos NISQ tendem a não suportar circuitos profundos porque o acúmulo de erros acaba “afogando” o sinal desejado; por isso, algoritmos de interesse para NISQ costumam priorizar baixa profundidade, resiliência ao ruído ou até mesmo, como no presente trabalho, estratégias híbridas que transferem parte do trabalho para processadores clássicos.

A integração entre técnicas clássicas e quânticas tem se tornado uma abordagem promissora no contexto dos dispositivos NISQ [16], permitindo que algoritmos já consolidados no aprendizado de máquina sejam expandidos para espaços de maior expressividade. Nesse cenário, surge a proposta do KNN com *kernel* quântico, em que os dados clássicos são codificados em estados quânticos, de acordo com as técnicas abordadas na próxima seção, possibilitando que a comparação entre amostras seja realizada não mais no espaço original, mas em um espaço de *Hilbert* de maior dimensionalidade. Essa hibridização mantém a estrutura interpretável do KNN clássico, enquanto explora vantagens potenciais decorrentes de características exclusivamente quânticas, como o entrelaçamento e a manipulação unitária dos dados.

2.7 CODIFICAÇÃO DOS DADOS

O sucesso de qualquer modelo de *Quantum Machine Learning* (QML) depende intrinsecamente da forma como os dados clássicos são mapeados para estados quânticos [12], constituindo essa etapa o primeiro e fundamental processo de preparação dos dados de entrada do modelo. Ainda, algumas estratégias podem ser levadas em consideração nesta etapa, dependendo, basicamente, da natureza dos dados possuídos.

1. *Basis Encoding* - Codificação na Base

Esta primeira técnica é adequada para codificação de dados discretos ou binários (*bitstrings*), onde cada *bitstring* de tamanho m , é convertido diretamente em um autoestado da base computacional usando m *qubits*. Assim, um vetor de características x_n , composto por valores binários $b_i \in \{0, 1\}$, é mapeado para um estado quântico $|x_n\rangle$.

O domínio quântico pode ser descrito por:

$$|X\rangle = \frac{1}{\sqrt{N}} \sum_{n=1}^N |x_n\rangle \quad . \quad (11)$$

2. *Angle Encoding* - Codificação com Rotações

O método permite codificar valores contínuos e arbitrários, rotacionando cada *qubit* de acordo com o valor da *feature* a ser codificada. Cada uma das N características do vetor clássico $x = [x_1, x_2, \dots, x_N]$ é mapeada para um *qubit* distinto [12].

O estado quântico $|x\rangle$ é expresso como um produto tensorial dos estados rotacionados de cada *qubit*:

$$|x\rangle = \bigotimes_{i=1}^N (\cos(x_i)|0\rangle + \sin(x_i)|1\rangle) \quad .$$

3. *Amplitude Embedding* - Codificação nas Amplitudes de Probabilidade

Esta é uma técnica poderosa que mapeia os dados clássicos diretamente nas amplitudes de probabilidade do estado quântico, codificado no espaço de Hilbert \mathcal{H} [12].

$$|\psi_y\rangle = \sum_{i=1}^{2^n} y_i |i\rangle. \quad (12)$$

Sua principal vantagem reside na possibilidade de compactar um grande número de dados em uma única sequência de amplitudes, viabilizando algoritmos de alta dimensão.

Para o presente trabalho, os dados clássicos serão codificados em estados quânticos utilizando a técnica de *Angle Embedding*.

3 METODOLOGIA

Neste capítulo, descrevem-se os procedimentos, técnicas e ferramentas adotados para o desenvolvimento do presente trabalho. Apresentam-se, também, as etapas que compõem o processo metodológico, desde a preparação e organização da base de dados até a implementação e avaliação dos métodos propostos no capítulo anterior.

No diagrama da Figura 3, são ilustradas as principais etapas, a começar pela organização da base de dados que será utilizada, extração das *features* de cada amostra, implementação de ambos os modelos (clássico e quântico) e avaliação dos resultados.

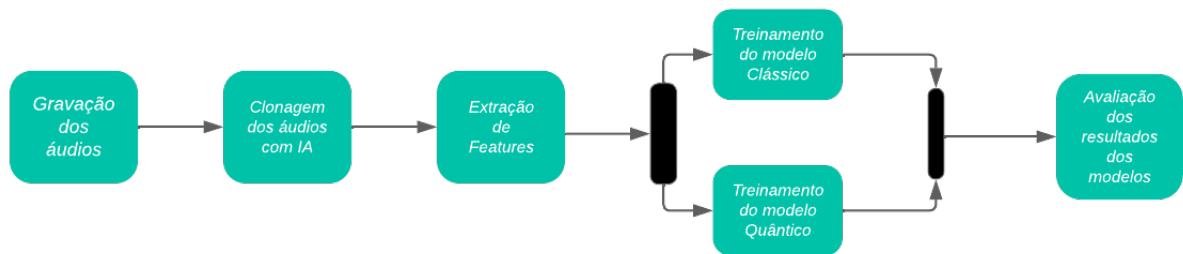


Figura 3 – Diagrama do processo metodológico

Fonte: Geração própria do autor

3.1 BASE DE DADOS

Nos capítulos anteriores, foi elucidado como os modelos de *Machine Learning* se comportam, sejam eles clássicos ou quânticos. Entretanto, a escolha da base de dados é uma etapa crucial neste e em qualquer trabalho na área.

No presente trabalho, visando um conjunto de dados detalhadamente adaptado, optou-se pela geração própria dos dados, não apenas garantindo controle e qualidade acerca das informações, mas também fortalecendo o pressuposto da popularização de agentes de IA com fácil acesso à ferramentas sofisticadas de geração de fala realista e semelhante à voz humana via técnicas de *voice cloning*.

As amostras foram obtidas a partir de 10 pessoas diferentes que, em um ambiente controlado, gravaram alguns minutos de áudio de maneira totalmente aleatória. As gravações foram realizadas utilizando um smartphone Apple iPhone 14 Pro Max que, assim como diversos outros smartphones, emprega tecnologias MEMS (*micro-electromechanical systems*) em seu dispositivo de captura de áudio.

Utilizando o Audacity, um editor de áudio *open-source*, os 10 áudios originais foram subdivididos em outros 10 segmentos de um segundo cada, de maneira totalmente aleatória. Ao final,

obteve-se um total de 100 áudios referentes às 10 pessoas participantes, os quais compõem a classe de áudios autênticos.

Em contraposição aos áudios originais e autênticos, têm-se os áudios *fake*, gerados por IA. Após uma análise das opções disponíveis no mercado, foi selecionada uma plataforma que oferece inúmeras ferramentas relacionadas à geração de fala humana, Eleven Labs [5]. A plataforma possui agentes especializados em *text-to-speech*, oferecendo a opção de *Voice Cloning* [19], que consta em clonar digitalmente a voz de uma pessoa, e o modelo utilizado foi o *Eleven Multilingual v2*.

O áudio original de cada participante foi utilizado como fonte para o processo de clonagem, e alguns parâmetros da plataforma foram ajustados manualmente com o objetivo de maximizar a fidelidade e semelhança dos áudios *fake*. Após a geração de 10 áudios para cada pessoa, aplicaram-se os mesmos procedimentos adotados anteriormente: edição no Audacity e posterior segmentação em 10 trechos individuais de 1 segundo cada.

Dessa forma, conclui-se a construção do dataset próprio. Foram obtidos 100 áudios autênticos de 1 segundo cada e 100 áudios *fake* de mesma duração, totalizando 200 áudios distribuídos em duas classes distintas no conjunto de dados.

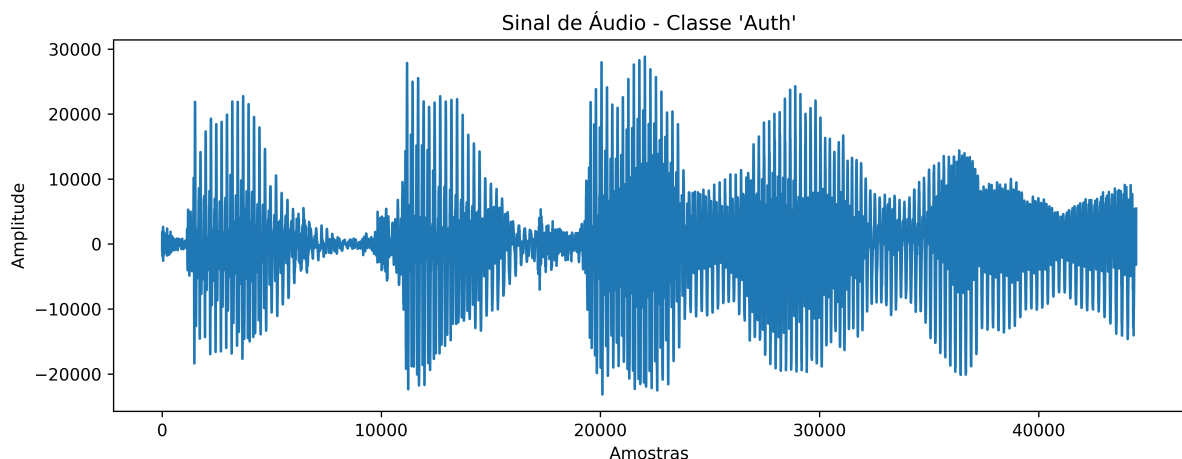


Figura 4 – Forma de onda do áudio da classe *auth*.

Fonte: Geração própria do autor.

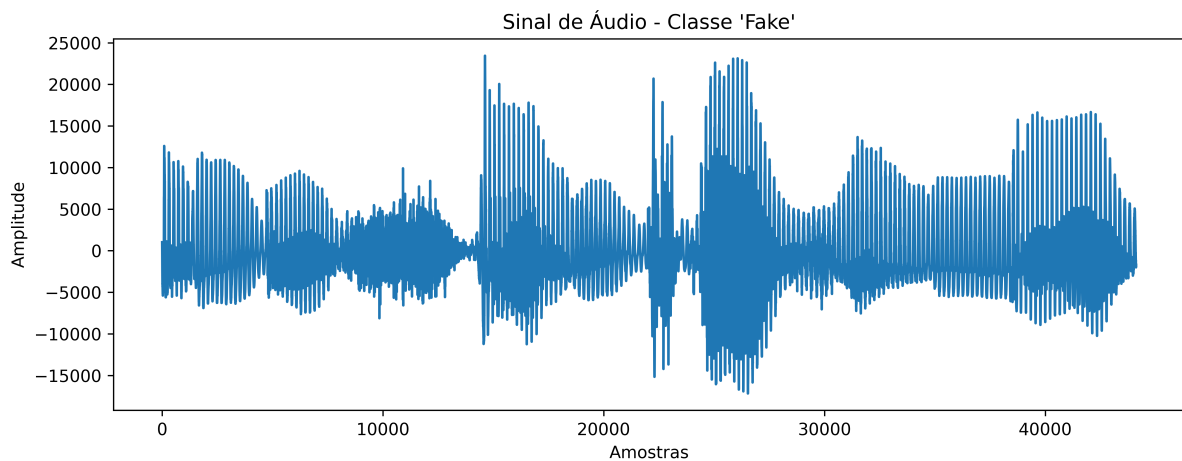


Figura 5 – Forma de onda do áudio da classe *fake*.
Fonte: Geração própria do autor.

Nas Figura 4 e Figura 5, observa-se que, em escala de tempo, as formas de onda dos dois sinais não apresentam padrões que os diferenciem e sejam perceptíveis a olho nu. Ambas apresentam amostras positivas e negativas, além de oscilações de amplitude distribuídas ao longo de todo o sinal, o que torna pouco eficaz a identificação visual de picos, vales e outras regiões e padrões que poderiam distinguir as classes. Dessa forma, a simples inspeção do traçado temporal evidencia uma forte similaridade estrutural entre os sinais, sendo insuficiente para separar com segurança as classes apenas pela aparência das formas de onda.

A avaliação visual das formas de onda, embora útil para verificar integridade e características gerais, é insuficiente para concluir sobre distinções entre classes. Portanto, a extração de *features* visa capturar todas as propriedades relevantes do sinal, sejam elas recursos no domínio de tempo e/ou frequência.

3.2 EXTRAÇÃO DE *FEATURES*

Nesta etapa, os segmentos de áudio obtidos anteriormente serão convertidos em informações relevantes para o processo de análise e aprendizagem dos modelos. Isso se dará por meio das técnicas conceituadas no Capítulo 2 do presente trabalho.

O primeiro passo desta etapa consiste na implementação das funções descritoras de características: Energia (A3), *Zero-Crossing Rate* (B3) e Entropia (C2). Cada uma dessas métricas foi aplicada a partir das formulações apresentadas nos artigos referenciados neste trabalho, que serviram como referência teórica. Para a extração dos MFCCs, foram utilizadas as funções já implementadas na versão 0.11.0 da biblioteca *Librosa* [11], por meio da linguagem de programação *Python*.

Em sequência, utilizando a biblioteca *Librosa*, os arquivos de áudio foram tratados como sinais em séries temporais. A partir disso, obteve-se um *dataset* composto por 200 vetores de sinais, sobre os quais foram extraídas as *features*.

Então, uma função de normalização foi aplicada apenas aos valores extraídos dos MFCCs, visto que as funções A3, B3 e C2 normalizam automaticamente os vetores de características para o *output*. Para padronizar a escala dos valores numéricos, foi aplicada ao *MinMaxScaler*, disponibilizada pela versão 1.6.1 da biblioteca *Scikit-learn* [15]. Esse método consiste em reescalar os dados para um intervalo predefinido, geralmente entre 0 e 1, preservando-se as relações proporcionais entre os valores originais.

É importante ressaltar que o processo de extração de *features* pode ser lento e demorado, demandando um alto esforço computacional. Na execução do presente trabalho, esta etapa exigiu um período de aproximadamente 24 horas até o fim da sua execução.

Como resultado da etapa de extração de características, obteve-se então um conjunto de dados robusto, em que cada amostra é representada por um vetor de 92 atributos, sendo 19 do método A3, 19 do método B3, 28 do método C2 e 26 atributos dos MFCCs. Essas *features* alimentam o modelo preditivo, permitindo que ele identifique diferenças e padrões que são difíceis de serem detectados manualmente. Variações sutis já podem ser observadas na análise espectral: como exemplo, os coeficientes MFCC evidenciam discrepâncias iniciais entre os espectros de áudios pertencentes a classes distintas. A Figura 6 ilustra essas diferenças, manifestadas por todas características tipicamente capturadas pelos MFCCs.

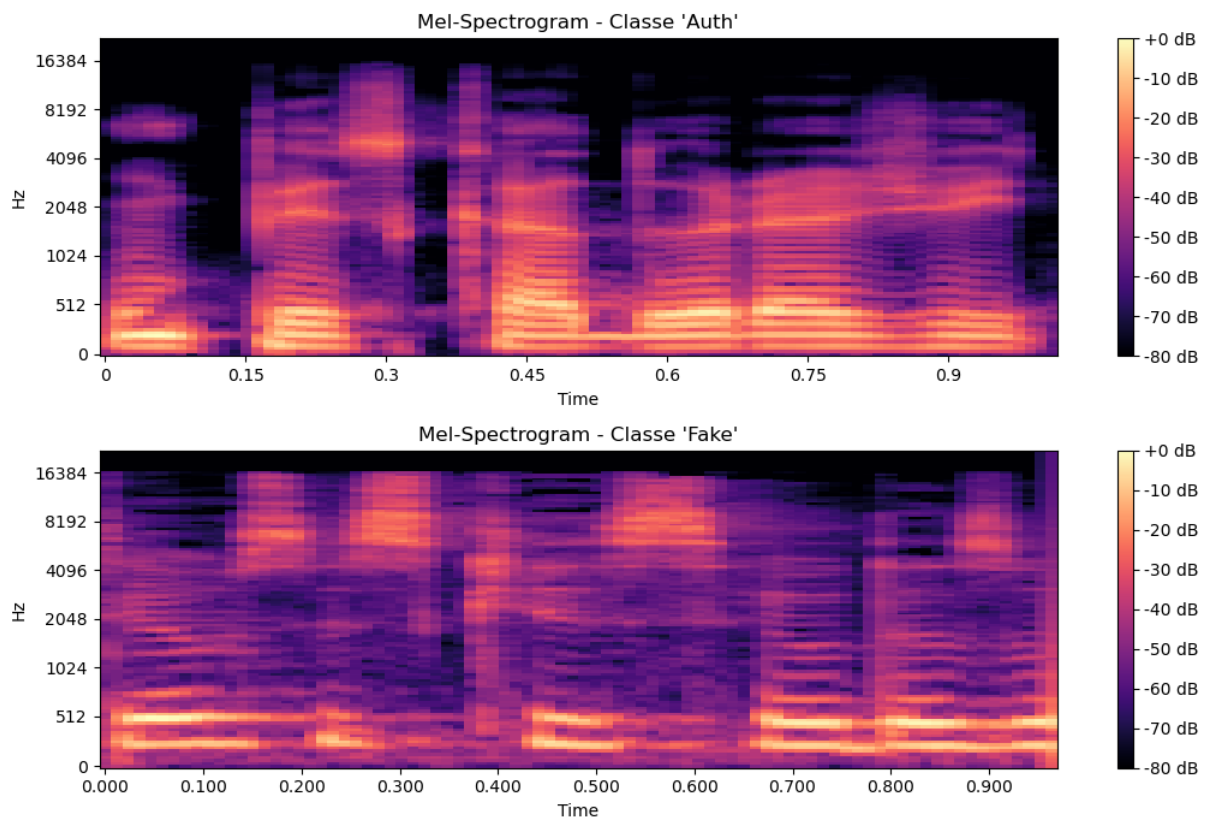


Figura 6 – Comparação entre os espectrogramas Mel das duas classes analisadas.

Fonte: Geração própria do autor.

Assim sendo, o vetor de atributos resultante sintetiza e estrutura todas as informações

relevantes do sinal necessárias para o treinamento dos modelos.

3.3 K-NEAREST NEIGHBOR CLÁSSICO

Com as características extraídas e o *dataset* devidamente estruturado, a etapa seguinte consiste na organização dos dados e no treinamento do modelo de *Machine Learning* baseado no KNN clássico.

3.3.1 Hiperparâmetros e Pré-processamento

Para otimizar o desempenho do modelo de classificação, foi empregada a função *GridSearchCV* da biblioteca *Scikit-learn*. Essa ferramenta realiza uma busca exaustiva por combinações de hiperparâmetros previamente definidos, avaliando cada configuração por meio de validação cruzada. Dessa forma, o *GridSearchCV* identifica automaticamente os valores que proporcionam o melhor desempenho ao modelo, reduzindo a necessidade de ajustes manuais e minimizando riscos de *overfitting*.

A Tabela 1 reúne os valores definidos para os principais hiperparâmetros empregados no *GridSearchCV* e descreve a finalidade de cada parâmetro no processo de ajuste do modelo.

Hiperparâmetro	Descrição	Valores testados
n_neighbors	Número de vizinhos mais próximos	{1, 3, 5}
weights	Função de ponderação dos vizinhos	{'uniform', 'distance'}
metric	Métrica de cálculo de distância	{'euclidean', 'manhattan', 'cosine'}

Tabela 1 – Hiperparâmetros utilizados no *GridSearchCV* para o modelo KNN clássico.

Executando a função representado no Código 3.1, temos, então, o conjunto de hiperparâmetros com os melhores resultados.

```

1 from sklearn.neighbors import KNeighborsClassifier
2 from sklearn.model_selection import GridSearchCV
3 from sklearn.model_selection import train_test_split
4
5 X_train, X_test, Y_train, Y_test = train_test_split(feas, classes,
6     test_size=0.2, random_state=42)
7 knn = KNeighborsClassifier()
8 #param_grid, utilizado no GridSearchCV
9 param_grid = [{
10     'n_neighbors': [1, 3, 5],
11     'weights': ['uniform', 'distance'],
12     'metric': ['euclidean', 'manhattan', 'cosine']
13 }]

```

```

14 gs = GridSearchCV(KNeighborsClassifier(), param_grid=param_grid, cv=10,
    scoring='accuracy', verbose=5)
15 gs.fit(X_train, Y_train)
16 best_params = gs.best_params_

```

Código 3.1 – Escolha de Hiperparâmetros com *GridSearchCV*

A classe instanciada executa o processo de *cross-validation* de 10 *folds*, avaliando por meio da acurácia o classificador KNN enquanto percorre todas as combinações de hiperparâmetros definidas no espaço de busca.

No processo de otimização da métrica de distância do modelo KNN, também foi avaliada a similaridade por cosseno. Entretanto, seus resultados apresentaram desempenho inferior em comparação às demais opções. Dessa forma, a escolha final concentrou-se entre as distâncias Euclidiana e de Manhattan, e após as validações cruzadas realizadas, a distância de Manhattan demonstrou maior acurácia média, sendo portanto selecionada como métrica adotada na configuração final do classificador. Além disso, definiu-se que a classificação das amostras será realizada considerando os 5 vizinhos mais próximos, adotando-se uma ponderação uniforme na contribuição de cada vizinho para a determinação da classe.

Para analisarmos o modelo de forma equivalente ao QKNN, foi empregado a técnica de Análise de Componentes Principais (PCA), onde projetamos os dados em um novo espaço de menor dimensionalidade, preservando as características mais relevantes para a classificação. Após o processo de seleção, optou-se por manter um total de 10 componentes principais.

```

1 from sklearn.decomposition import PCA
2 pca = PCA(n_components=10)
3 X_pca = pca.fit_transform(X)
4 print(X_pca.shape)
5 >>> (200, 10)

```

Código 3.2 – Aplicação do PCA na Base de dados

3.3.2 Treinamento do modelo

Com os hiperparâmetros definidos na seção anterior e o PCA aplicado no dataset, procedeu-se à etapa de treinamento do modelo KNN clássico. O conjunto de dados foi particionado em treino e teste por meio da função *train_test_split*, disponibilizada pela biblioteca *Scikit-learn*.

Para garantir maior robustez estatística e mitigar o risco de sobreajuste, os processos de particionamento do *dataset*, treinamento do modelo e avaliação dos resultados foram repetidos diversas vezes, variando-se a *random_seed* a cada execução, de modo a treinar e testar o classificador sobre diferentes porções dos dados disponíveis.

Em cada execução, diferentes métricas de avaliação foram registradas, permitindo uma análise mais detalhada do desempenho do modelo posteriormente.

3.4 *K-NEAREST NEIGHBOR* QUÂNTICO

Nesta etapa, será apresentado o desenvolvimento e a implementação das técnicas quânticas no classificador KNN, abrangendo o pré-processamento de dados e a codificação dos dados clássicos em estados quânticos até a utilização de um *quantum kernel* para o treinamento do modelo e avaliação de desempenho.

3.4.1 Pré-processamento dos dados

A hibridização entre componentes quânticos e clássicos pode gerar elevados custos computacionais nos pontos onde arquiteturas tradicionais deixam de ser eficientes. A simulação quântica, ao utilizar qubits e explorar seu espaço de estados altamente dimensional, impõe um custo computacional expressivo. Por esse motivo, torna-se necessária a aplicação de técnicas de redução de dimensionalidade, de forma a preservar apenas os atributos mais relevantes ao modelo, mantendo o equilíbrio entre desempenho e viabilidade computacional.

Empregando-se a técnica de Análise de Componentes Principais (PCA), foi possível projetar os dados em um novo espaço de menor dimensionalidade, preservando as características mais relevantes para a classificação. Posteriormente, cada feature da base de dados irá corresponder diretamente a um *qubit* no circuito quântico.

A etapa de pré-processamento dos dados é concluída com a aplicação da mesma função utilizada anteriormente para normalização, ao *MinMaxScaler*, garantindo que todas as *features* permaneçam na mesma escala antes do treinamento do modelo.

3.4.2 Construção do *Kernel* Quântico

No capítulo de referencial teórico, é discutido sobre os dispositivos NISQ e apresentado as motivações para a hibridização em algoritmos de QML. Nesse contexto, adota-se neste trabalho uma abordagem híbrida por meio da incorporação de um *Kernel* Quântico ao modelo clássico de KNN.

Na classe *KNeighborsClassifier*, utilizada para o desenvolvimento de modelos baseados no algoritmo KNN, há o parâmetro *metric*, responsável por definir a função de cálculo de distância entre as amostras. No KNN clássico, esse parâmetro assume valores correspondentes a funções tradicionalmente empregadas, como a distância Euclidiana ou Manhattan, já implementadas internamente na biblioteca. Entretanto, na proposta deste trabalho, a métrica será substituída por uma função personalizada, capaz de codificar os dados em estados quânticos e calcular a distância também em domínio quântico.

A etapa inicial consiste na inicialização do ambiente quântico, a partir da definição do número de *qubits*, como já abordado na subseção anterior. No Código 3.3, inicializa-se um dispositivo quântico (simulador) da biblioteca *Pennylane* [2].

```
1 dev_kernel = qml.device("lightning.qubit", wires=n_qubits)
```

Código 3.3 – Inicialização do dispositivo quântico

O *lightning.qubit* é um simulador de vetor de estado [12] significativamente mais rápido, implementado em C++, sendo uma excelente opção para um número mediano de qubits, tal qual ocorre no presente trabalho. Além disso é criada uma matriz projetora que irá atuar no estado fundamental de todos os 10 *qubits*.

Para a construção do kernel quântico, cada vetor de entrada é transformado em um estado quântico por meio da camada *AngleEmbedding*, que aplica rotações *RY* para codificar cada componente do vetor em um *qubit*. No Código 3.4, o circuito implementado aplica a operação adjunta correspondente ao segundo vetor, permitindo comparar os dois estados dentro do próprio espaço quântico. Após essa composição, calcula-se a expectativa do projetor no estado base, cujo resultado indica o grau de similaridade entre as duas codificações. Esse valor corresponde à fidelidade entre os estados quânticos gerados e é adotado como o kernel utilizado no modelo, servindo posteriormente como medida de proximidade no classificador k-NN.

```

1 @qml.qnode(dev_kernel)
2 def kernel(x1, x2):
3     """The_Quantum_Kernel"""
4     AngleEmbedding(x1, wires=range(n_qubits), rotation="Y")
5     qml.adjoint(AngleEmbedding)(x2, wires=range(n_qubits), rotation="Y")
6     return qml.expval(qml.Hermitian(projector, wires=range(n_qubits)))

```

Código 3.4 – Implementação do Kernel Quântico

Para o cálculo da distância, iremos utilizar a similaridade dada pelo *kernel* e aplicar à fórmula de *Hilbert-Schmidt*, já conceituada no capítulo anterior:

```

1 def dist(x1, x2):
2     return np.sqrt(2*(1-kernel(x1, x2)))

```

Código 3.5 – Implementação da função de distância Hilbert-Schmidt

Com isso, o modelo KNN é instanciado utilizando a função personalizada do *kernel* quântico e segue o mesmo fluxo de processamento do KNN clássico. O modelo é treinado em 100 execuções independentes, cada uma inicializada com uma *random_seed* distinta, de modo a garantir robustez estatística. Todos os resultados obtidos são armazenados para posterior análise e avaliação comparativa.

4 RESULTADOS

A partir dos testes realizados, será abordado neste capítulo os resultados obtidos. Todos os experimentos foram executados em uma máquina local, utilizando a linguagem de programação *Python*. Para o pré-processamento dos dados, avaliação das métricas e instanciação dos modelos classificadores, empregaram-se bibliotecas como o *Scikit-learn*. Além disso, utilizou-se a biblioteca *PennyLane*, que é um ecossistema para funções de QML e suas simulações quânticas. As visualizações gráficas dos resultados foram produzidas por meio das funções do módulo *Pyplot* da biblioteca *Matplotlib*, complementadas pelos recursos da biblioteca *Seaborn*.

Os modelos foram submetidos a um esquema de treinamento repetido com o objetivo de aumentar a robustez estatística da avaliação. Ao todo, foram realizadas 100 execuções independentes em cada um dos modelos (clássico e quântico) e, em cada uma delas, a base de dados foi dividida novamente em conjuntos de treino e teste, variando tanto a proporção dessa divisão quanto a *random seed* utilizada. Essa abordagem garantiu que os modelos fossem expostos a diferentes combinações de amostras, reduzindo a influência de partições específicas e permitindo observar o comportamento médio em cenários variados de ambos os classificadores. Dessa forma, os resultados obtidos refletem de maneira mais confiável o desempenho geral dos modelos, mitigando os efeitos de aleatoriedade e possíveis vieses associados a uma única divisão da base.

Na avaliação do desempenho dos classificadores, utilizou-se a matriz de confusão, que organiza os resultados em quatro categorias fundamentais, os *True Negatives* (Verdadeiros Negativos) representam os casos em que o modelo corretamente identificou um áudio autêntico como tal; os *False Positives* (Falsos Positivos) que correspondem, por sua vez, às situações em que o modelo classificou um áudio como *fake*, embora ele fosse autêntico; os *False Negative* (Falsos Negativos), que ocorrem quando o modelo afirmou que o áudio era autêntico, mas ele pertencia à classe de áudios *fake*; e, por fim, os *True Positives* (Verdadeiros Positivos), que indicam acertos do modelo ao identificar corretamente áudios gerados por IA.

Considerando que a classe positiva corresponde aos áudios falsificados, as métricas derivadas dessa matriz permitem entender o comportamento do modelo de maneira mais aprofundada. A precisão indica a proporção de áudios falsos corretamente classificados como *fake*, sendo sensível ao número de falsos positivos, analisando a qualidade das previsões positivas do modelo. Logo, uma maior precisão significa menos casos em que um áudio autêntico foi erroneamente marcado como falso. Já o *recall* mede a capacidade do modelo de encontrar todas as instâncias positivas reais, refletindo diretamente os falsos negativos; quando elevado, significa que poucos áudios falsificados foram classificados incorretamente como autênticos. O *F1-score*, que harmoniza precisão e *recall*, torna-se útil especialmente em cenários de possível desequilíbrio entre classes ou quando tanto FP quanto FN possuem importância similar. Assim, diferenças entre essas métricas revelam se o modelo adota uma postura mais conservadora ou mais agressiva ao

classificar um áudio como *fake*.

4.1 APRESENTAÇÃO DO RESULTADOS DO KNN CLÁSSICO

Com o classificador KNN Clássico sendo executado 100 vezes, os resultados foram salvos e para cada execução foram calculadas as métricas de acurácia, precisão, *recall* e *F1-score*, apresentadas nas Tabelas 2 e 3.

Métrica	Média	Desvio Padrão
Acurácia	63,57%	7,10%
Precisão	63,00%	11,12%
Recall	62,73%	10,04%
F1-score	62,16%	8,38%

Tabela 2 – Médias e desvios padrão das métricas para o KNN Clássico

Métrica	Mínimo	Máximo
Acurácia	45,50%	77,50%
Precisão	35,00%	88,89%
Recall	37,50%	83,34%
F1-score	38,70%	80,85%

Tabela 3 – Valores mínimo e máximo das métricas do KNN Clássico

A Figura 7 ilustra a distribuição das métricas ao longo das 100 execuções do KNN Clássico.

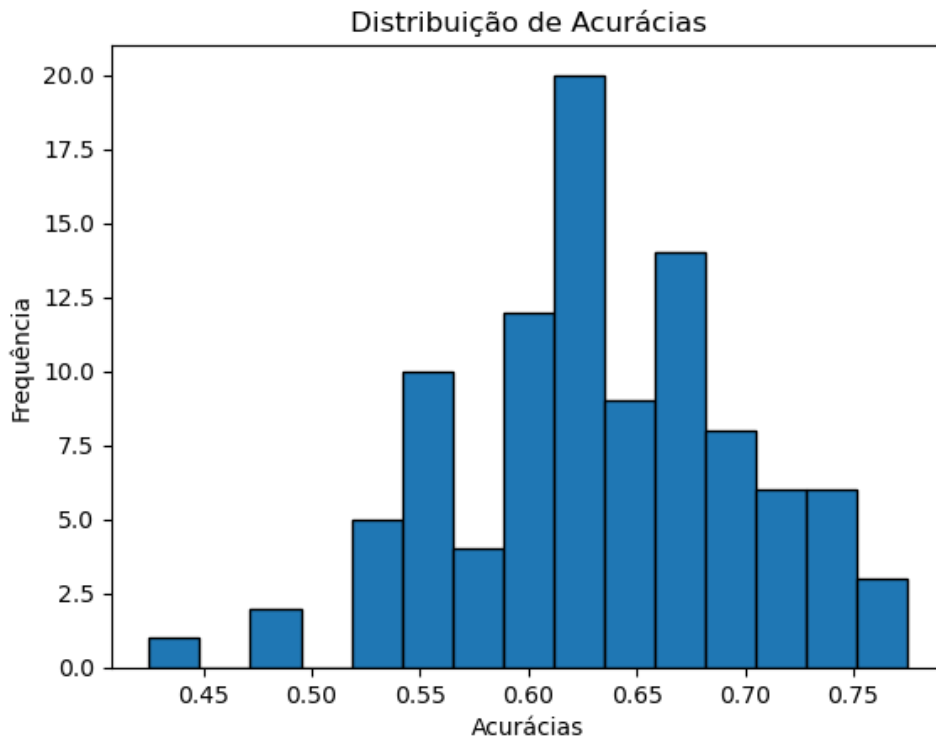


Figura 7 – Histograma de acurácia do KNN Clássico.

Fonte: Geração própria do autor.

4.2 APRESENTAÇÃO DO RESULTADOS DO KNN QUÂNTICO

Da mesma forma, o classificador KNN Quântico foi executado 100 vezes.

São apresentados, nas Tabelas 4 e 5, os valores estatísticos médio, mínimo, máximo e desvio padrão para a Acurácia, Precisão, *Recall* e *F1-Score*. Ademais, a Figura 8 ilustra a frequência das acurácias através de um gráfico de histograma.

Métrica	Média	Desvio Padrão
Acurácia	55,15%	6,26%
Precisão	54,32%	10,62%
Recall	56,36%	10%
F1-score	54,50%	7,80%

Tabela 4 – Médias e desvios padrão das métricas para o KNN Quântico

Métrica	Mínimo	Máximo
Acurácia	37,50%	67,50%
Precisão	30%	85,71%
Recall	34,78%	80%
F1-score	35,89%	69,77%

Tabela 5 – Valores mínimo e máximo das métricas do KNN Quântico

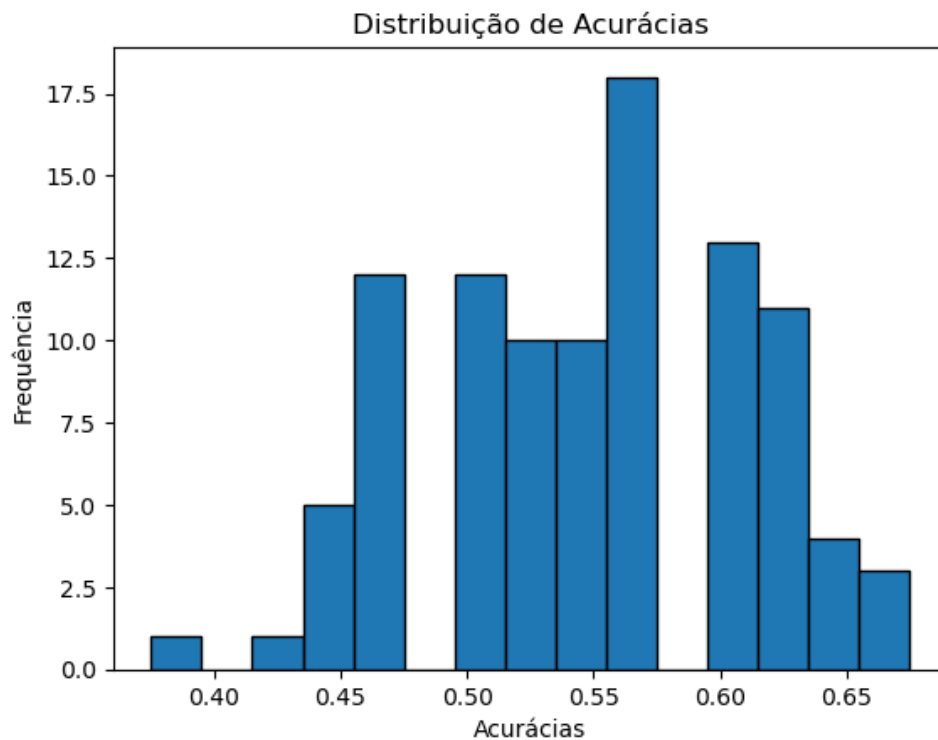


Figura 8 – Histograma de acurácia do KNN Quântico.

Fonte: Geração própria do autor.

4.3 COMPARAÇÃO ENTRE OS MÉTODOS

Nesta seção, comparam-se os resultados das métricas de avaliação dos dois métodos de classificadores. A Tabela 6 apresenta a comparação entre as médias das métricas dos dois modelos e a diferença entre estes valores.

Métrica	KNN Clássico	KNN Quântico	Diferença
Acurácia	63,57%	55,15%	8,42%
Precisão	63,00%	54,32%	8,68%
Recall	62,73%	56,36%	6,37%
F1-score	62,16%	54,50%	7,66%

Tabela 6 – Comparação das métricas médias entre os classificadores

4.3.1 Visualizações

Para complementar a análise numérica apresentada na Tabela 6, inclui-se nesta subseção um conjunto de visualizações que tem como objetivo ilustrar de forma mais clara o comportamento dos dois modelos ao longo das 100 execuções realizadas.

O histograma da Figura 9 contém a distribuição das acurácias obtidas pelos dois classificadores ao longo das 100 execuções, unindo a Figura 7 e Figura 8. Ele permite visualizar como os resultados de cada modelo se concentram ao longo do intervalo observado, destacando diferenças na frequência de valores altos e baixos, bem como a tendência geral de desempenho de cada abordagem.

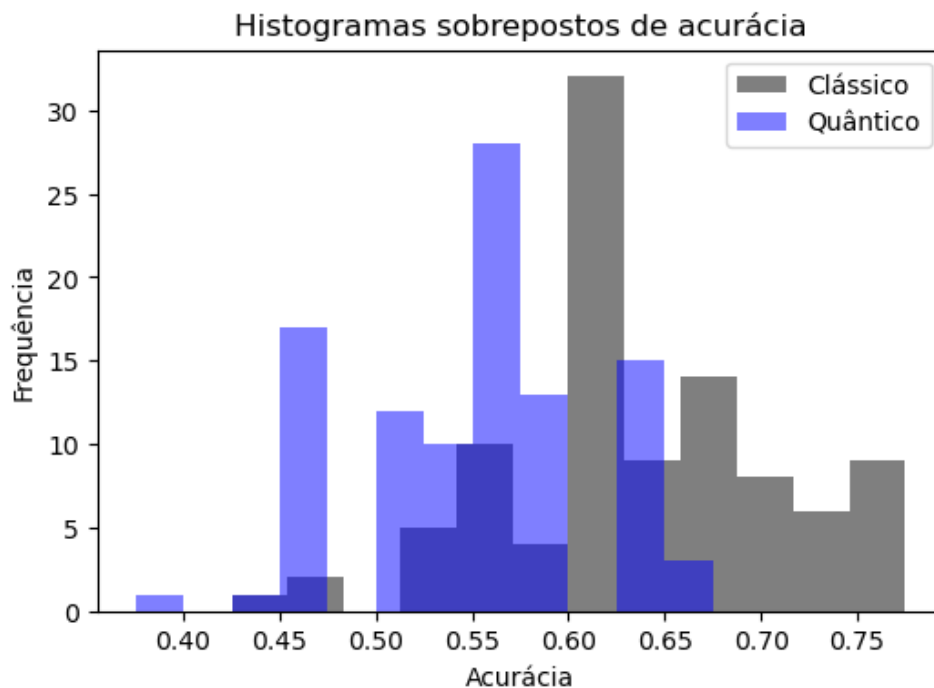


Figura 9 – Histograma de acurácia dos métodos KNN Clássico e QKNN.

Fonte: Geração própria do autor.

No *boxplot* comparativo da Figura 10, ilustra-se a dispersão, mediana e a presença de possíveis outliers para cada modelo, facilitando a identificação de diferenças na variabilidade e na consistência dos resultados entre o KNN Clássico e o KNN Quântico.

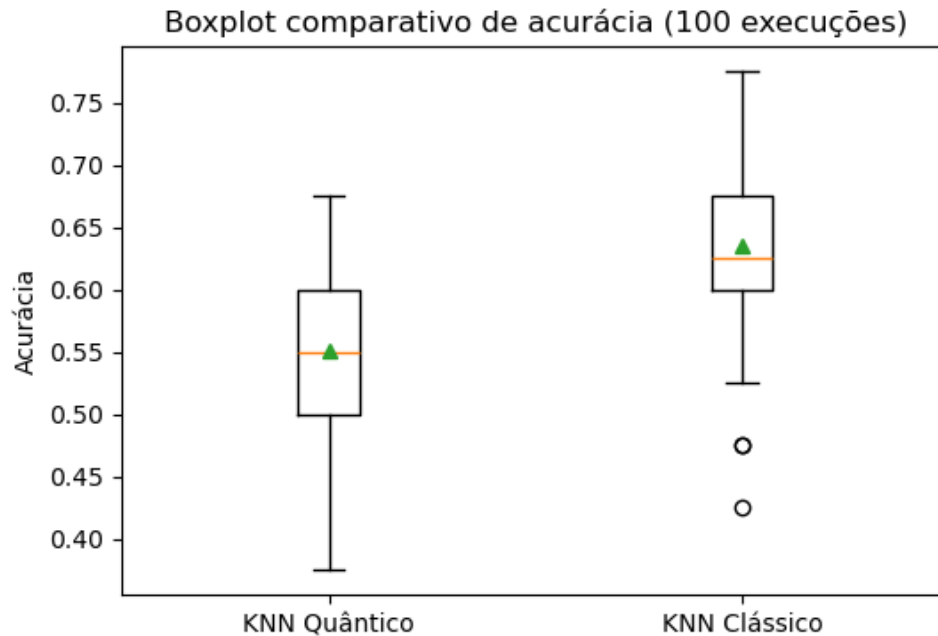


Figura 10 – *Boxplot* comparativo das acurácias dos métodos KNN Clássico e QKNN.
Fonte: Geração própria do autor.

O gráfico de *scatter* (Figura 11) pareado mostra, para cada execução, a relação direta entre o desempenho do KNN Clássico e do KNN Quântico. Cada ponto representa uma execução, permitindo observar em quantas ocasiões um modelo superou o outro e quão distantes os valores estão entre si, oferecendo uma visão clara da comparação ponto a ponto entre os classificadores.

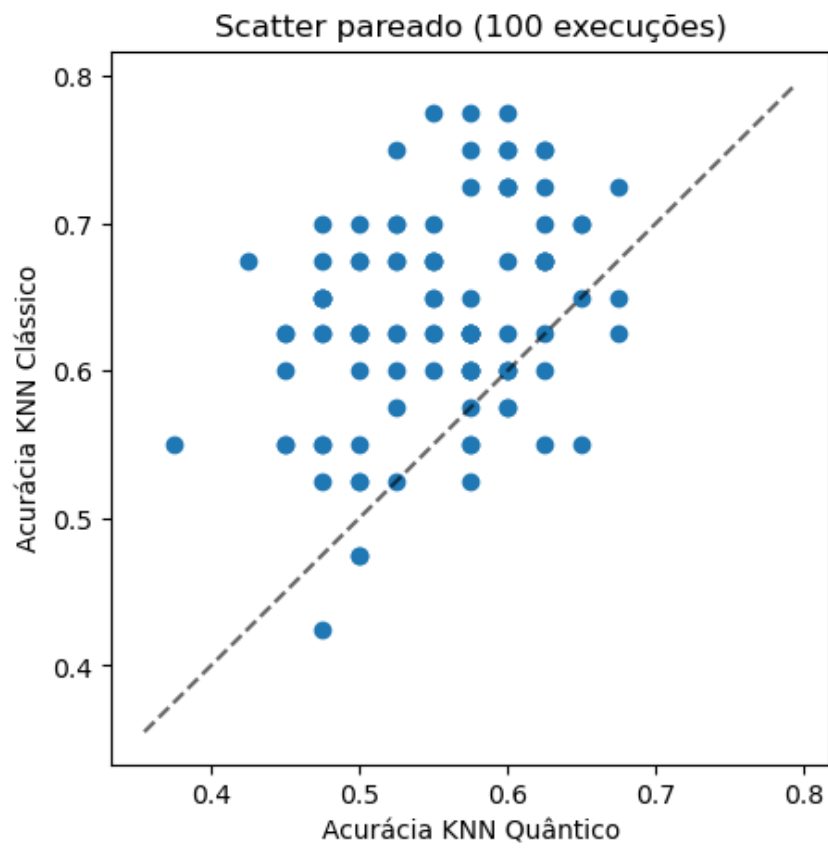


Figura 11 – *Scatter* pareado da acurácia em cada execução dos métodos KNN Clássico e QKNN.
Fonte: Geração própria do autor.

5 CONCLUSÃO

Teve-se como objetivo, no presente trabalho, comparar duas abordagens do classificador KNN: a implementação tradicional, baseada nos métodos clássicos do modelo, e uma variação que incorpora um *kernel* quântico aplicado ao mesmo algoritmo.

Os resultados obtidos indicam que a abordagem clássica do KNN apresentou desempenho superior nas condições analisadas, mostrando-se eficiente na classificação de áudios autênticos e sintéticos, alcançando uma acurácia máxima de 63,57%, com uma diferença de apenas 8,42% a mais que a abordagem quântica. Além de possuir um custo computacional extremamente baixo, para este dataset. Em contraste, embora os métodos quânticos apresentem potencial para alcançar melhores resultados em cenários mais complexos, o desempenho observado neste estudo ficou aquém do esperado, tendo uma acurácia média de 55,15%. Ademais, o custo computacional do método quântico demonstrou ser substancialmente maior, uma vez que a abstração e o processamento de qubits em máquinas clássicas demandam maior tempo de execução, aproximadamente 100 vezes superior ao da abordagem tradicional.

Cabe destacar que o desempenho inferior do QKNN neste estudo não invalida seu potencial, mas reforça a importância de realizar um ajuste criterioso dos parâmetros do modelo, ampliar o conjunto de dados e explorar diferentes configurações de arquitetura, especialmente considerando a metodologia adotada, que consistiu na implementação de um *kernel* quântico na classe KNN. A aplicação de modelos de QML demanda não apenas maior investimento em recursos computacionais, mas também um planejamento cuidadoso para que seu potencial seja plenamente aproveitado, dado o caráter altamente sensível e custoso das operações envolvendo simulação e manipulação de *qubits* em ambientes clássicos.

Portanto, este estudo contribui para a compreensão das vantagens e limitações entre a abordagem clássica do KNN e sua variação com *kernel* quântico, ressaltando a importância de avaliar cuidadosamente os recursos computacionais, a complexidade do problema e os requisitos do ambiente antes de optar por técnicas clássicas ou por métodos baseados em computação quântica.

5.1 TRABALHOS FUTUROS

Com base nos resultados identificados neste estudo, diversas direções para trabalhos futuros podem ser exploradas com o intuito de aprimorar as abordagens quânticas para a detecção de conteúdos gerados por Inteligência Artificial.

A abordagem mais simples consiste na ampliação e diversificação da base de dados, seja por meio da coleta de um número maior de amostras ou pela utilização de diferentes ferramentas de geração e clonagem de voz por IA. Além da ampliação do conjunto de amostras, considera-se relevante também investigar novas características extraídas a partir do próprio dataset, de forma

a ampliar a variabilidade e a representatividade das informações presentes nos áudios analisados.

Outra direção promissora para trabalhos futuros envolve o aprimoramento do kernel quântico utilizado no QKNN, por meio da exploração de diferentes estratégias de mapeamento e codificação, como angle embedding, amplitude embedding e diversos feature maps, além da avaliação de arquiteturas alternativas de circuitos quânticos e *kernels* híbridos. Complementarmente, recomenda-se comparar a performance do modelo em diferentes ambientes de execução, incluindo simulações em CPU e GPU, aceleradores específicos e hardware quântico disponível na nuvem, a fim de avaliar de forma abrangente os *trade-offs* entre precisão, tempo de processamento e viabilidade prática dessas abordagens.

REFERÊNCIAS

- [1] Ali A Amer, Sri Devi Ravana e Riyaz Ahamed Ariyaluran Habeeb. “Effective k-nearest neighbor models for data classification enhancement”. Em: *Journal of Big Data* 12.1 (2025), p. 86.
- [2] Ville BERGHOLM et al. *PennyLane*. PennyLane. USA, 2018. URL: <https://arxiv.org/abs/1811.04968>.
- [3] Yijie Dang et al. “Image classification based on quantum K-Nearest-Neighbor algorithm”. Em: *Quantum Information Processing* 17.9 (2018), p. 239.
- [4] Li Deng e Douglas O’Shaughnessy. *Speech processing: a dynamic and optimization-oriented approach*. CRC Press, 2003.
- [5] ElevenLabs. *Plataforma de síntese de voz baseada em inteligência artificial*. 2025. URL: <https://elevenlabs.io> (acesso em 21/12/2025).
- [6] Rodrigo Capobianco GUIDO. “A tutorial on signal energy and its applications”. Em: *Neurocomputing* 264 (2016a), pp. 264–282. DOI: 10.1016/j.neucom.2015.12.012.
- [7] Rodrigo Capobianco GUIDO. “ZCR-aided neurocomputing: A study with applications”. Em: *Knowledge-Based Systems* 105 (2016b), pp. 248–269.
- [8] Rodrigo Capobianco Guido. “A tutorial review on entropy-based handcrafted feature extraction for information fusion”. Em: *Information Fusion* 41 (2018), pp. 161–175.
- [9] MD Malkauthekar. “Analysis of euclidean distance and manhattan distance measure in face recognition”. Em: *Third International Conference on Computational Intelligence and Information Technology (CIIT 2013)*. IET. 2013, pp. 503–507.
- [10] Youness Manzali et al. “A feature weighted K-nearest neighbor algorithm based on association rules”. Em: *Journal of Ambient Intelligence and Humanized Computing* 15.7 (2024), pp. 2995–3008.
- [11] Brian McFee et al. “librosa: Audio and Music Signal Analysis in Python”. Em: *SciPy*. 2015. URL: doi.org.
- [12] Jogi Suda NETO et al. *Aprendizado de Máquina Quântica: Teoria e Aplicações*. 1ª ed. CBNA, 2025. ISBN: 978-65-01-33992-4.
- [13] Alan V Oppenheim e George C Verghese. *Signals, systems & inference*. Pearson London, 2017.
- [14] Masanao Ozawa. “Entanglement measures and the Hilbert–Schmidt distance”. Em: *Physics Letters A* 268.3 (2000), pp. 158–160.
- [15] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. Em: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

- [16] John Preskill. “Quantum computing in the NISQ era and beyond”. Em: *Quantum* 2 (2018), p. 79.
- [17] Yue Ruan et al. “Quantum algorithm for k-nearest neighbors classification based on the metric of hamming distance”. Em: *International Journal of Theoretical Physics* 56.11 (2017), pp. 3496–3507.
- [18] Jiangyan Yi et al. “Add 2023: the second audio deepfake detection challenge”. Em: *arXiv preprint arXiv:2305.13774* (2023).
- [19] Jiangyan Yi et al. “Audio deepfake detection: A survey”. Em: *arXiv preprint arXiv:2308.14970* (2023).
- [20] Fang Zheng, Guoliang Zhang e Zhanjiang Song. “Comparison of different implementations of MFCC”. Em: *Journal of Computer science and Technology* 16.6 (2001), pp. 582–589.