Maria Eduarda Maciel Pinto

Detecção De Replay Attacks Em Espaço de Dissimilaridade

Curitiba - PR, Brasil

2025

Maria Eduarda Maciel Pinto

Detecção De Replay Attacks Em Espaço de Dissimilaridade

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de mestre em Informática.

Pontifícia Universidade Católica do Paraná - PUCPR

Programa de Pós-Graduação em Informática - PPGIa

Orientador: André Hochuli

Coorientador: Alceu Britto Jr

Curitiba - PR, Brasil

2025

Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Informática

Curitiba, 17 de abril de 2024.

31-2024

DECLARAÇÃO

Declaro para os devidos fins, que **MARIA EDUARDA MACIEL PINTO** defendeu a dissertação de Mestrado intitulada "**Detecção de Replay Attacks Em Espaço de Dissimilaridade"**, na área de concentração Ciência da Computação no dia 08 de dezembro de 2023, no qual foi aprovada.

Declaro ainda, que foram feitas todas as alterações solicitadas pela Banca Examinadora, cumprindo todas as normas de formatação definidas pelo Programa.

Por ser verdade firmo a presente declaração.

Documento assinado digitalmente
gov.br
EMERSON CABRERA PARAISO
Data: 17/04/2024 14:33:13-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Emerson Cabrera Paraiso
Coordenador do Programa de Pós-Graduação em Informática

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central
Gisele Alves – CRB 9/1578

P659d
2025

Pinto, Maria Eduarda Maciel
Detecção de replay attacks em espaço de dissimilaridade / Maria Eduarda Maciel Pinto ; orientador : André Hochuli ; coorientador : Alceu Britto Jr. – 2025.
49 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2025
Bibliografia: f. 46-49

1. Reconhecimento automático da voz. 2. Aprendizado profunto. I. Hochuli, André Gustavo. II. Britto Júnior, Alceu de Souza. III. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. IV. Título.

CDD. 20. ed. – 004

Agradecimentos

Em primeiro lugar, gostaria de agradecer a minha família, em especial minha mãe Luciane, por todo o apoio, cuidado, amor, dedicação e incentivo que me deu durante toda a minha vida. Também agradeço minha irmã, Ana Luíza, por todo o apoio e paciência que teve comigo durante esse tempo de desenvolvimento do mestrado.

Também gostaria de agradecer aos meus amigos, especialmente a Marina de Lara, por todo o suporte, carinho e conselhos sobre o mundo acadêmico e ao Felipe e João Victor pela ajuda e companhia no laboratório. Além disso, um muitíssimo obrigada a todos os meus outros amigos que, durante esses 3 anos, me ouviram pacientemente falar sobre o projeto, me incentivaram e torceram por mim, sendo eles: Isadora, Gabriele, Renata, Camila, David, Katriny e Bianca, sou muito grata pela amizade de todos.

Por fim, gostaria de agradecer aos professores da PUCPR que me acompanharam nessa jornada, especialmente meu orientador André Hochuli e meu coorientador Alceu Britto Jr por terem permitido o desenvolvimento desse projeto, e a professora Rafaela Otemaier pelo incentivo e apoio desde a Graduação até o fim do Mestrado.

Resumo

Os sistemas de reconhecimento de voz (sistemas ASV) tem sido cada vez mais utilizados em diversos dispositivos e tecnologias como uma alternativa de segurança biométrica. No entanto, esses sistemas ainda possuem muitas falhas e são passíveis de sofrer ataques de spoofing, como são denominados os procedimentos de falsificação de voz. Um dos ataques mais complexos de se identificar, porém o mais simples de ser desempenhado é o Replay Attack. Nele, o atacante utiliza dispositivos de gravação para captar a voz da vítima e posteriormente a reproduz para desbloquear o acesso a dispositivos. Nos últimos 6 anos, foram desenvolvidos estudos como tentativa de reduzir o risco de ataques, de forma a desenvolver um sistema que consiga identificar áudios fraudulentos. No entanto, as abordagens atuais são bastante repetitivas, baseadas em alternar as frequências de áudios e usá-las de input de redes neurais conhecidas, como CNNs tradicionais, LSTMs, RNNs etc. Uma técnica que tem obtido bons resultados na inteligência artificial, especialmente no reconhecimento de imagens, é a da Dissimilaridade, que visa calcular a distância entre os objetos de amostra e objetos de referência como forma de definir à qual classe um objeto pertence. Ela se divide em duas abordagens: Vetores de Dissimilaridade e Espaço de Dissimilaridade. Diante disso, surge a oportunidade de aplicar a Dissimilaridade dentro do contexto dos Replay Attacks e verificar qual seu impacto dentro do problema. Portanto, nesse estudo, foi proposta uma abordagem de sistema ASV para identificação de Replay Attacks utilizando a abordagem de Dissimilaridade. No método proposto, são utilizadas diferentes arquiteturas de redes neurais como extratores de características, em seguida é realizada aplicação do algoritmo k-means para seleção de objetos de referência, que serão utilizados para o cálculo de dissimilaridade. Posteriormente, os resultados são utilizados como input para classificadores rasos. O melhor resultado obtido nesse trabalho foi um EER de 0,1211 na técnica de Vetores de Dissimilaridade, utilizando ResNet e um classificador SVM.

Palavras-chave: anti-spoofing, audio fake, ataque de replay, deep learning, dissimilaridade, reconhecimento de voz.

Abstract

Speech Recognition Systems (ASR systems) have been increasingly used in various devices and technologies as an alternative biometric security measure. However, these systems still have many flaws and are susceptible to spoofing attacks, which are procedures for voice falsification. One of the most complex yet simplest attacks to execute is the Replay Attack. In this attack, the attacker uses recording devices to capture the victim's voice and later replays it to unlock access to devices. Over the past six years, studies have been developed in an attempt to reduce the risk of attacks, aiming to create a system capable of identifying fraudulent audio. Nevertheless, current approaches are quite repetitive, relying on alternating audio frequencies and using them as input for well-known neural networks such as traditional CNNs, LSTMs, RNNs, etc. A technique that has shown promising results in artificial intelligence, especially in image recognition, is Dissimilarity, which calculates the distance between sample objects and reference objects to determine the class to which an object belongs. It is divided into two approaches: Dissimilarity Vectors and Dissimilarity Space. In light of this, there is an opportunity to apply Dissimilarity within the context of Replay Attacks and assess its impact on the problem. Therefore, this study proposes an ASV system approach for identifying Replay Attacks using the Dissimilarity approach. In the proposed method, different neural network architectures are used as feature extractors. Subsequently, the k-means algorithm is applied to select reference objects, which will be used for dissimilarity calculation. Later, the results are used as input for shallow classifiers. The best result achieved in this work was an Equal Error Rate (EER) of 0.1211 using the Dissimilarity Vectors technique, employing ResNet and an SVM classifier. **Keywords**: spoofing detection, fake audio, replay attack, deep learning, dissimilarity, speech recognition.

Lista de ilustrações

Figura 1 – Exemplo de *Replay Attack* 17
Figura 2 – Fonte: Diferença entre um áudio *bonafide* e *spoofed* 18
Figura 3 – Frequência de um áudio em diferentes ambientes 19
Figura 4 – Espectrogramas 21
Figura 5 – Escala Mel 21
Figura 6 – MFCCs 22
Figura 7 – Arquitetura de uma rede neural artificial (ANN) 22
Figura 8 – Arquitetura de uma rede neural convolucional (CNN) 23
Figura 9 – Abordagens de Dissimilaridade 25
Figura 10 – Distribuição dos dados com e sem dissimilaridade 25
Figura 11 – Matrizes de dis(similaridade) criadas usando objetos de cinco classes diferentes 26
Figura 12 – Vetores de Dissimilaridade 27
Figura 13 – Resultado Vetores de Dissimilaridade 27
Figura 14 – EER - Curva ROC 28
Figura 15 – ASV setup de gravação 30
Figura 16 – Fluxo de etapas dos experimentos 34
Figura 17 – Arquitetura VGG 16 36
Figura 18 – Arquitetura InceptionV3 36
Figura 19 – Arquitetura ResNet50 37
Figura 20 – Distribuição dos áudios no espaço 37
Figura 21 – Centróides 38
Figura 22 – Seleção de referências 38
Figura 23 – Fluxo Vetores de Dissimilaridade 39
Figura 24 – Fluxo Espaço de Dissimilaridade 40
Figura 25 – a) métricas b) matriz de confusão 41
Figura 26 – a) métricas b) matriz de confusão 43
Figura 27 – a) métricas b) matriz de confusão 44

Lista de tabelas

Tabela 1 – Estado da Arte 33
Tabela 2 – Base 35
Tabela 3 – Resultados1 42
Tabela 4 – Resultados2 43
Tabela 5 – Resultados3 44

Lista de abreviaturas e siglas

ANN	Artificial Neural Network
ASV	Automatic Speaker Verification
ASVspoof	Automatic Speaker Verification Spoofing and Countermeasure
BLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
CQCC-GMM	Constant-Q Cepstral Coefficient - Gaussian Mixture Model
ConvRBM	Convolutional Restrict Boltzmann Machine
DCF	Detection Cost Function
DFT	Discrete Fourier Transform
DNN	Deep Neural Network
EER	Equal Error Rate
ESA	Energy Separation Algorithm
FDNN	Feature-based Deep Neural Network
HFCC	High Frequency Cepstral Coefficient
LPCC	Linear Prediction Cepstral Coefficient
MFCC	Mel Frequency Cepstral Coefficient
NIST	National Institute of Standards and Technology
ResNet	Residual Neural Network
RPS	Relative Phase Shift
SMOTE	Synthetic Minority Oversampling Technique
SS	Speech Synthesis
STFT	Short-Term Fourier Transform
SVM	Support Vector Machine

t-DCF	Tandem Detection Cost Function
VC	Voice Conversion
VESA-IFCC	Variable Length Energy Separation Algorithm

Sumário

1 INTRODUÇÃO 13
1.1 Objetivos 14
1.1.1 Objetivo geral 14
1.1.2 Objetivos específicos 15
1.2 Contribuições 15
1.3 Organização do Documento 15

2 FUNDAMENTAÇÃO TEÓRICA 16
2.1 Sistemas ASV e Audio Spoofing 16
2.1.1 Ataques de *Replay* 17
2.2 Propriedades de áudio 20
2.2.1 Mel-frequency Cepstral Coefficients (MFCCs) 20
2.3 Redes Neurais e Extração de Características 21
2.4 Espaço de Dissimilaridade 24
2.5 Métricas de Avaliação em Replay Attack 26

3 ESTADO DA ARTE 29
3.1 Bases de Dados 29
3.2 Detecção de Replay Attacks 30

4 METODOLOGIA 34
4.1 Experimentos 34
4.1.1 Análise dos dados 35
4.1.2 Conversão dos áudios 35
4.1.3 Extração de características 36
4.1.4 Seleção de Referências 37
4.1.5 Cálculo de Dissimilaridade 38
4.1.6 Classificação 40

5 RESULTADOS 41
5.1 Experimentos 41
5.1.1 Experimentos em Feature Space 41
5.1.2 Experimentos com Vetores de Dissimilaridade 42
5.1.3 Experimentos com Espaço de Dissimilaridade 43

6 DISCUSSÕES E CONCLUSÃO 45

7 TRABALHOS FUTUROS 47

REFERÊNCIAS 48

1 Introdução

Ao longo da última década, as tecnologias de biometria avançaram significativamente e estão sendo cada vez mais inseridas no cotidiano (LAVRENTYEVA et al., 2017). Com o crescimento acelerado da quantidade de dados e a grande necessidade de proteger informações valiosas ou confidenciais, muitos sistemas de dispositivos móveis, bancos e assistentes virtuais, passaram a aderir ao uso de soluções biométricas para permissões de acesso. A biometria desempenha um papel imprescindível no regulamento de acesso a recursos e informação, e uma das abordagens consideradas mais confiáveis para conseguir um sistema de segurança adequado é a verificação de voz (RAHULT et al., 2020).

Diante disso, foi dado início ao desenvolvimento de sistemas de verificação automática de voz (ASV), que foram incorporados aos procedimentos de acesso a serviços e informações. Seu principal objetivo é desempenhar uma análise na voz de entrada, identificar a sua veracidade e, com base no resultado, permitir ou não o acesso. Os sistemas ASV são amplamente utilizados em call-centers, unidades de resposta audíveis e também em aplicações móveis. Eles são ativamente comercializados devido à sua confiabilidade, conveniência, baixo custo e segurança fornecida (LAVRENTYEVA et al., 2017).

Entretanto, a ampla utilização dos sistemas ASV traz consigo uma vulnerabilidade significativa, visto que, existe a possibilidade de burlar o reconhecimento de voz por meio da falsificação da fala. Essas técnicas de fraude são denominadas ataques de *spoofing* ou *audio spoofing*. Há diferentes procedimentos que podem ser utilizados para passar pelos sistemas ASV, que são selecionados pelo atacante de acordo com o objetivo e contexto da situação que desejam fraudar. Os ataques são categorizados da seguinte forma (WU et al., 2015): a) *Speech Syntheis* (SS), b) *Impersonation*, c) *Voice Conversion (VC)* e d) *Replay Attack*.

Dentre essas categorias, os ataques de *replay* são os mais fáceis de se implementar, porém são considerados os mais indistinguíveis métodos de *spoof*, já que eles não requerem que o atacante possua nenhum tipo de conhecimento especializado e podem ser performados com aparelhos de consumidores comuns (RAHULT et al., 2020).

Considerando a necessidade de desenvolvimento de mais estudos no campo de sistemas de detecção de *spoofing*, a iniciativa ASVspoof (Automatic Speaker Verification Spoofing and Countermeasures), financiada por um conjunto de instituições, provém uma plataforma para que pesquisadores estudem e comparem sistemas de detecção de spoofing (JUNG et al., 2019).

Os desafios da competição ASVspoof tiveram início em 2015 e, desde então, têm ocorrido a cada dois anos. Ao longo das edições, foram exploradas muitas técnicas para a

detecção de ataques de replay. A primeira etapa necessária para a realização de experimentos com áudios, é o seu pré-processamento e a análise de suas frequências, para que assim torne-se possível extrair informações relevantes. Os estudos desenvolvidos na competição do ASVspoof realizaram experimentos com diferentes frequências de áudio, utilizando principalmente a MFCC (Mel-frequency Cepstral Coefficients) e a CQCC (Constant Q Cepstral Coefficients). Em relação às abordagens de Machine Learning e Deep Learning (aplicadas após o pré-processamento), foram utilizadas diferentes arquiteturas de redes neurais individuais e combinadas. Houve uma grande quantidade de experimentos com CNNs tradicionais, CNN combinada com LSTM e RNNs.

Muitos destes trabalhos obtiveram resultados satisfatórios, no entanto, a grande maioria deles utiliza as mesmas técnicas, alternando apenas entre os tipos de frequências e/ou as arquiteturas das redes neurais. Portanto, observa-se a ausência de trabalhos que explorem outras abordagens.

Nos últimos anos, uma técnica que tem ganhado espaço nos estudos da inteligência artificial e vem produzindo bons resultados, é a da dissimilaridade. No contexto do reconhecimento de padrões, o uso das abordagens de dissimilaridade é baseado na ideia de que, de alguma forma, os seres humanos utilizam as dissimilaridades existentes entre os padrões para fazer a distinção entre eles (COSTA et al., 2020). Na dissimilaridade, são colhidas amostras das diferentes classes existentes no problema, para que sejam utilizadas como referências. A partir dessas referências, é realizado um cálculo de distância entre elas e cada um dos elementos da base de estudo. De acordo com o resultado desse cálculo, é possível prever à qual classe um elemento pertence, considerando que quanto menor o seu valor de distância resultante, maior é a chance de que o elemento faça parte da mesma classe da referência e vice-versa. No entanto, embora seja uma técnica que esteja trazendo bons resultados em outros contextos, ela ainda não foi utilizada no campo da detecção de áudios fakes.

Diante do fato de que os estudos realizados até o momento têm seguido apenas a mesma linha de experimentos, juntamente com a ainda existente necessidade de evoluir os resultados para chegar a um sistema verdadeiramente eficaz, a empregabilidade de novas técnicas dentro do contexto de audio spoofing, como a dissimilaridade, pode trazer contribuições para os estudos dentro desse tema.

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo deste trabalho é desenvolver uma nova abordagem associando redes neurais e espaço de dissimilaridade para detectar áudios fradulentos gerados a partir de

ataques de replay.

1.1.2 Objetivos específicos

1. Pré-processar os dados da base ASVspoof utilizando a frequência MFCC.

2. Comparar diferentes arquiteturas de redes neurais para a extração e representação de características do problema.

3. Avaliar as duas abordagens existentes dentro da dissimilaridade e métodos de seleção de referências.

4. Comparação de classificadores rasos no contexto da dissimilaridade.

4. Explorar a métrica de EER no contexto do *audio spoofing*.

5. Análise crítica sobre a dissimilaridade quando comparada ao feature space.

1.2 Contribuições

As contribuições deste estudo são:

- Elucidar, por meio de análise crítica, a comparação entre a dissimilaridade e o Feature Space.
- Apresentar uma abordagem de dissimilaridade no contexto de detecção de Replay Attacks.

1.3 Organização do Documento

Essa dissertação está organizada de forma que, o Capítulo 2 traz a explicação dos fundamentos teóricos utilizados como base para este estudo, bem como as principais abordagens utilizadas na resolução do problema; o Capítulo 3 apresenta um levantamento e revisão bibliográfica em relação aos principais estudos desenvolvidos no contexto da temática deste projeto, bem como as bases existentes para a condução de experimentos de detecção de ataques de replay; o Capítulo 4 mostra as abordagens sugeridas pelo presente estudo, bem como os experimentos realizados; O Capítulo 5 apresenta os resultados obtidos, bem como uma discussão sobre eles; O Capítulo 7 apresenta a conclusão e, por fim, o Capítulo 8 sugere possíveis trabalhos futuros relacionados ao tema.

2 Fundamentação Teórica

Neste capítulo serão abordadas as definições em relação ao tema do *audio spoofing* e do espaço de dissimilaridade. Também serão explicadas as métricas de avaliação dos sistemas de detecção de *audio spoofing*.

2.1 Sistemas ASV e Audio Spoofing

A biometria tem sido cada vez mais escolhida como o principal método de segurança de acesso em várias tecnologias da atualidade. Ela é utilizada em sistemas das mais diversas áreas, como desbloqueio de dispositivos móveis, acesso a dados, sistemas e recursos, autorização de realização de transações, monitoramento, etc.

Uma das formas de biometria mais utilizadas no momento é a de verificação de voz. Um estudo mostrou que 90% das pessoas são adeptas a utilizar soluções de biometria de voz no lugar de métodos de autenticação tradicionais (BERANEK, 2013).

O maior empecilho enfrentado pelas técnicas de biometria é a existência da possibilidade de fraude no processo da identificação. O principal fator que permite a execução dessas fraudes é que, muitas vezes, os dados biométricos estão publicamente disponíveis, já que não há como esconder completamente nossas vozes, impressões digitais etc. "Traços biométricos não são segredos", cita (GALBALLY; MARCEL; FIERREZ, 2014).

O problema atualmente não é a questão de se a biometria pode ou não ser manipulada, mas sim a de que, até que ponto os sistemas são robustos em relação a esses tipos de ataques e, no caso de não serem, quais são as possíveis contramedidas para detectá-los (PATIL; KAMBLE, 2018).

O objetivo de um sistema ASV (Automatic Speaker Verification) é determinar ou verificar a identidade da voz de um indivíduo (PATIL; KAMBLE, 2018). A detecção de Spoofing para sistemas ASV tem sido o foco de muitas pesquisas nos últimos anos e, pode ser definida como a tarefa de distinguir entre enunciados genuínos e fraudulentos para proteger os sistemas ASV de ataques de spoofing (YOON; YU, 2021).

Ataques spoofing em sistemas ASV são definidos como um mascaramento de voz, ou seja, fazem com que uma voz se passe por vozes de outros indivíduos com o intuito de conseguir acesso a dados protegidos (EVANS; KINNUNEN; YAMAGISHI, 2013).

Como citado anteriormente, há quatro tipos diferentes de audio spoofing: a) *Speech Syntheis* (SS): técnica de geração de voz utilizando exclusivamente ferramentas computacionais e artificiais, b) *Impersonation*: método de edição de voz na qual o atacante

altera computacionalmente a sua própria voz para parecer com a voz de outra pessoa, c) *Voice Conversion* (VC)": técnica de conversão da voz da vítima para fazer com que sejam ditas frases as quais ela originalmente não falou e d) *Replay Attack*: gravação e posterior reprodução do áudio de uma vítima.

2.1.1 Ataques de *Replay*

Este trabalho irá se delimitar na categoria de *Replay Attacks* ou ataques de replay. Nos *Replay Attacks*, impostores apresentam amostras de voz gravadas de um usuário genuíno a um sistema de verificação (BALAMURALI et al., 2019).

Esse tipo de ataque é um dos mais ameaçadores aos sistemas ASV, já que são os mais complexos de serem detectados. Em contrapartida, eles podem ser implementados com muita facilidade, pois não exigem que o atacante apresente um alto grau de conhecimento em nenhuma área específica e podem ser performados utilizando dispositivos simples, como *smartphones* ou aparelhos gravadores comuns.

Na Figura 1 é apresentado um exemplo de ataque de *replay*. A imagem mostra um indivíduo utilizando a biometria de voz para acessar as funcionalidades de automatização da sua casa, como por exemplo, destrancar a porta. Ao seu lado, há um atacante que, utilizando um *smartphone* comum, faz a gravação do comando de voz do usuário. Por fim, o atacante reproduz a gravação anteriormente captada e dessa forma, consegue desbloquear o acesso à casa.

Figura 1 – Exemplo de *Replay Attack*

Fonte: Autoria Própria

A tarefa de detecção de *replay attacks* é identificar se um dado sinal de voz é gravado a partir de uma reprodução de voz genuína ou se é gravado de maneira intermediária (gravação + dispositivo de playback para reprodução fraudulenta) (PATIL; KAMBLE, 2018).

O sinal de voz genuína $s[n]$ pode ser modelado como uma convolução do fluxo de ar glotal $p[n]$ e a resposta ao impulso de trato vocal $h[n]$, como representado na Equação 2.1 (QUATIERI, 2006).

$$s[n] = p[n] * h[n] \tag{2.1}$$

Enquanto que, o sinal de voz de replay $r[n]$ pode ser modelado como a convolução do sinal de voz genuína $s[n]$ e a resposta ao impulso dos dispositivos intermediários $N[n]$ (dispositivos de gravação e reprodução) juntamente com a propagação do ambiente e é dado como Equação 2.2:

$$r[n] = s[n] * N[n] \tag{2.2}$$

Onde $N[n]$ são os componentes extras convolucionados, ou seja, a combinação das respostas de impulso do aparelho de gravação $hmic[n]$, ambiente de gravação $a[n]$, dispositivo de reprodução (autofalante) $hspk[n]$, e ambiente de reprodução $b[n]$, o que pode ser representado na Equação 2.3:

$$N[n] = hmic[n] * a[n] * hspk[n] * b[n] \tag{2.3}$$

A tarefa desafiadora no contexto dos ataques de *replay*, é conseguir detectar as características acústicas, já que a diferença do sinal de voz natural e da reproduzida é quase imperceptível. O sinal de fala gravado com o dispositivo de reprodução contém a convolução e a adição das distorções dos dispositivos intermediários.

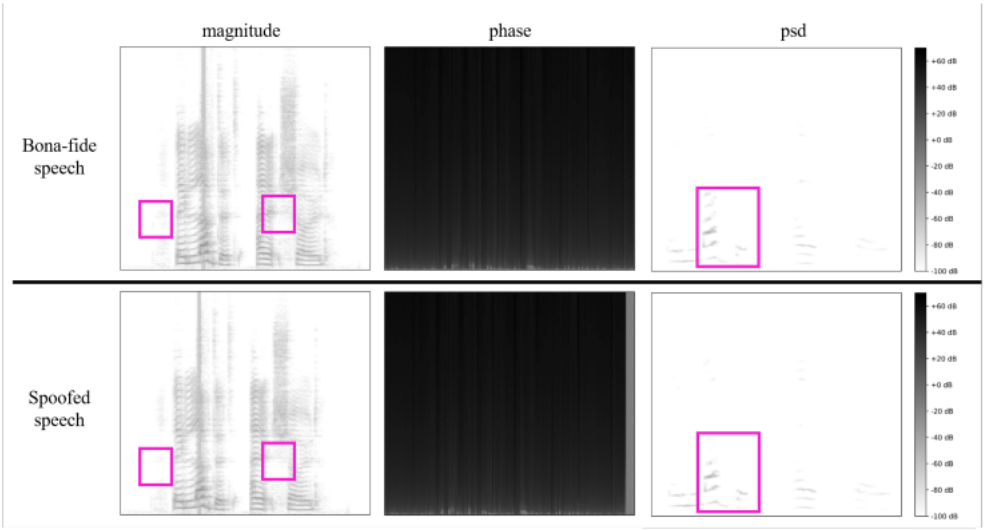


Figura 2 – Fonte: Diferença entre um áudio *bonafide* e *spoofed* (JUNG et al., 2019)

A parte mais crucial da detecção do *replay attack* é o processo de extração de características. Para obter as informações discriminatórias entre os sinais de fala natural

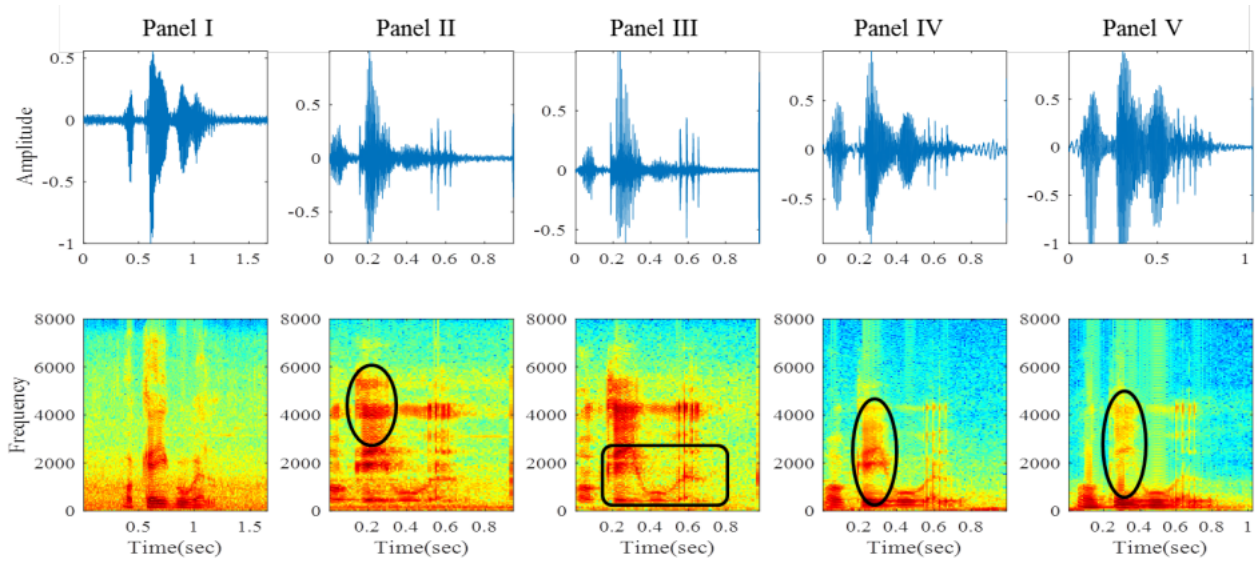
e reproduzida, o foco no processo da extração de características deve representar as características espectrais do dispositivo intermediário.

A Equação 2.4 representa o termo de convolução que se transforma na relação de adição no domínio cepstral (RAFI; MURTY; NAYAK, 2017):

$$r = s + h \tag{2.4}$$

onde r, s e h representam vetores cepstrais de replay, sinal de voz natural e resposta de impulso do dispositivo, respectivamente. As características obtidas do vetor h pode ser usada de forma a subtrair o sinal natural de fala do sinal de fala de replay (RAFI; MURTY; NAYAK, 2017).

Na Figura 3, tem-se um exemplo de um áudio reproduzido em diferentes ambientes, nela pode-se observar que a troca de ambiente provoca alterações significativas nos sinais do áudio.



Fonte: (PATIL; KAMBLE, 2018)

Figura 3 – Frequência de um áudio em diferentes ambientes

O desenvolvimento de uma contramedida para detecção de *audio spoofing* habitualmente consiste em duas etapas: A primeira é definida pela extração de características, além de pré e pós processar os sinais de áudio. A segunda parte é a criação de um modelo que, utilizando as características extraídas dos áudios, seja treinado e determine se o áudio é genuíno ou fraudulento (BALAMURALI et al., 2019).

2.2 Propriedades de áudio

Para analisar um áudio, é necessário visualizá-lo de uma maneira na qual suas características sejam destacadas, o que pode ser feito por meio da geração de waveforms, espectrogramas, diagramas de coeficientes cepstrais etc. A seguir serão abordadas algumas das formas mais utilizadas de tratamento dos áudios para o desenvolvimento de sistemas de reoconhecimento de voz.

O pré-processamento de um áudio consiste em duas etapas: pré-enfatizar e realçar o arquivo de áudio. A etapa de pré-enfatizar visa impulsionar a quantidade de energia presente nas frequências mais altas. As gravações são então realçadas através da remoção de ruídos utilizando um método de subtração de ruído espectral, no qual uma estimativa do espectro de ruído é subtraída do espectro de energia do áudio e as diferenças negativas são setadas como zero (BALAMURALI et al., 2019).

Os espectrogramas apresentam uma representação visual do espectro de frequências presentes em um sinal de áudio ao longo do tempo. Eles são bastante utilizados como recurso em imagens e áudios para sistemas de classificação. Especialmente na esfera da fala, eles têm apresentado uma melhora significativa no desempenho do sistema de classificação quando associados com outros *frameworks* de áudio (MFCCs e LPCCs por exemplo). O espectrogramas podem ser gerados de muitas maneiras, porém a mais utilizada é a partir da Transformada de Fourier de Curta Duração (STFT), que fornece amplitude e informações de fase para análise. Na 4 pode-se observar exemplos de espectrogramas mostrando amplitude e frequências de um áudio.

2.2.1 Mel-frequency Cepstral Coefficients (MFCCs)

Os coeficientes MFCCs focam nos aspectos perceptualmente relevantes do espectro da fala e são um dos mais utilizados na área de reconhecimento de voz. O processo de extração dos MFCCs segue o seguinte fluxo: Primeiro é aplicada a Transformação Discreta de Fourier (DFT) no áudio de entrada para obter seu espectro. Esse espectro resultante irá indicar a quantidade de energia presente nas várias bandas de frequência do áudio.

Como a audição humana é muito sensível para as frequências mais baixas, porém menos capazes de distinguir sons de alta frequência, usa-se uma representação que considere esse fator: a mel-spectrum. Para calcular essa representação, o espectro é modificado para uma escala mel-spectrum através do uso de um banco de filtros triangulares. Uma vez que o espectro é modificado, o logaritmo da energia presente nas várias regiões do espectro da fala são estimados. Por fim, esse espectro é transformado novamente ao domínio de tempo, obtendo-se então os MFCCs (RABINER; JUANG, 1993).

Nas figuras 5 e 6 são exemplificados os espectrogramas de um áudio convertido em Escala Mel e MFCC.

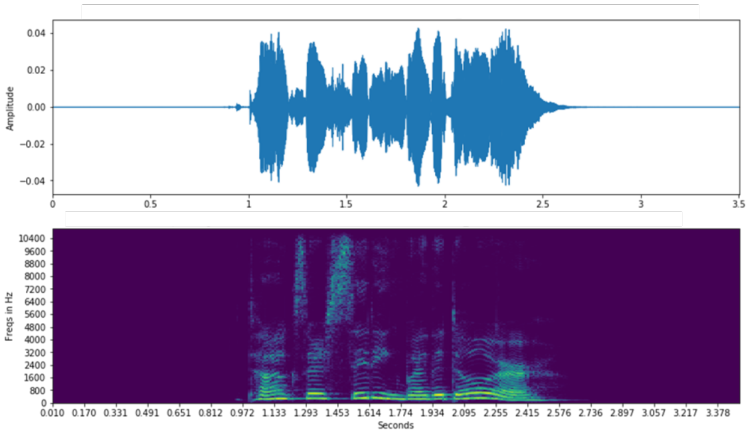


Figura 4 – Espectrogramas

Fonte: Autoria Própria

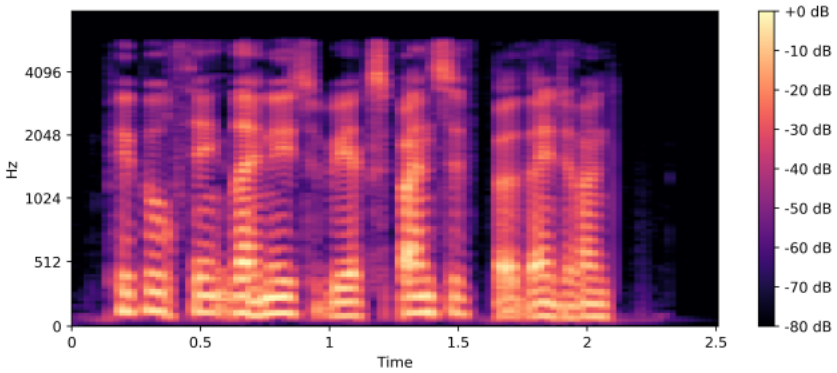


Figura 5 – Escala Mel

Fonte: (RAHULT et al., 2020)

2.3 Redes Neurais e Extração de Características

Redes neurais podem ser definidas como modelos quantitativos que conectam adaptativamente dados de entrada e saídas em um processo de aprendizagem análogo ao utilizado pelo cérebro humano. As redes consistem em nós, que são chamados de neurônios rotulados. Eles são conectados por uma série de regras e pesos. Os nós codificam características e elas aparecem em camadas, sendo a primeira camada a de entrada e a última a de saída. Os dados em análise são processados através das diferentes camadas compondo a rede, com o aprendizado ocorrendo através da alteração dos pesos que conectam os nós (ABDI; VALENTIN; EDELMAN, 1999).

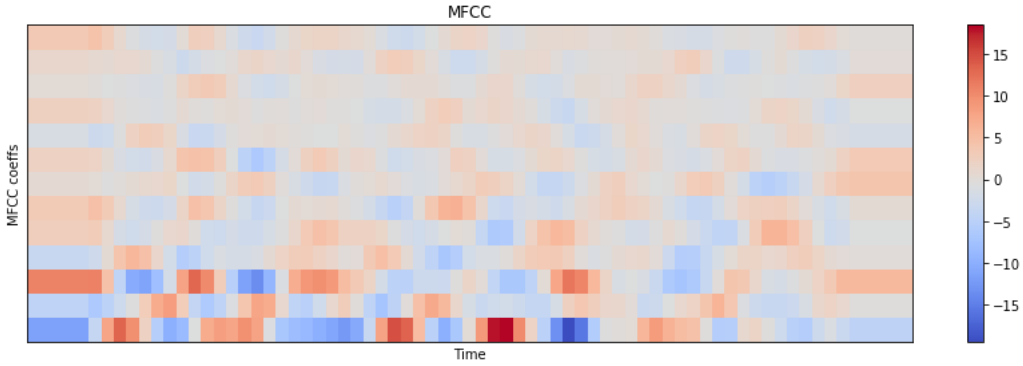


Figura 6 – MFCCs

Fonte: Autoria Própria

Uma rede neural artificial (ANN - *Artificial Neural Network*), pode ser estruturada como representado na Figura 7. Primeiramente são carregados os dados de *input*, normalmente no formato de um vetor multidimensional na camada de input, que serão então distribuídos para as camadas ocultas, também chamadas de *hidden layers*. As camadas ocultas vão então fazer decisões a partir da camada anterior e, a partir dos pesos, avaliar como as mudanças estocásticas ajudariam ou prejudicariam o resultado final. Este processo é chamado de aprendizado. A existência de múltiplas camadas intermediárias empilhadas uma sobre a outra é comumente chamada de *deep learning* (O'SHEA; NASH, 2015).

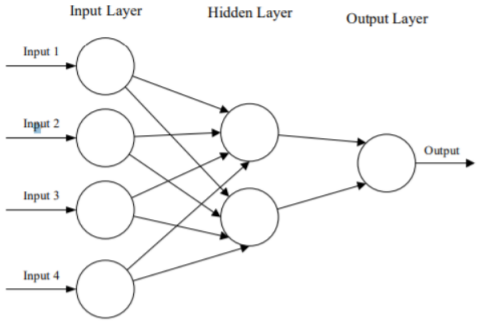


Figura 7 – Arquitetura de uma rede neural artificial (ANN)

Fonte: (O'SHEA; NASH, 2015)

Devido às habilidades das redes neurais artificiais de aprenderem a partir de dados estruturados e de demonstrar alta performance, elas estão sendo cada vez mais utilizadas em aplicações práticas e processos críticos de decisão, tais como técnicas de descoberta de conhecimento, direção autônoma de carros ou análise de imagens médicas (ALBER et al., 2018).

Redes neurais convolucionais (CNNs) são bastante análogas às ANNs. A diferença notável entre as CNNs e as ANNs tradicionais, é que as CNNs são primordialmente utiliza-

das no contexto do reconhecimento de padrões em imagens. Isso permite a codificação de características específicas das imagens na arquitetura, bem como a redução dos parâmetros requeridos para configurar o modelo (O'SHEA; NASH, 2015).

As CNNs são consistidas em três tipos de camadas, que são as camadas convolucionais, as camadas de *pooling* e as camadas *fully-connected*.

- Camadas convolucionais: Esse tipo de camada tem o objetivo de aprender as representações de características dos inputs. Ela é composta por vários *kernels* convolucionais, que são utilizados para computar diferentes mapas de características, funcionando como filtros.
- Camadas de *pooling*: Essa categoria de camada visa atingir um *shift-invariance* (invariância de deslocamento) por meio da redução da resolução dos mapas de características. Elas são inseridas normalmente entre duas camadas convolucionais.
- Camadas *fully-connected*: Camadas deste tipo possuem a intenção de performar um raciocínio de alto nível. Elas pegam todos os neurônios da camada anterior e conectam eles em cada um dos neurônios da camada atual, dessa forma gerando uma semântica de informação global.

Quando todas essas camadas são empilhadas, uma arquitetura de CNN se forma. Um exemplo de arquitetura de CNN pode ser observado na Figura 8.

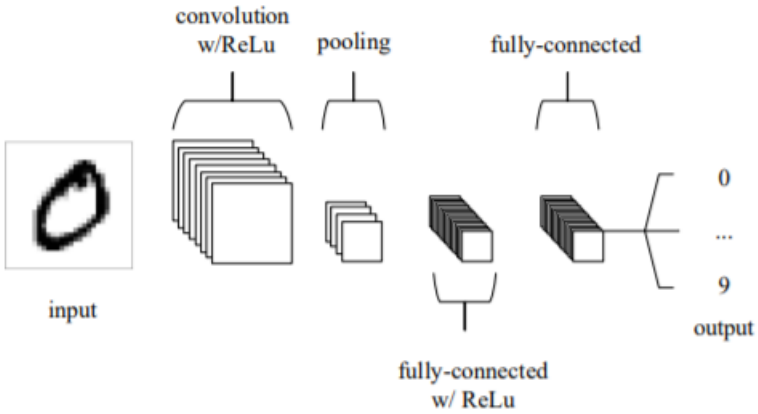


Figura 8 – Arquitetura de uma rede neural convolucional (CNN)

Fonte: (O'SHEA; NASH, 2015)

Existem várias arquiteturas oriundas de CNNs pré-criadas na literatura, que são bastante reconhecidas e obtêm bons resultados em inúmeros contextos. Nesse estudo, foram escolhidas duas dessas arquiteturas para os experimentos, sendo elas: InceptionV3 e VGG 16.

Outra categoria de redes que tem trazido resultados bastante satisfatórios, são as Redes Neurais Residuais (ResNet). Elas foram desenvolvidas como sendo uma evolução das CNNs tradicionais, já que solucionam o problema de limitação que elas possuem em relação à adição de múltiplas camadas para solucionar problemas complexos. A ResNet consiste na ideia de blocos residuais, cujo funcionamento se baseia em pular conexões, dessa forma conseguindo conectar camadas que não eram consecutivas dentro da rede. Essa técnica consegue solucionar o problema de desaparecimento de gradiente e melhorar a performance do modelo (AMELIO et al., 2023).

Para este trabalho, também foi utilizada uma rede neural residual, a ResNet50.

2.4 Espaço de Dissimilaridade

No contexto do reconhecimento de padrões, o uso das abordagens de dissimilaridade é baseado na ideia de que, de alguma forma, os seres humanos utilizam as dissimilaridades existentes entre os padrões para fazer a distinção entre eles (COSTA et al., 2020).

Segundo Pekalska 2002, as dissimilaridades podem ser encaradas como uma ponte que conecta a percepção e o conhecimento de alto nível, desempenhando um papel importante na forma como o ser humano realiza o reconhecimento e a categorização.

O valor de dissimilaridade captura a noção de proximidade entre dois objetos, que pode ser interpretada como uma distância em um espaço adequado, ou ser usada para construir novos espaços (DUIN et al., 2010).

Essa abordagem teve como inspiração a observação de que, a atenção do ser humano ao analisar objetos, capta primeiro as diferenças entre eles e apenas depois considera a descrição geral, de características e modelos de cada um (DUIN; PEKALSKA, 2005).

Considerando que a maioria das técnicas utilizadas na inteligência artificial possui como base o modelo de pensamento e aprendizado dos seres humanos, alguns estudos decidiram explorar o conceito da dissimilaridade no contexto da computação.

A ênfase dada nos últimos estudos sobre dissimilaridade, tem sido a construção de espaços vetoriais que sejam adequados para treinar classificadores, utilizando recursos de estatística, machine learning e reconhecimento de padrões (DUIN; PęKALSKA, 2012).

Dentro do contexto da dissimilaridade, há duas abordagens diferentes: Espaço de Dissimilaridade e Vetores de Dissimilaridade, como mostra a figura 9.

O Espaço de Dissimilaridade é um espaço vetorial no qual as dimensões são definidas por vetores de dissimilaridade que medem a dissimilaridade (distância) entre exemplos e objetos individuais de um conjunto R. Dessa forma, a representação de dissimilaridade $D(X, R)$ é tratada como um mapeamento dependente de dados de um conjunto inicial de objetos X para um espaço de dissimilaridade, equipado com o tradicional produto

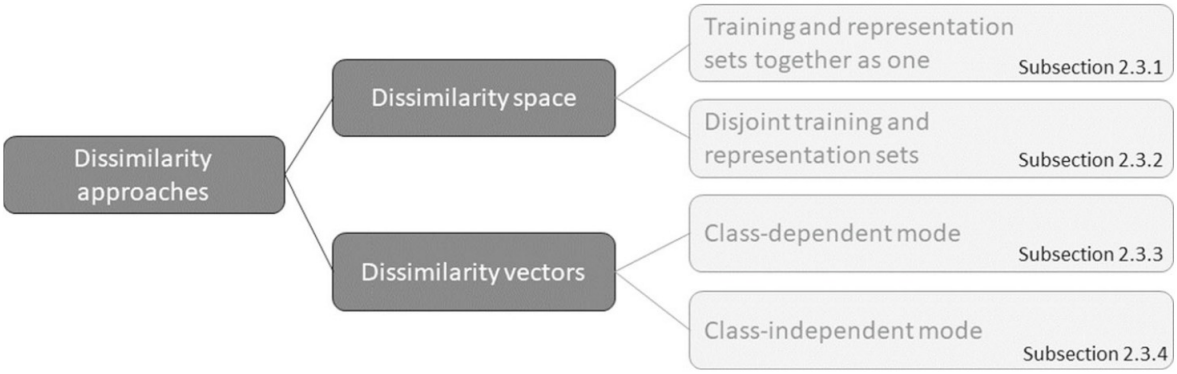


Figura 9 – Abordagens de Dissimilaridade

Fonte: (COSTA et al., 2020)

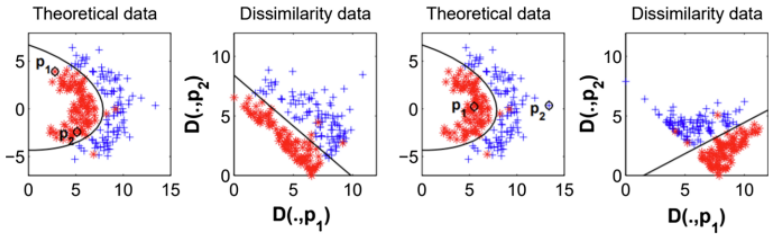


Figura 10 – Distribuição dos dados com e sem dissimilaridade

Fonte: (COSTA et al., 2020)

interno e a métrica euclidiana. O conjunto de representação pode ser escolhido como sendo o conjunto completo de treinamento T, um conjunto cuidadosamente selecionado, um conjunto de protótipos construídos ou ainda um conjunto de objetos arbitrariamente rotulados ou não rotulados (DUIN; PęKALSKA, 2012).

Para melhor ilustrar o conceito e a criação de uma matriz de dissimilaridade, (COSTA et al., 2020) descreveram um exemplo visual, utilizando formas geométricas para representar 5 classes diferentes. A Figura 11 ilustra uma matriz quadrada de dissimilaridade criada a partir de 5 objetos que pertencem a diferentes classes de formas geométricas. Na figura 11(a) e 11(b), é possível observar a sobreposição dos pares de objetos considerando todas as combinações possíveis, no formato de uma matriz de dissimilaridade. Na figura 11(c) pode-se ver a dissimilaridade entre cada par de objetos e na figura 11(d) a similaridade.

Já a abordagem de Vetores de Dissimilaridade, consiste em vetores calculados a partir da diferença entre cada feature de dois vetores de características, que posteriormente são classificados como pertencentes ou não à uma classe (COSTA et al., 2020). Nas figuras 12 e 13, é possível observar o funcionamento do processo de cálculo de vetores de dissimilaridade. Na primeira, mostra-se cada vetor de treino, referências e teste e os valores de features que os compõe. Já na segunda imagem, observa-se o resultado da distância

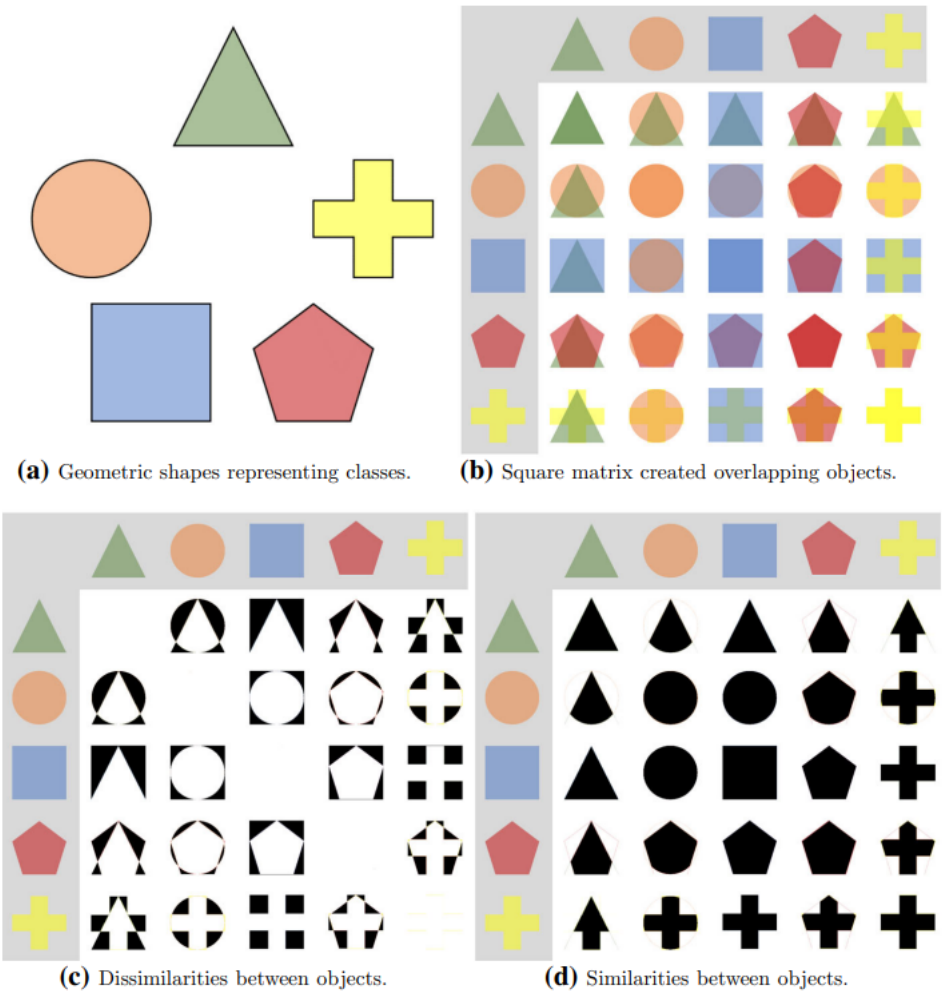


Figura 11 – Matrizes de dis(similaridade) criadas usando objetos de cinco classes diferentes

Fonte: (COSTA et al., 2020)

entre cada feature de cada vetor de treino e teste em relação aos vetores de referência.

2.5 Métricas de Avaliação em Replay Attack

Além das métricas de avaliação mais tradicionais presentes nos trabalhos do campo da Inteligência Artificial, como accuracy e matriz de confusão, no caso de sistemas anti-spoofing, existem algumas métricas mais específicas que podem ser melhor aplicadas para a avaliação desse tipo de sistema.

Na competição do ASVspoof, bem como em trabalhos externos à ela, a métrica utilizada para a mensuração dos resultados foi a EER - Equal Error Rate. Essa métrica é uma estatística normalmente utilizada para medir a performance no contexto da biometria. Ela pode ser definida como o ponto de localização de uma curva ROC, como mostra a Figura 14, na qual a taxa de falsos positivos e falsos negativos são equivalentes. Nessa

Sample	Features				
	F1	F2	F3	F4	F5
Train					
Tr1	1.60	1.56	1.57	1.27	0.52
Tr2	1.29	1.65	1.20	1.41	1.29
Tr3	0.88	1.12	0.69	1.36	0.54
Tr4	4.06	4.48	4.50	2.72	3.57
Tr5	2.41	2.50	3.49	4.88	2.81
Tr6	3.83	3.30	4.51	2.52	3.53
Tr7	4.76	5.22	5.51	4.88	4.85
Tr8	5.12	5.32	5.29	5.08	4.81
Tr9	5.05	5.14	4.82	4.76	5.11
Reference					
Re1	2.10	1.96	1.96	2.28	2.06
Re2	2.04	2.32	1.84	2.14	2.17
Re3	2.23	2.07	1.94	2.00	1.95
Re4	5.14	5.52	4.87	5.04	4.98
Re5	4.61	4.91	4.64	5.17	4.82
Re6	5.28	4.79	4.91	4.95	5.22
Test					
Te1	1.95	2.04	1.77	1.77	2.02
Te2	5.02	4.89	5.06	4.88	5.10

Figura 12 – Vetores de Dissimilaridade

Fonte: (COSTA et al., 2020)

Combined samples	Features				
	F1	F2	F3	F4	F5
Positive samples					
(Te1, Re1)	0.15	0.08	0.19	0.51	0.04
(Te1, Re2)	0.09	0.27	0.07	0.37	0.15
(Te1, Re3)	0.27	0.03	0.17	0.23	0.07
(Te2, Re4)	0.12	0.63	0.19	0.16	0.11
(Te2, Re5)	0.41	0.02	0.42	0.29	0.28
(Te2, Re6)	0.26	0.10	0.15	0.07	0.12
Negative samples					
(Te1, Re4)	3.19	3.47	3.10	3.27	2.96
(Te1, Re5)	2.66	2.87	2.87	3.40	2.80
(Te1, Re6)	3.32	2.75	3.14	3.18	3.20
(Te2, Re1)	2.92	2.93	3.10	2.60	3.04
(Te2, Re2)	2.98	2.57	3.22	2.74	2.93
(Te2, Re3)	2.79	2.82	3.12	2.88	3.15

Figura 13 – Resultado Vetores de Dissimilaridade

Fonte: (COSTA et al., 2020)

métrica, quanto menor for o valor resultante, maior é a acurácia do sistema. A métrica é calculada a partir da Equação 2.5, onde FPR(False Positive Rate) corresponde à taxa de falsos positivos e FNR(False Negative Rate) a taxa de falsos negativos.

$$EER = \frac{FPR + FNR}{2} \tag{2.5}$$

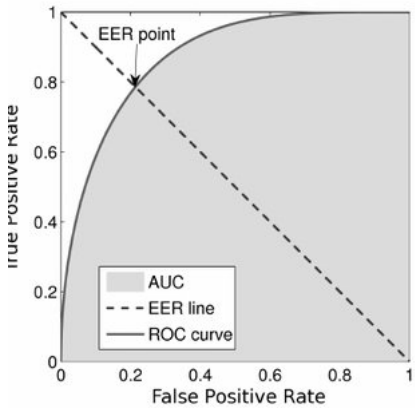


Figura 14 – EER - Curva ROC

Para calcular as taxas de falsos positivos (FPR) e de falsos negativos FNR, são utilizados os resultados obtidos pela matriz de confusão, que exibe os erros e acertos da clssificação. O cálculo do (FPR) pode ser observado na Equação 2.6, onde (FP) corresponde ao número de falsos positivos e (TN) ao número de verdadeiros negativos. No caso do FNR, o cálculo é definido como mostra a Equação 2.7, onde (FN) é o número de falsos negativos e (TP) o de verdadeiros positivos.

$$FPR = \frac{FP}{FP + TP} \tag{2.6}$$

$$FNR = \frac{FN}{FN + TP} \tag{2.7}$$

Além da métrica EER, também foram utilizadas para fins de suporte, as métricas *accuracy* e *f1-score*.

3 Estado da Arte

Nesse capítulo são brevemente descritos alguns dos principais estudos desenvolvidos recentemente no âmbito da detecção de ataques de replay, bem como as bases de dados de áudio nesse contexto existentes na literatura.

3.1 Bases de Dados

Atualmente existem duas bases principais que focam no contexto dos ataques de replay, sendo elas a ASVspoof (ERGUNAY et al., 2015) e a ASVspoof 2019 Challenge (YAMAGISHI et al., 2019).

A. Base AVspoof: Essa é a primeira base a ter introduzido a possibilidade de trabalhar com os ataques de *replay*, já que as anteriores contemplavam somente outros tipos de ataques de *spoofing*. Ela foi utilizada no *BTAS 2016 Challenge* e é composta por 44 indivíduos, sendo 31 homens e 13 mulheres, tendo cada um deles gravado sessões em diferentes condições de ambiente e dispositivos. Os participantes foram gravados lendo sentenças, pronunciando frases pré definidas e também falando livremente sobre algum tópico.

B. ASVspoof 2019 Challenge Database: A segunda base é a ASVspoof 2019, do ASVspoof Challenge (YAMAGISHI et al., 2019). Sua primeira versão surgiu em 2015 e desde então, passou por algumas atualizações até chegar a de 2019. Ela é dividida em 3 categorias: *Speech Synthesis*, *Physical Access* e *Logical Access*. Para o propósito de detecção de ataques de *replay*, a parte da base a ser utilizada é a de *Physical Access*. A categoria de *Physical Access* divide-se em três partes:

- *Training Set* - 54.000 áudios
- *Development Set* - 29.700 áudios
- *Evaluation Set* - 134.730 áudios

As gravações corresponem às vozes de 20 indivíduos diferentes, sendo 12 mulheres e 8 homens, pronunciando sentenças aleatórias e com alternância entre 27 acústicas diferentes, incluindo 3 tamanhos de espaço físico, 3 níveis de reverberação e 3 distâncias diferentes entre o autofalante e o microfone do sistema ASV. Também são feitas alternâncias entre 9 configurações de *setup* dos ataques de replay, compreendendo 3 distâncias diferentes entre o atacante e o atacado e 3 dispositivos de qualidades diferentes de reprodução.

Na Figura 15, pode-se observar como foi estruturado o *setup* de simulação de *replay attacks* para a gravação dos áudios. As tentativas de acesso ao sistema ASV com áudios genuínos são caracterizadas pela propagação do áudio, juntamente com uma configuração acústica definida por um espaço físico (sala) de tamanho S, com uma reverberação R e a uma distância Ds entre o indivíduo falante e o microfone. Já as tentativas de acesso com áudios fraudulentos são caracterizadas pela propagação nas mesmas configurações acústicas (espaço físico (sala) de tamanho S, com uma reverberação R e a uma distância Ds entre o indivíduo falante e o microfone), porém entre o indivíduo falante e o atacante que captura uma cópia do áudio genuíno a uma distância Da do indivíduo falante. As gravações são então reproduzidas usando um autofalante de qualidade Q à mesma distância do microfone em relação ao indivíduo falante nas tentativas com áudios genuínos.

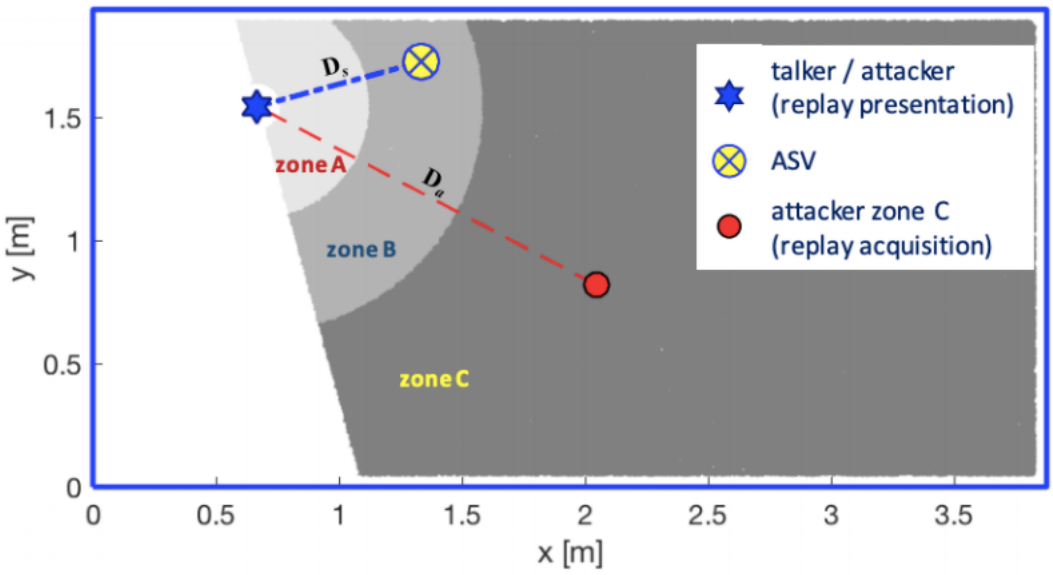


Figura 15 – ASV setup de gravação

Fonte: (YAMAGISHI et al., 2019)

3.2 Detecção de Replay Attacks

Em 2017, (WITKOWSKI et al., 2017) desenvolveram um sistema de detecção de *replay attacks* para o *ASVspoof Challenge 2017*, utilizando como base uma abordagem que detectava quando os sinais de áudio analisados haviam passado por múltiplas conversões de analógico para digital e mostrou que, a maioria das características que possibilitavam essa detecção, estavam presentes nas bandas de alta frequência das gravações reproduzidas. O estudo conseguiu reduzir em 30% a taxa de erro em comparação ao sistema de *baseline* fornecido pelo desafio.

Também em 2017, (CAI et al., 2017) propuseram a avaliação de diferentes técnicas para a construção de uma ferramenta de detecção de ataques de replay. Nessa abordagem, foi utilizada como base a representação de coeficientes espectrais CQCC-GMM, que serviram de input para diferentes tipos de redes neurais, como CNN, FDNN e BLSTM. Além disso, também foram utilizadas estratégias de *Data Augmentation*. O resultado obtido foi uma taxa de erro de 16,39%.

(PATIL et al., 2017) em 2017 exploraram uma abordagem para detectar *replay attacks* baseada na da utilização de um algoritmo de separação de energia ESA associado às representações espectrais VESA-IFCC e CQCC. Foi obtida uma taxa de erro de 18,33%.

Em 2017, (NAGARSHETH et al., 2017) construíram uma abordagem para detectar replay attacks utilizando uma rede DNN que recebeu como input representações de áudio com características espectrais de alto nível (HFCC). Dessa maneira, conseguiram investigar mais discrepâncias entre os dados da base. O resultado foi uma taxa de erro de 11,5%.

Em 2018, (PATIL; KAMBLE, 2018) propuseram outra abordagem. Sendo essa a utilização de características baseadas em *Relative Phase Shift (RPS)* conjuntamente com representações de *delay* e CQCC. O resultado obtido foi uma taxa de erro de 18,58%.

Ainda em 2018, (SAILOR; KAMBLE; PATIL, 2018) publicaram um estudo descrevendo o desenvolvimento de um sistema cujo objetivo era detectar ataques de replay utilizando um banco de filtros associado a uma ConvRBM (Máquina de Convolução Restrita de Boltzmann), que usou um algoritmo de separação de energia (ESA) para a extração de características. O resultado final obtido foi uma taxa de erro de 8.89%.

Também em 2018, (TOM; JAIN; DEY, 2018) propuseram um sistema para detecção de ataques de replay que utilizava um mecanismo de atenção nas repesentações de tempo e frequência dos sinais de áudio, sendo este baseado em um grupo de características de delay. Os áudios então foram usados de input em uma adaptação da rede neural ResNet-18. O autor afirma ter alcançado uma taxa de erro de 0%.

Em 2019, (ALZANTOT; WANG; SRIVASTAVA, 2019) desenvolveram um modelo de detecção de replay attacks utilizando uma rede neural convolucional residual. Foram feitos experimentos altenando o input da rede entre três tipos diferentes de representação de características, sendo eles MFCC, STFT e CQCC, bem como a fusão dos três. O resultado obtido foi que a fusão das três repsentações conseguiu uma taxa de erro de 2,78%.

Também em 2019, (JUNG et al., 2019) sugeriram uma abordagem que utilizava espectrogramas de alta resolução, waverforms brutos e i-vector como inputs de uma DNN *fully-connected*. O resultado foi uma taxa de erro de 2.45% mesclando as representações de áudio.

Em 2020, (RAHULT et al., 2020) implementaram uma abordagem de detecção de

replay attacks utilizando as técnicas de representação de energia espectral Mel como input de uma rede neural residual, nesse caso uma adaptação da rede ResNet-34. Os resultados obtidos foram uma taxa de erro de 5,32%.

No ano de 2022, (FATHAN; ALAM; KANG, 2022) sugeriram utilizar diferentes codecs e técnicas de *data augmentation* de áudio para gerar as informações espectrais, e depois utilizaram as redes VGG e Wavelet CNN para a classificação. Seu melhor resultado foi utilizando a técnica Cosine Restarts com uma taxa de EER de 2,8%

Também em 2022, (WU et al., 2022) desempenharam um estudo cujo objetivo era identificar áudios parcialmente *fakes*, ou seja, apontar quais trechos específicos dos áudios foram gerados de forma fraudulenta. Para isso, eles utilizaram as frequências MSTFT, MFCC e LFCC. O melhor resultado obtido foi de um EER de 12,5% a partir da frequência LFCC.

Em 2023, (XU et al., 2023) desenvolveram um estudo cujo foco era desacoplar as informações dos aúdios que fossem pertencentes aos ruídos dos dispositivos de gravação e reprodução. Para isso, eles utilizaram transformações lineares, uma DNN e uma ResNet com várias features diferentes. Como resultado, obtiveram uma tava de EER de 13,03% para a DNN e 14,34% para a ResNet.

Autor	Método	EER
WITKOWSKI et al., 2017	**Transformada de Fourier**	**30%**
CAI et al., 2017	**CQCC-GMM com CNN, FDNN e BLSTM**	**16,39%**
PATIL et al., 2017	**Algoritmo de separaçõ de energia ESA associado com VESA-IFCC e CQCC**	**18,33%**
NAGARSHETH et al., 2017	**DNN com áudios espectrais de alto nível (HFCC)**	**11,5%**
PATIL; KAMBLE, 2018	**Banco de filtros associado a uma ConvRBM com algoritmo ESA**	**11,5%**
TOM; JAIN; DEY, 2018	**Mecanismo de atenção em representação de tempo e frequência e ResNEt-18**	**0%**
ALZANTOT; WANG; SRIVASTAVA, 2019	**Rede neural residual com áudios representados em MFCC, STFT e CQCC**	**2,78%**
JUNG et al., 2019	**Waveforms brutos e i-vector como input de uma DNN fully-connected**	**2,45%**
RAHULT et al., 2020	**Áudios a frequência Mel como input de uma adaptação da ResNEt-34**	**5,32%**
FATHAN; ALAM; KANG, 2022	**Codecs e audio augmentation como input de VGG e Wavelet CNN**	**2,8%**
WU et al., 2022	**Detecção de áudios parcialmente fakes representados nas frequências MSTFT, MFCC e LFCC.**	**12,5%**
XU et al., 2023	**Transformações lineares nos áudios de input de uma DNN e ResNet**	**13,03%(DNN) e 14,34%(ResNet)**

Tabela 1 – Estado da Arte

Fonte: Autoria Própria

4 Metodologia

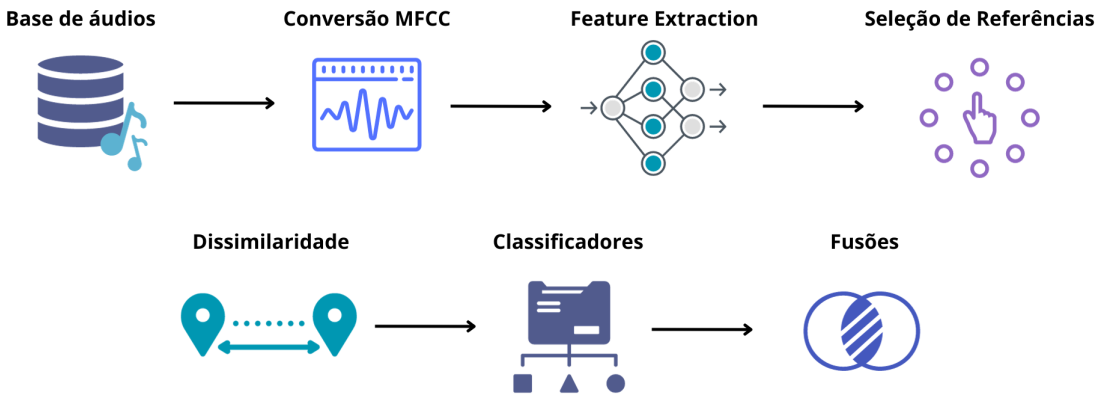
Nesse capítulo é descrita a estrutura da base de dados utilizada na condução dos experimentos, bem como uma descrição individual de cada um dos testes realizados.

4.1 Experimentos

Para a realização dos experimentos iniciais foram utilizadas as pastas de *Training Set* e *Dev Set (test)* da base de dados. O processo foi dividido em 6 etapas:

1. Análise dos dados e Balanceamento
2. Conversão dos áudios para MFCC
3. Extração de características
4. Seleção de referências (k-means)
5. Cálculo da dissimilaridade
6. Classificação com as técnicas de fusão de votação da maioria e soma das probabilidades.

Figura 16 – Fluxo de etapas dos experimentos



Fonte: Autoria Própria

A linguagem utilizada para a implementação dos experimentos foi a Python 3, instalada através do Anaconda. Eles foram desenvolvidos no formato de Jupyter Notebooks, utilizando uma extensão do Jupyter no Visual Studio Code. O projeto foi executado em

um computador do modelo Dell G3, com 500GB de armazenamento, processador intel i7, 16GB de memória RAM e placa gráfica Nvidia Geforce RTX 2060.

4.1.1 Análise dos dados

Para analisar os dados, primeiramente foi necessário obter informações em relação à quantidade de áudios *spoof* e *bonafide* havia em cada uma das pastas. Como pode-se observar na Tabela 2, na pasta de *Training Set*, havia um total de 54.000 áudios, sendo 5.400 *bonafide* (genuínos) e 48.600 *spoof* (fakes). Já na pasta de Dev Set, havia um total de 134.730 áudios, sendo 18.090 *bonafide* e 116.640 *spoof*.

Divisão	**Bonafide**	**Spoof**	**Total**
Training	**5.400**	**48.600**	**54.000**
Development	**5.400**	**24.300**	**29.700**
Evaluation	**18.090**	**116.640**	**134.730**

Tabela 2 – Distribuição de áudios na base ASVspoof 2019

Fonte: Autoria Própria

Com esses valores, foi possível perceber que a base estava desbalanceada, pois na pasta de Training havia um número muito maior de áudios *spoof* do que de áudios *bonafide*. O desbalanceamento de uma base pode prejudicar muito os resultados do sistema, portanto é necessário realizar o balanceamento dos dados.

Nesse caso, foram testadas duas maneiras de balanceamento. A primeira, foi remover os áudios em excesso da classe de spoof, deixando assim a pasta de Training com 5.400 áudios *bonafide* e 5.400 áudios *spoof*, para que o treinamento não pendesse para nenhum lado. A desvantagem dessa técnica, é que há uma perda de dados e informações muito grande.

Outra técnica de balancemaneto considerada foi a de redimensionamento da classe bonafide, ou seja, foi aplicada uma abordagem denominada SMOTE (Synthetic Minority Oversampling Technique) para a geração artificial, com base nos áudios já existentes, de mais exemplos de bonafide para equilibrar a base, no entanto, devido ao tempo de processamento e necessidade de melhores recursos computacionais para as execuções, foi optado seguir com a primeira alternativa de balanceamento por remoção.

Além disso, foram executados experimentos com e sem o processo de redução de características (PCA).

4.1.2 Conversão dos áudios

Para que seja possível analisar as características dos áudios, é necessário convertê-los em algum tipo de frequência e/ou espectrograma. Para estes experimentos, foi escolhido

converter os áudios em MFCCs (*Mel Frequency Cepstral Coefficients*), pois é uma das representações de sinais de áudio mais utilizadas nos estudos da área, já que consegue destacar até mesmo as características mais ocultas dos sinais. Para esse processo, foi utilizada a biblioteca Librosa para Python, que possui variadas funções para análise e manipulação de áudios. Os parâmetros utilizados na conversão foram os valores de 13 bandas de mfcc, e um sample rate de 22050Hz.

4.1.3 Extração de características

Foram realizadas as extrações de características dos áudios utilizando 3 arquiteturas distintas de redes neurais: VGG 16, InceptionV3 e ResNet50. Considerando que as redes foram utilizadas apenas para a extração de características, foi necessário remover a camada final delas para que dessa forma, o processo fosse interrompido previamente à etapa da classificação. Nas figuras 17, 18 e 19, é possível observar a arquitetura da VGG, Inception e ResNet respectivamente.

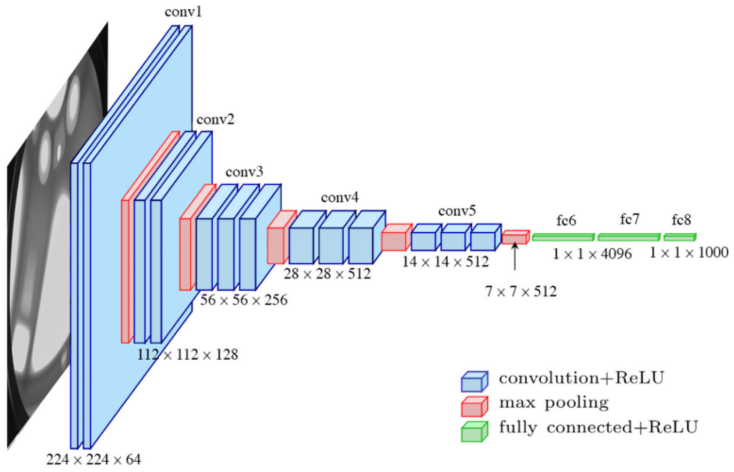


Figura 17 – Arquitetura VGG 16

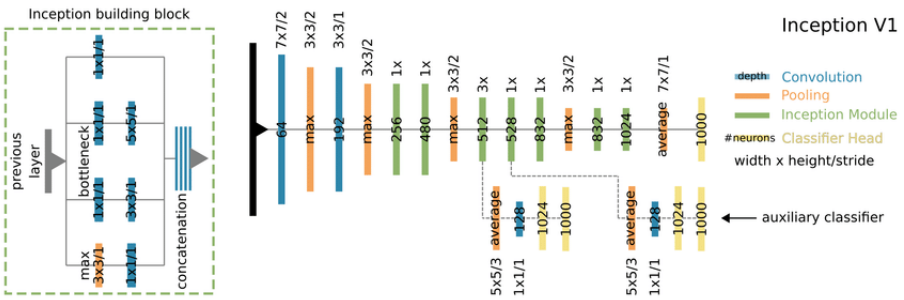


Figura 18 – Arquitetura InceptionV3

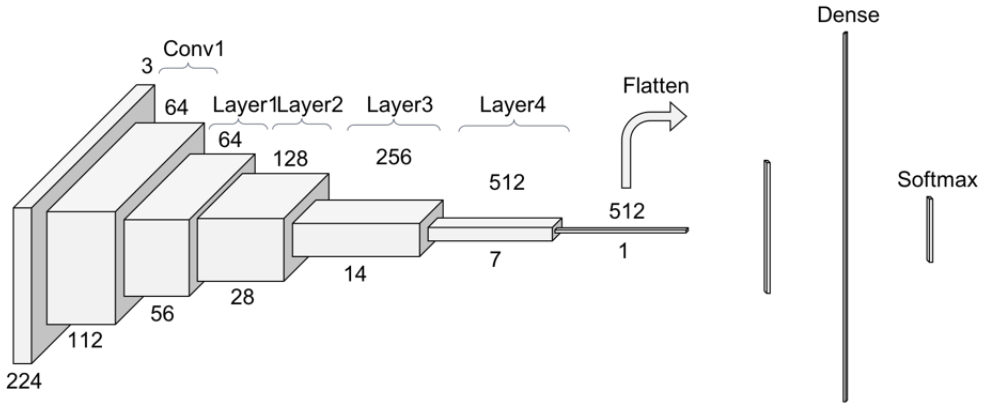


Figura 19 – Arquitetura ResNet50

4.1.4 Seleção de Referências

Para que o cálculo de dissimilaridade possa ser realizado, é necessário selecionar vetores de referência de cada uma das classes. Para esta etapa, foi utilizado o algoritmo *k-means*, que divide a distribuição dos pontos em *clusters* e seleciona o ponto central (centróide) de cada um dos *clusters*. Embora o k-means seja um bom algoritmo, a distribuição dos áudios desse problema é bastante complexa, já que todos estão muito próximos e desordenados, como é possível observar na figura 20.

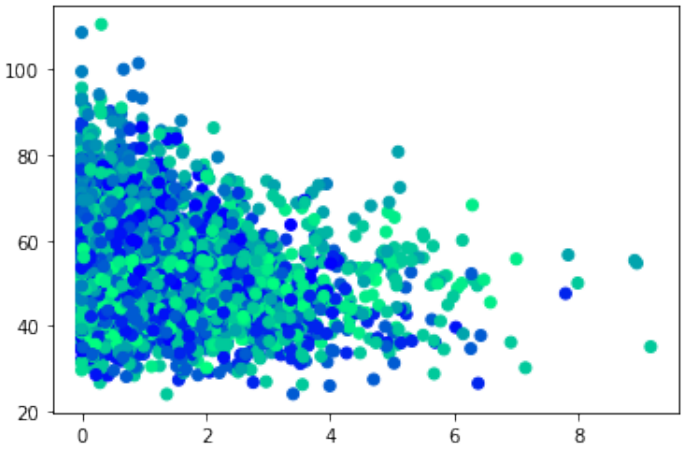


Figura 20 – Distribuição dos áudios no espaço

Fonte: Autoria Própria

Em relação ao número de referências selecionadas, foi definida a realização de experimentos com 5, 10 e 15 vetores de referência de cada uma das duas classes (*spoof*) e *bonafide*). Na imagem 21 é apresentada a localização dos centróides escolhidos pelo *k-means* quando solicitadas 15 referências.

O processo de seleção dos vetores de referência pode ser observado na Figura 22, onde é ilustrada a base de dados que é utilizada como entrada para o *k-means*, e as

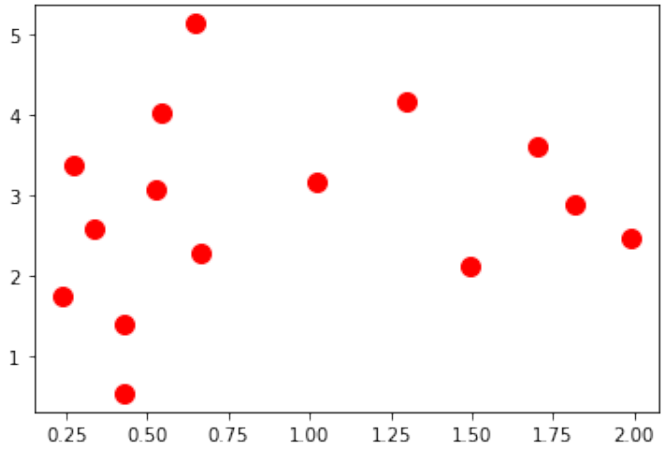


Figura 21 – Centróides

Fonte: Autoria Própria

referências *spoof* e *bonafide* geradas por ele.

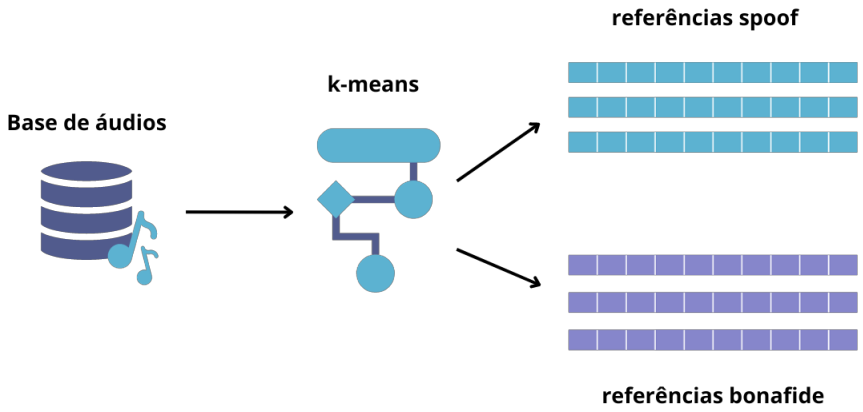


Figura 22 – Seleção de referências

Fonte: Autoria Própria

4.1.5 Cálculo de Dissimilaridade

É preciso ressaltar que nos experimentos deste estudo, foram utilizadas as duas abordagens de dissimilaridade: Vetores de Dissimilaridade e Espaço de Dissimilaridade.

Na abordagem de Vetores de Dissimilaridade, para cada posição de um vetor da base, é feita a subtração em relação à sua posição equivalente do vetor de referência, cujo cálculo pode ser observado a partir da Equação (Equação 4.2), onde d é a distância, v o vetor original, e r a referência. No caso deste estudo, foi definido que os vetores de distância obtidos a partir da dissimilaridade, serão a entrada de um classificador que dirá o resultado final. Além disso, foi optado por separar os classificadores de acordo com a classe das referências, ou seja, os vetores de distância obtidos a partir do cálculo das

referências spoof, foram utilizados como entrada para um classificador, e da mesma forma, os vetores resultantes do cálculo a partir das referências bonafide, foram utilizados como entrada para outro classificador. Na imagem 23, está ilustrado um exemplo de como isso ocorre, onde $Vet1$ seria um vetor de características da base original, $Ref1$, $Ref2$ e $Ref3$ seriam os vetores de referência e $Dist1$, $Dist2$ e $Dist3$, os vetores de distância obtidos pela dissimilaridade. Posteriormente, esses vetores resultantes são destinados à um classificador cada.

$$d\left(v, r\right) = \left(v_i - r_i\right) \tag{4.1}$$

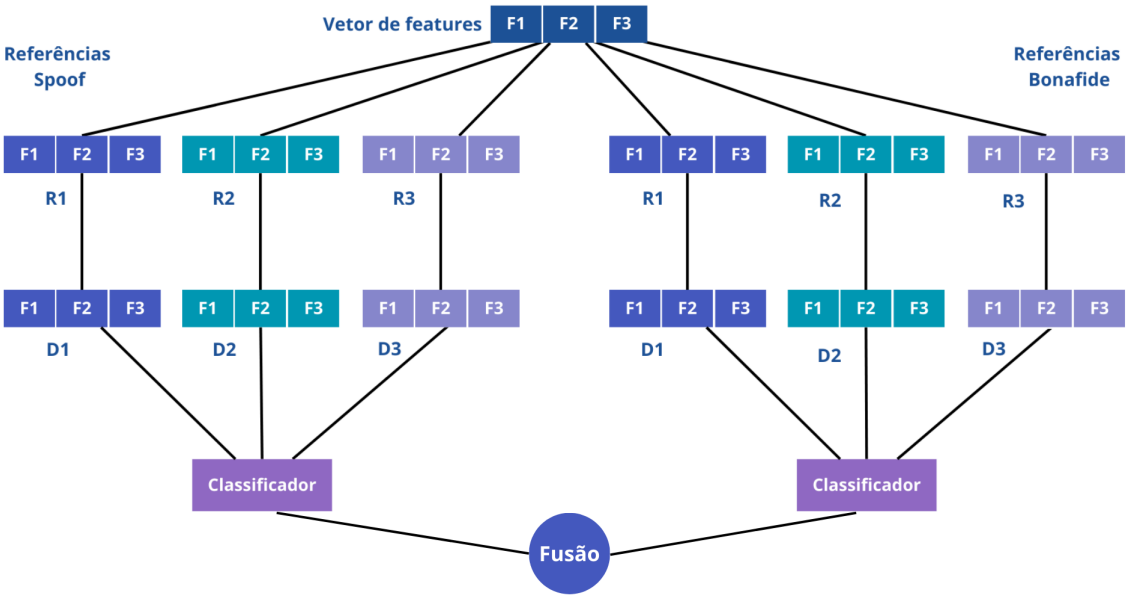


Figura 23 – Fluxo Vetores de Dissimilaridade

Fonte: Autoria Própria

No caso da técnica de Espaço de Dissimilaridade, é realizado o cálculo da distância Euclidiana (Equação (Equação 4.1)) entre os vetores de forma geral, ou seja, é criado um único vetor de distância para o vetor iriginal, onde cada uma de suas posições equivale à distância Euclidiana em relação à um vetor de referência, dessa forma, se houver 3 referências, será obtido um vetor de distância de 3 posições. O processo é ilustrado na Figura 24.

$$d\left(p, q\right) = \sqrt{\sum_{i=1}^{n} \left(q_i - p_i\right)^2} \tag{4.2}$$

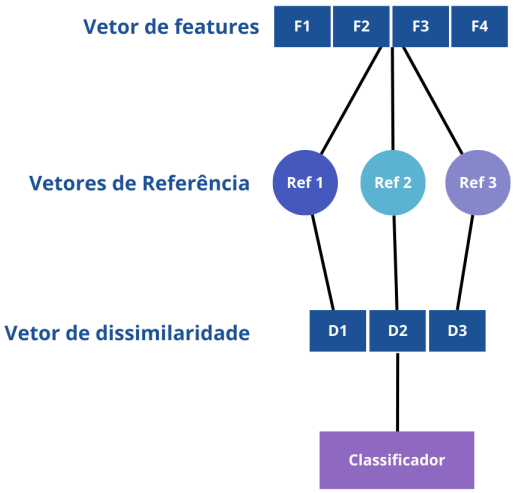


Figura 24 – Fluxo Espaço de Dissimilaridade

Fonte: Autoria Própria

4.1.6 Classificação

Após os vetores de diferença terem sido obtidos, eles são utilizados como input para um classificador, que finalmente irá fornecer o resultado de classificação de cada áudio. Nestes experimentos foram testados dois classificadores tradicionais, o SVM (Support Vector Machine) e o Random Forest, e também uma rede 1DCNN. Além dos resultados obtidos pela classificação individual com spoof e bonafide, também foram aplicadas duas técnicas de fusão, sendo elas: Votação pela maioria, a qual analisa a quantidade de vezes que um vetor de entrada foi classificado como spoof e como bonafide, e em seguida escolhe a classe com o maior número de classificação, e a técnica de Soma das probabilidades, na qual são somadas as probabilidades resultantes dos classificadores à qual classe o vetor pertenceria para definir o resultado de classificação final.

5 Resultados

No presente capítulo são apresentados os resultados de cada um dos experimentos conduzidos, bem como uma discussão em relação aos valores obtidos.

5.1 Experimentos

5.1.1 Experimentos em Feature Space

Inicialmente, o fluxo de experimentos foi realizado sem a etapa do cálculo de dissimilaridade, para que posteriormente fosse possível comparar e avaliar o impacto da abordagem de dissimilaridade no contexto do problema de ataques de replay. Os experimentos foram executados com as três arquiteturas de redes neurais previamente citadas: Inception, ResNet e VGG, combinadas com os classificadores Random Forest e SVM, bem como uma rede 1DCNN. Os resultados de Acurácia, EER e f1-score obtidos a partir desta primeira etapa podem ser observados na tabela 25.

Para os experimentos em Feature Space, o melhor resultado obtido foi o da ResNet, utilizando a 1DCNN para a classificação, com os dados desbalanceados e sem o uso do PCA. Nele, a taxa de EER atingiu **0.0937**. Além disso, a taxa de EER com menos acertos foi a do experimento a partir da rede VGG, com a 1DCNN, dados balanceados e sem PCA, obtendo **0.4244**. Nessa etapa dos experimentos, foi possível observar que a rede com o melhor desempenho foi a ResNet, já que, além de ter obtido o melhor resultado dentre todas as combinações, o seu menor EER foi de **0.2540**, enquanto as redes VGG e Inception obtiveram quase todos os seus valores acima deste. Na figura 25(a) pode-se obsevar o classification report e na figura 25(b) está representada a matriz de confusão, nela é possível observar que o número de acertos foi muito superior para a classe spoof.

	precision	recall	f1-score	support
0	0.93	0.98	0.95	24300
1	0.87	0.66	0.75	5400
accuracy			0.92	29700
macro avg	0.90	0.82	0.85	29700
weighted avg	0.92	0.92	0.91	29700

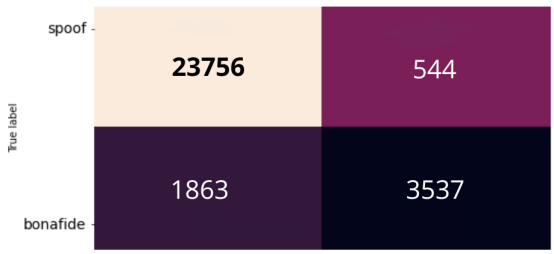


Figura 25 – a) métricas b) matriz de confusão

Fonte: Autoria Própria

CNN	Architecture	Representation	Train Samples	Feature Selection	EER	ACC	Macro F1	Weigthed F1
ResNet	1DCNN	feature_space	balanced	nopca	0.1143	0.8990	0.84	0.90
ResNet	1DCNN	feature_space	balanced	pca	0.1173	0.8821	0.83	0.89
ResNet	**1DCNN**	**feature_space**	**imbalanced**	**nopca**	**0.0937**	**0.9190**	**0.85**	**0.91**
ResNet	1DCNN	feature_space	imbalanced	pca	0.1200	0.9029	0.81	0.89
ResNet	RandomForest	feature_space	balanced	nopca	0.1534	0.8576	0.79	0.87
ResNet	RandomForest	feature_space	balanced	pca	0.2030	0.8474	0.76	0.85
ResNet	RandomForest	feature_space	imbalanced	nopca	0.1648	0.8303	0.52	0.76
ResNet	RandomForest	feature_space	imbalanced	pca	0.2540	0.8182	0.45	0.74
ResNet	SVM	feature_space	balanced	nopca	0.1150	0.9010	0.84	0.90
ResNet	SVM	feature_space	balanced	pca	0.1060	0.9089	0.85	0.91
ResNet	SVM	feature_space	imbalanced	nopca	0.1058	0.9079	0.81	0.90
ResNet	SVM	feature_space	imbalanced	pca	0.1057	0.9150	0.83	0.91
VGG	1DCNN	feature_space	balanced	nopca	0.4244	0.1818	0.15	0.06
VGG	1DCNN	feature_space	balanced	pca	0.2486	0.7345	0.67	0.76
VGG	1DCNN	feature_space	imbalanced	nopca	0.2626	0.8182	0.45	0.74
VGG	1DCNN	feature_space	imbalanced	pca	0.2617	0.8064	0.69	0.81
VGG	RandomForest	feature_space	balanced	nopca	0.2435	0.6919	0.64	0.73
VGG	RandomForest	feature_space	balanced	pca	0.3712	0.6147	0.55	0.66
VGG	RandomForest	feature_space	imbalanced	nopca	0.2379	0.8182	0.45	0.74
VGG	RandomForest	feature_space	imbalanced	pca	0.3781	0.8136	0.46	0.74
VGG	SVM	feature_space	balanced	nopca	0.4046	0.6381	0.54	0.67
VGG	SVM	feature_space	balanced	pca	0.2361	0.7682	0.69	0.79
VGG	SVM	feature_space	imbalanced	nopca	0.3322	0.8195	0.47	0.74
VGG	SVM	feature_space	imbalanced	pca	0.2535	0.8335	0.72	0.83
Inception	1DCNN	feature_space	balanced	nopca	0.2299	0.7479	0.68	0.77
Inception	1DCNN	feature_space	balanced	pca	0.2363	0.7665	0.69	0.79
Inception	1DCNN	feature_space	imbalanced	nopca	0.2358	0.8393	0.71	0.83
Inception	1DCNN	feature_space	imbalanced	pca	0.2547	0.8203	0.70	0.82
Inception	RandomForest	feature_space	balanced	nopca	0.2802	0.6939	0.63	0.73
Inception	RandomForest	feature_space	balanced	pca	0.3015	0.7347	0.64	0.76
Inception	RandomForest	feature_space	imbalanced	nopca	0.2974	0.8173	0.48	0.75
Inception	RandomForest	feature_space	imbalanced	pca	0.3530	0.8157	0.45	0.74
Inception	SVM	feature_space	balanced	nopca	0.2737	0.6990	0.64	0.73
Inception	SVM	feature_space	balanced	pca	0.2311	0.7743	0.70	0.79
Inception	SVM	feature_space	imbalanced	nopca	0.2262	0.8439	0.73	0.84
Inception	SVM	feature_space	imbalanced	pca	0.2526	0.8329	0.72	0.83

Tabela 3 – Resultados com *Feature Space*

5.1.2 Experimentos com Vetores de Dissimilaridade

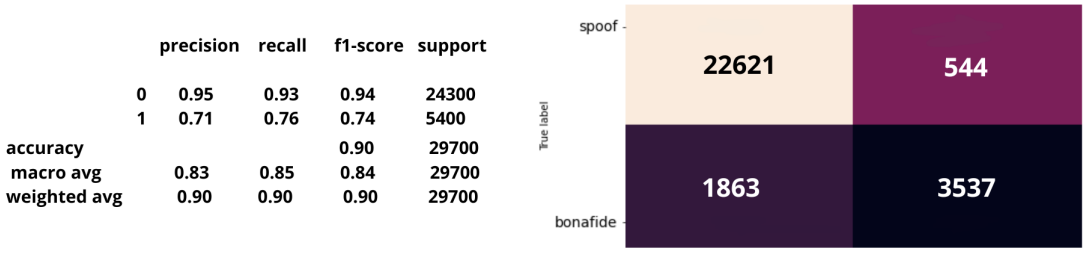
Como citado anteriormente, há duas abordagens distintas para a dissimilaridade. Nessa etapa, serão apresentados os resultados obtidos pela abordagem de Vetores de Dissimilaridade.

Considerando o fato de que nos experimentos anteriores os melhores resultados obtidos foram resultantes da ResNet, foi optado por aplicar a dissimilaridade apenas nos experimentos dessa rede. Novamente, foram utilizados os classificadores SVM e Random Forest e a rede 1DCNN e com dados balanceados e desbalanceados. Como foi realizado o processo do cálculo de dissimilaridade, também foi necessário testar com diferentes números de referências selecionadas pelo *k-means*, portanto foram feitos testes com 5, 10 e 15 referências. Além disso, foram executados experimentos considerando um modelo de classificação individual para cada classe (*spoof* e *bonafide*), bem como duas técnicas de fusão: Votação da maioria e Soma de probabilidades.

Neste conjunto de experimentos, o melhor resultado obtido foi um EER de **0.1211** para a ResNet utilizando a técnica de fusão de Soma das Probabilidades a partir dos

resultados dos classificadores SVM de spoof e bonafide, dados balanceados e 15 vetores de referência para o cálculo da dissimilaridade. Já o resultado mais inferior foi de **0.1653** em duas configurações: *Random Forest* com fusão por Soma com 15 referências e *Random Forest* com fusão por *Voting* também com 15 referêncisas. No geral, os resultados atingiram boas taxas nas métricas. Além disso, é possível observar que quando aplicada a dissimilaridade, o SVM obteve resultados superiores à 1DCNN, que foi a melhor técnica nos experimentos de *Feature Space*, no entanto, o Random Forest segue obtendo os resultados menos acurados.

Na figura 26(a) é apresentado o *classification report* e na 26(b), a matriz de confusão para o melhor resultado dessa etapa. Novamente, o número de acertos foi maior para a classe *Spoof*.



	precision	recall	f1-score	support
0	0.95	0.93	0.94	24300
1	0.71	0.76	0.74	5400
accuracy			0.90	29700
macro avg	0.83	0.85	0.84	29700
weighted avg	0.90	0.90	0.90	29700

Figura 26 – a) métricas b) matriz de confusão

Fonte: Autoria Própria

CNN	Architecture	Representation	Train Samples	Ref Vectors	EER	ACC	Macro F1	Weigthed F1
ResNet	1DCNN	dissimilarity_vector-sum	balanced	10	0.1323	0.8860	0.82	0.89
ResNet	1DCNN	dissimilarity_vector-sum	balanced	15	0.1317	0.8903	0.83	0.89
ResNet	1DCNN	dissimilarity_vector-sum	balanced	5	0.1329	0.8855	0.83	0.89
ResNet	1DCNN	dissimilarity_vector-vote	balanced	10	0.1323	0.8859	0.82	0.89
ResNet	1DCNN	dissimilarity_vector-vote	balanced	15	0.1317	0.8899	0.83	0.89
ResNet	1DCNN	dissimilarity_vector-vote	balanced	5	0.1329	0.8863	0.82	0.89
ResNet	RandomForest	dissimilarity_vector-sum	balanced	10	0.1618	0.8371	0.77	0.85
ResNet	RandomForest	dissimilarity_vector-sum	balanced	15	0.1653	0.8332	0.77	0.85
ResNet	RandomForest	dissimilarity_vector-sum	balanced	5	0.1650	0.8364	0.77	0.85
ResNet	RandomForest	dissimilarity_vector-vote	balanced	10	0.1618	0.8396	0.77	0.85
ResNet	RandomForest	dissimilarity_vector-vote	balanced	15	0.1653	0.8337	0.77	0.85
ResNet	RandomForest	dissimilarity_vector-vote	balanced	5	0.1650	0.8395	0.77	0.85
ResNet	SVM	dissimilarity_vector-sum	balanced	10	0.1219	0.8975	0.83	0.90
ResNet	**SVM**	**dissimilarity_vector-sum**	**balanced**	**15**	**0.1211**	**0.9003**	**0.84**	**0.90**
ResNet	SVM	dissimilarity_vector-sum	balanced	5	0.1263	0.8940	0.83	0.90
ResNet	SVM	dissimilarity_vector-vote	balanced	10	0.1219	0.8985	0.83	0.90
ResNet	SVM	dissimilarity_vector-vote	balanced	15	0.1211	0.9009	0.84	0.90
ResNet	SVM	dissimilarity_vector-vote	balanced	5	0.1263	0.8958	0.83	0.90

Tabela 4 – Resultados de Vetores de Dissimilaridade

5.1.3 Experimentos com Espaço de Dissimilaridade

O último conjunto de experimentos a ser apresentado é o de Espaço de Dissimilaridade. Novamente, foi utilizada a rede ResNet para a extração de características, os classificadores SVM e *Random Forest* e a rede 1DCNN para classificação.

Para este conjunto, o melhor resultado atingido foi o de uma taxa de EER de **0.1659** com o classificador SVM, dados balanceados e a utilização de 10 referências. Além disso, novamente os piores resultados foram os oriundos do Random Forest. Na figura 27(a) é apresentado o *classification report* e na figura 27(a) a matriz de confusão para o melhor resultado desse conjunto. A classe *Spoof* novamente teve uma taxa de acertos superior à classe *Bonafide*.

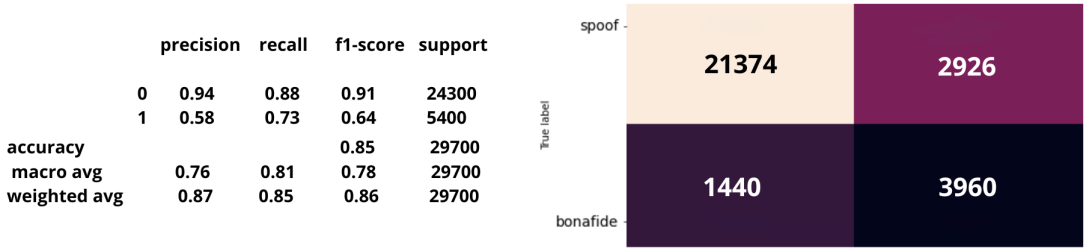


Figura 27 – a) métricas b) matriz de confusão

Fonte: Autoria Própria

CNN	Archi	Representation	Train Samples	Ref Vectors	EER	ACC	Macro F1	Weigthed F1
ResNet	1DCNN	dissimilarity_space	balanced	10	0.1917	0.7964	0.73	0.82
ResNet	1DCNN	dissimilarity_space	balanced	15	0.1846	0.8009	0.74	0.82
ResNet	1DCNN	dissimilarity_space	balanced	5	0.2048	0.7190	0.67	0.75
ResNet	1DCNN	dissimilarity_space	imbalanced	10	0.1914	0.8398	0.58	0.79
ResNet	1DCNN	dissimilarity_space	imbalanced	15	0.1991	0.8347	0.55	0.78
ResNet	1DCNN	dissimilarity_space	imbalanced	5	0.1967	0.8356	0.56	0.78
ResNet	RandomForest	dissimilarity_space	balanced	10	0.2029	0.8137	0.74	0.83
ResNet	RandomForest	dissimilarity_space	balanced	15	0.2061	0.8059	0.73	0.82
ResNet	RandomForest	dissimilarity_space	balanced	5	0.2059	0.8070	0.73	0.82
ResNet	RandomForest	dissimilarity_space	imbalanced	10	0.1958	0.8329	0.54	0.77
ResNet	RandomForest	dissimilarity_space	imbalanced	15	0.1946	0.8326	0.54	0.77
ResNet	RandomForest	dissimilarity_space	imbalanced	5	0.2020	0.8334	0.55	0.78
ResNet	**SVM**	**dissimilarity_space**	**balanced**	**10**	**0.1659**	**0.8530**	**0.78**	**0.86**
ResNet	SVM	dissimilarity_space	balanced	15	0.1720	0.8482	0.77	0.86
ResNet	SVM	dissimilarity_space	balanced	5	0.1890	0.8322	0.75	0.84
ResNet	SVM	dissimilarity_space	imbalanced	10	0.1880	0.8542	0.64	0.82
ResNet	SVM	dissimilarity_space	imbalanced	15	0.1826	0.8553	0.64	0.82
ResNet	SVM	dissimilarity_space	imbalanced	5	0.2221	0.8440	0.60	0.80

Tabela 5 – Resultados de Espaço de Dissimilaridade

6 Discussões e Conclusão

No presente estudo foi abordado o problema da detecção de ataques de *replay* no âmbito do *audio spoofing*. Nele, foram descritas as principais características dos ataques de *spoofing*, quais os principais estudos publicados dentro do contexto, além da sugestão e implementação de uma abordagem que ainda não havia sido aplicada na esfera dos sistemas ASV.

A principal abordagem sugerida nesse projeto foi a da utilização de redes neurais para a extração de características dos áudios, em conjunto com o método do cálculo de dissimilaridade e da aplicação dos resultados obtidos no cálculo como a entrada de um classificador tradicional para a obtenção de um sistema capaz de detectar áudios fraudulentos e genuínos. Conforme os experimentos descritos neste estudo, foi possível observar que as duas abordagens de dissimilaridade aplicadas em detecção de *Replay Attacks*, embora ainda não tenham sido suficientes para superar os valores resultantes dos experimentos que não fizeram o uso da dissimilaridade, atingiram resultados bastante satisfatórios.

Neste estudo, foram conduzidos 72 experimentos, englobando diferentes combinações de técnicas, considerando 3 diferentes arquiteturas de redes neurais para extração de características, abordagens com e sem dissimilaridade e 3 tipos de classificação.

Considerando todos os resultados obtidos a partir das diferentes configurações experimentais, foi possível verificar que, embora a dissimilaridade tenha obtido bons resultados, a melhor taxa de EER ainda foi a obtida na abordagem sem o cálculo de dissimilaridade. Além disso, dentro das duas abordagens existentes para a dissimilaridade, a técnica de Vetores de Dissimilaridade se sobressaiu, especialmente quando usado o classificador SVM. Também foi possível concluir que nessa técnica, os experimentos cujo número de vetores de referência utilizados foi 15, houve maior acurácia e menor EER. Já no caso do Espaço de Dissimilaridade, os melhores resultados foram obtidos a partir da utilização de 10 referências. Também precisa-se ressaltar que o classificador SVM conseguiu atingir os melhores resultados nas duas abordagens de dissimilaridade, perdendo apenas na técnica de Feature Space, na qual a rede 1DCNN se sobressaiu. Em relação ao número de acertos apresentado nas matrizes de confusão, nota-se que a classe Spoof possui um número significativamente maior em relação aos acertos da classe Bonafide, o que indica que o algoritmo está tendendo mais para uma classe do que para outra, fator que pode ocorrer devido a problemas de balanceamento ou número pequeno de amostras.

Como citado nas seções anteriores, o objetivo deste projeto pode ser resumido como uma investigação do impacto da dissimilaridade dentro do contexto de detecção de

áudios fakes. Com base nos experimentos realizados, foi possível avaliar o desempenho das abordagens de dissimilaridade dentro deste contexto, portanto, pode-se afirmar que o objetivo deste trabalho foi atingido.

Diante disso, pode-se observar que a dissimilaridade é uma técnica de bastante potencial quando aplicada dentro do contexto de detecção de áudios *fakes*. Além disso, com explorações mais aprofundadas e outras configurações experimentais, existe a possibilidade de que sejam atingidos resultados superiores aos apresentados nesse estudo.

7 Trabalhos Futuros

Para a realização de trabalhos futuros, sugere-se que sejam melhor exploradas e refinadas as técnicas de seleção de vetores de referência para o cálculo de dissimilaridade. Além disso, testar diferentes abordagens de balanceamento dos dados e realizar a execução dos experimentos com a base completa. Também é interessante aplicar outras arquiteturas de redes neurais para a extração de características, bem como inserir a abordagem de fine-tunning nas redes utilizadas. Por fim, explorar outros classificadores e redes para a etapa de classificação final, além de outras técnicas de fusão.

Referências

ABDI, Hervé; VALENTIN, Dominique; EDELMAN, Betty. *Neural networks*. [S.l.]: Sage, 1999. Citado na página 21.

ALBER, Maximilian; LAPUSCHKIN, Sebastian; SEEGERER, Philipp; HAGELE, Miriam; SCHUTT, Kristof T; MONTAVON, Gregoire; SAMEK, Wojciech; MULLER, Klaus-Robert; DAHNE, Sven; KINDERMANS, Pieter-Jan. innvestigate neural networks! p. 8, 2018. Citado na página 22.

ALZANTOT, Moustafa; WANG, Ziqi; SRIVASTAVA, Mani B. Deep residual neural networks for audio spoofing detection. In: *Interspeech 2019*. ISCA, 2019. p. 1078–1082. Disponível em: <https://www.isca-speech.org/archive/interspeech_2019/alzantot19_interspeech.html>. Citado na página 31.

AMELIO, Alessia; BONIFAZI, Gianluca; CAUTERUCCIO, Francesco; CORRADINI, Enrico; MARCHETTI, Michele; URSINO, Domenico; VIRGILI, Luca. Representation and compression of residual neural networks through a multilayer network based approach. *Expert Systems with Applications*, v. 215, p. 119391, 2023. ISSN 0957-4174. Citado na página 24.

BALAMURALI, BT; LIN, Kinwah Edward; LUI, Simon; CHEN, Jer-Ming; HERREMANS, Dorien. Toward robust audio spoofing detection: A detailed comparison of traditional and learned features. *IEEE Access*, IEEE, v. 7, p. 84229–84241, 2019. Citado 3 vezes nas páginas 17, 19 e 20.

BERANEK, Brett. Voice biometrics: Success stories, success factors and what's next. *Biometric technology today*, Elsevier, v. 2013, n. 7, p. 9–11, 2013. Citado na página 16.

CAI, Weicheng; CAI, Danwei; LIU, Wenbo; LI, Gang; LI, Ming. Countermeasures for automatic speaker verification replay spoofing attack: On data augmentation, feature representation, classification and fusion. In: *Interspeech 2017*. ISCA, 2017. p. 17–21. Disponível em: <https://www.isca-speech.org/archive/interspeech_2017/cai17_interspeech.html>. Citado na página 31.

COSTA, Yandre M. G.; BERTOLINI, Diego; BRITTO, Alceu S.; CAVALCANTI, George D. C.; OLIVEIRA, Luiz E. S. The dissimilarity approach: a review. *Artificial Intelligence Review*, v. 53, n. 4, p. 2783–2808, Apr 2020. ISSN 0269-2821, 1573-7462. Citado 5 vezes nas páginas 14, 24, 25, 26 e 27.

DUIN, Robert PW; PEKALSKA, Elzbieta. *Dissimilarity Representation For Pattern Recognition, The: Foundations And Applications*. [S.l.]: World scientific, 2005. v. 64. Citado na página 24.

DUIN, Robert P.W.; PęKALSKA, Elżbieta. The dissimilarity space: Bridging structural and statistical pattern recognition. v. 33, p. 826–832, May 2012. ISSN 01678655. Citado 2 vezes nas páginas 24 e 25.

DUIN, Robert P. W.; LOOG, Marco; PKALSKA, Elżbieta; TAX, David M. J. Feature-based dissimilarity space classification. In: _____. *Recognizing Patterns in*

Signals, Speech, Images and Videos. Springer Berlin Heidelberg, 2010. (Lecture Notes in Computer Science, v. 6388), p. 46–55. ISBN 978-3-642-17710-1. Disponível em: <http://link.springer.com/10.1007/978-3-642-17711-8_5>. Citado na página 24.

ERGUNAY, Serife Kucur; KHOURY, Elie; LAZARIDIS, Alexandros; MARCEL, Sebastien. On the vulnerability of speaker verification to realistic voice spoofing. p. 6, 2015. Citado na página 29.

EVANS, Nicholas WD; KINNUNEN, Tomi; YAMAGISHI, Junichi. Spoofing and countermeasures for automatic speaker verification. In: *Interspeech*. [S.l.: s.n.], 2013. p. 925–929. Citado na página 16.

FATHAN, Abderrahim; ALAM, Jahangir; KANG, Woo Hyun. Mel-spectrogram image-based end-to-end audio deepfake detection under channel-mismatched conditions. In: *2022 IEEE International Conference on Multimedia and Expo (ICME)*. [S.l.: s.n.], 2022. p. 1–6. Citado na página 32.

GALBALLY, Javier; MARCEL, Sébastien; FIERREZ, Julian. Biometric antispoofing methods: A survey in face recognition. *IEEE Access*, IEEE, v. 2, p. 1530–1552, 2014. Citado na página 16.

JUNG, Jee-weon; SHIM, Hye-jin; HEO, Hee-Soo; YU, Ha-Jin. Replay attack detection with complementary high-resolution information using end-to-end dnn for the asvspoof 2019 challenge. In: *Interspeech 2019*. ISCA, 2019. p. 1083–1087. Disponível em: <https://www.isca-speech.org/archive/interspeech_2019/jung19_interspeech.html>. Citado 3 vezes nas páginas 13, 18 e 31.

LAVRENTYEVA, Galina; NOVOSELOV, Sergey; MALYKH, Egor; KOZLOV, Alexander; KUDASHEV, Oleg; SHCHEMELININ, Vadim. Audio replay attack detection with deep learning frameworks. In: *Interspeech 2017*. ISCA, 2017. p. 82–86. Disponível em: <https://www.isca-speech.org/archive/interspeech_2017/lavrentyeva17_interspeech.html>. Citado na página 13.

NAGARSHETH, Parav; KHOURY, Elie; PATIL, Kailash; GARLAND, Matt. Replay attack detection using dnn for channel discrimination. In: *Interspeech 2017*. ISCA, 2017. p. 97–101. Disponível em: <https://www.isca-speech.org/archive/interspeech_2017/nagarsheth17_interspeech.html>. Citado na página 31.

O'SHEA, Keiron; NASH, Ryan. An introduction to convolutional neural networks. *arXiv:1511.08458 [cs]*, Dec 2015. ArXiv: 1511.08458. Disponível em: <http://arxiv.org/abs/1511.08458>. Citado 2 vezes nas páginas 22 e 23.

PATIL, Hemant A.; KAMBLE, Madhu R. A survey on replay attack detection for automatic speaker verification (asv) system. In: *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2018. p. 1047–1053. ISBN 978-988-14768-5-2. Disponível em: <https://ieeexplore.ieee.org/document/8659666/>. Citado 4 vezes nas páginas 16, 17, 19 e 31.

PATIL, Hemant A.; KAMBLE, Madhu R.; PATEL, Tanvina B.; SONI, Meet H. Novel variable length teager energy separation based instantaneous frequency features for replay detection. In: *Interspeech 2017*. ISCA, 2017. p. 12–16. Disponível em:

<https://www.isca-speech.org/archive/interspeech_2017/patil17_interspeech.html>. Citado na página 31.

QUATIERI, Thomas F. *Discrete-time speech signal processing: principles and practice*. [S.l.]: Pearson Education India, 2006. Citado na página 17.

RABINER, Lawrence; JUANG, Biing-Hwang. *Fundamentals of speech recognition*. [S.l.]: Prentice-Hall, Inc., 1993. Citado na página 20.

RAFI, B Shaik Mohammad; MURTY, K Sri Rama; NAYAK, Shekhar. A new approach for robust replay spoof detection in asv systems. In: IEEE. *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. [S.l.], 2017. p. 51–55. Citado na página 19.

RAHULT, P; ARAVIND, PR; RANJITH, C; NECHIYIL, Usamath; PARAMPARAMBATH, Nandakumar. Audio spoofing verification using deep convolutional neural networks by transfer learning. *ArXiv: abs/2008.03464*, 2020. Citado 3 vezes nas páginas 13, 21 e 31.

SAILOR, Hardik; KAMBLE, Madhu; PATIL, Hemant. Auditory filterbank learning for temporal modulation features in replay spoof speech detection. In: *Interspeech 2018*. ISCA, 2018. p. 666–670. Disponível em: <https://www.isca-speech.org/archive/interspeech_2018/sailor18c_interspeech.html>. Citado na página 31.

TOM, Francis; JAIN, Mohit; DEY, Prasenjit. End-to-end audio replay attack detection using deep convolutional networks with attention. In: *Interspeech 2018*. ISCA, 2018. p. 681–685. Disponível em: <https://www.isca-speech.org/archive/interspeech_2018/tom18_interspeech.html>. Citado na página 31.

WITKOWSKI, Marcin; KACPRZAK, Stanisław; ŻELASKO, Piotr; KOWALCZYK, Konrad; GAŁKA, Jakub. Audio replay attack detection using high-frequency features. In: . ISCA, 2017. p. 27–31. Disponível em: <https://www.isca-speech.org/archive/interspeech_2017/witkowski17_interspeech.html>. Citado na página 30.

WU, Haibin; KUO, Heng-Cheng; ZHENG, Naijun; HUNG, Kuo-Hsuan; LEE, Hung-Yi; TSAO, Yu; WANG, Hsin-Min; MENG, Helen. Partially fake audio detection by self-attention-based fake span discovery. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2022. p. 9236–9240. Citado na página 32.

WU, Zhizheng; EVANS, Nicholas; KINNUNEN, Tomi; YAMAGISHI, Junichi; ALEGRE, Federico; LI, Haizhou. Spoofing and countermeasures for speaker verification: A survey. *Speech Communication*, v. 66, p. 130–153, Feb 2015. ISSN 01676393. Citado na página 13.

XU, Longting; YANG, Jichen; YOU, Chang Huai; QIAN, Xinyuan; HUANG, Daiyu. Device features based on linear transformation with parallel training data for replay speech detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, v. 31, p. 1574–1586, 2023. Citado na página 32.

YAMAGISHI, Junichi; TODISCO, Massimiliano; SAHIDULLAH, Md; DELGADO, Héctor; WANG, Xin; EVANS, Nicholas; KINNUNEN, Tomi; LEE, K Aik; VESTMAN, Ville; NAUTSCH, Andreas. *Asvspoof 2019: Automatic speaker verification spoofing and countermeasures challenge evaluation plan*. [S.l.], 2019. Citado 2 vezes nas páginas 29 e 30.

YOON, Sunghyun; YU, Ha-Jin. Multiple-point input and time-inverted speech signal for the asvspoof 2021 challenge. In: *2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*. ISCA, 2021. p. 37–41. Disponível em: <https://www.isca-speech.org/archive/asvspoof_2021/yoon21_asvspoof.html>. Citado na página 16.