UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

INSTITUTO DE BIOCIÊNCIAS, LETRAS E CIÊNCIAS EXATAS
IBILCE / UNESP - CÂMPUS DE SÃO JOSÉ DO RIO PRETO

Heinrich Zago Doi

Estratégia anti-ataque de regravação para verificação de locutores

São José do Rio Preto

2022

Heinrich Zago Doi

Estratégia anti-ataque de regravação para verificação de locutores

Trabalho de Conclusão de Curso apresentado ao Departamento de Ciências de Computação e Estatística do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Aleardo Manacero Junior

São José do Rio Preto

2022

Heinrich Zago Doi

Estratégia anti-ataque de regravação para verificação de locutores

Trabalho de Conclusão de Curso apresentado ao Departamento de Ciências de Computação e Estatística do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Banca examinadora:

Prof. Dr. Aleardo Manacero Junior
UNESP – São José do Rio Preto
Orientador

Prof. Dr. Rodrigo Capobianco Guido
UNESP – São José do Rio Preto

Prof. Dr. Valeriano Antunes de Oliveira
UNESP – São José do Rio Preto

São José do Rio Preto
2022

Agradecimentos

Agradeço imensamente a toda minha família: Mara, Nide, Beto, Alice, Zinho, Yara, Darci, Bruno (e tantos outros que nem cabem numa lista), por todo o apoio ao longo desses anos, foi uma oportunidade única que não seria possível sem o auxílio e apoio de todos.

Ainda, sou grato a todos os professores e funcionários do IBILCE que nos incentivaram e motivaram a não apenas estudar e buscar mais conhecimento técnico, como também ajudaram a enxergar o mundo de uma maneira diferente.

‘ Agradeço também a Salesforce e meu gestor por todo o auxílio ao longo desses últimos meses, tanto com questões profissionais quanto acadêmicas e até mesmo pessoais.

Finalmente, a todos os meus amigos que estiveram comigo ao longo dessa jornada nada fácil, pois tenho certeza absoluta de que não teria chegado até aqui sem o apoio e auxílio de cada um de vocês, especialmente a todos os membros do Comp. Cinematic. Universe, minha companheira Lorena que auxiliou a não perder a cabeça e meu grande amigo Felipe que apesar do tempo e distância, sempre esteve comigo.

A todos os que me acompanharam até aqui, meu mais sincero Obrigado.

Resumo

Com a crescente evolução e popularização de ferramentas para autenticação biométrica por voz, a necessidade de garantir a confiabilidade e segurança desses mecanismos também aumenta proporcionalmente. Sistemas de verificação automática de locutor tentam garantir que o usuário tentando acessar determinado dispositivo/*software* é realmente quem alega ser. Porém, ataques nos quais grava-se a voz do alvo previamente e tenta-se utilizar a gravação para se passar pela pessoa ainda representam uma grande ameaça. Logo, encontra-se a necessidade de aprimorar as técnicas de identificação. Considerando esse objetivo, o *ASVspoof Challenge* foi lançado, de modo a não apenas prover uma base de dados consistente e robusta para a criação e melhoria das estratégias *Anti-spoofing*, mas também para estimular pesquisadores de todo o mundo a criarem suas próprias metodologias. Dessa forma, este trabalho consiste no desenvolvimento de uma estratégia para identificar e classificar ataques de regravação de locutor dentro do contexto previsto pelo *ASVspoof Challenge*. Com isso, espera-se que este trabalho possa contribuir para reduzir as taxas de erro de autenticação. Os resultados foram validados utilizando o método de avaliação de performance *Equal Error Rate*.

Palavras-chave: *Automatic Speaker Verification, ASVspoof Challenge, Spoofing*

Lista de Imagens

Lista de Tabelas

Lista de Quadros

Lista de Abreviaturas e Siglas

ASV	Automatic Speaker Verification (Verificação Automática de Locutor)
IoT	Internet of Things (Internet das Coisas)
VPN	Virtual Private Network (Rede Privada Virtual)
EER	Equal Error Rate (Taxa de erro igual)
FAR	False Acceptance Rate (Taxa de falso positivo)
FRR	False Rejection Rate (Taxa de falso negativo)
CNN	Convolutional Neural Network (Rede neural convolucional)
PA	Physical Access (Acesso físico)
LA	Logical Access (Acesso lógico)

Sumário

1 Introdução

Autenticação biométrica é um conceito na área de segurança de dados que consiste na utilização de características do corpo humano (impressão digital, padrões de voz, íris, entre outros) como forma de validar ou não um acesso. Esse tipo de verificação encontra-se cada vez mais presente nos mais diversos casos de uso: partindo de smartphones e dispositivos com tecnologias de "Internet das coisas" (IoT) até bancos, aeroportos e sistemas de segurança globais[1]. Dentro desse contexto, o reconhecimento de fala é uma alternativa em situações nas quais não é possível utilizar as mãos ou identificação facial devido ao uso de equipamentos de segurança, luvas ou quaisquer tipos de acessórios que possam vir a interferir. Ainda, em ocorrências como dentro de veículos, onde o condutor não pode se distrair, ou apenas quando se deseja somar mais de um método de validação, a verificação por fala é uma opção atrativa.

Sistemas que realizam todo o processo de reconhecimento, validação e tomada de decisão com análises computacionais, possuem a denominação de "*Automatic Speaker Verification (ASV) System*" ou "Sistema de verificação automática de locutor" (HANSEN et. al. 2015). Tais ferramentas atuam em duas etapas: inicialmente identificam o usuário e, em seguida, decidem se ele é realmente quem alega ser. A validação que ocorre nessa segunda etapa é necessária para identificar e prevenir possíveis ataques de falsificação, ou "*Spoofing*", que podem ser classificados como:

- ***Impersonation*:** Ocorre quando tenta-se produzir uma voz similar a do alvo, podendo ser feita por dubladores, mímicos ou até mesmo irmãos;

- ***Synthetic Speech*:** Ao invés de utilizar pessoas para tentar imitar uma fala, neste tipo de falsificação utiliza-se de ferramentas de *Text-to-Speech* para sintetizar a voz pelo computador;

- ***Voice conversion*:** Consiste em utilizar ferramentas para fazer a voz de uma pessoa, parecer com a do alvo;

[1] https://www.gov.br/pt-br/noticias/transito-e-transportes/2021/06/brasil-testa-primeira-ponte-aerea-com-reconhecimento-facial-do-mundo

- ***Replay*:** Ataque no qual a voz do alvo é gravada previamente e reproduzida no sistema em questão.

Dessa forma, é necessário que o sistema de ASV esteja preparado para lidar com os diversos tipos de ataques para que o processo de autenticação seja confiável. Neste trabalho, daremos ênfase aos ataques do tipo "*Replay*", pois além de afetarem seriamente tanto sistemas que dependem da interpretação do texto quanto os que não (PATIL et. al. 2018), com o avanço na qualidade de equipamentos de gravação e reprodução de sons, eles podem acabar passando despercebidos.

1.1 Motivação e justificativa

A utilização de sistemas de voz tanto para autenticação de usuários quanto para navegação e uso de aplicativos tem se tornado cada vez mais popular: assistentes virtuais como *Google Home* e *Amazon Echo* já possuem uma baixa taxa de erro[2] na identificação de palavras e são capazes de identificar e diferenciar vários usuários[3]. Além disso, quando se trata de acessibilidade, dispositivos como smartphones e *Wearables*[4] usam majoritariamente ferramentas que usam a fala do usuário como método de entrada de dados, tal qual *Speech-to-Text* (Voz para texto). Com isso, faz-se necessário garantir que tais sistemas possuam proteção contra ataques de *Spoofing*, uma vez que podem possuir informações sensíveis dos usuários como dados pessoais, métodos de pagamento e até mesmo acesso a listas de contatos.

Visando auxiliar pesquisadores e desenvolvedores a criar novas soluções para detectar e prevenir ataques, em 2015 foi criado o primeiro *ASVspoof challenge*[5]. O desafio consistia em prover um banco de dados, contendo *spoofings* gerados utilizando diversas técnicas, para que os participantes desenvolvessem técnicas de classificação e detecção em um contexto mais genérico e próximo do mundo real. As edições seguintes (2017 e 2019) evoluíram a

[2] https://venturebeat.com/2017/05/17/googles-speech-recognition-technology-now-has-a-4-9-word-error-rate/

[3] https://arstechnica.com/gadgets/2017/04/google-home-gets-support-for-multiple-users/

[4] https://usemobile.com.br/wearable/

[5] https://www.asvspoof.org/index2015.html

qualidade dos dados, focando mais em ataques do tipo *Replay,* tanto em cenários de *Physical Access* (Acesso físico ou PA), nos quais o *Spoofing* ocorre diretamente no microfone/entrada do sistema, quanto em *Logical Access* (Acesso Lógico ou LA), onde a amostra é inserida no sistema via software ou diretamente no processo de tomada de decisão (KAMBLE et al., 2020). Com isso, houve um aumento na quantidade de documentação e técnicas a respeito de estratégias para combater ataques de *Spoofing*.

Ainda, a evolução de técnicas para a extração de características em sinais digitais de fala (SHAYEB et al., 2020), juntamente de ferramentas baseadas em inteligência artificial e *Deep Learning* (ALHAWITI, 2015), abre um leque de possibilidades para novas estratégias de ASV sejam criadas visando não apenas confiabilidade, mas também desempenho e escalabilidade. Observa-se, portanto, a oportunidade de contribuir no combate a ataques de regravação em sistemas de verificação de locutores por meio do desenvolvimento de uma nova estratégia.

Ainda, considerando todos esses fatores, torna-se extremamente atrativo contribuir com uma área que não apenas auxilia pesquisadores acadêmicos como também beneficia diversos tipos de aplicações de mercado. Finalmente, o *ASVspoof challenge* facilita o ingresso nessa área, disponibilizando uma grande quantidade não apenas de dados, como referências.

1.2 Objetivo e escopo

Considerando os artigos e documentos já publicados dentro do contexto de ASV e também utilizando o *ASVspoof challenge*, esse trabalho busca definir uma estratégia para a identificação e categorização de ataques de regravação de voz (ou *Replay*). Essa metodologia proposta será para cenários de PA, buscando não apenas atender aos critérios propostos pelo desafio, como também propor uma maneira diferente de extrair características.

Com isso, espera-se contribuir não apenas com o *ASVspoof*, mas também para futuros projetos e pesquisas que busquem otimizar ou desenvolver novas técnicas de ASV. A forma como essa nova estratégia foi desenvolvida, será melhor descrita no capítulo 3.

1.3 Metodologia

O desenvolvimento deste trabalho foi estruturado em quatro etapas. A primeira etapa tem como objetivo a melhor compreensão sobre o cenário atual das pesquisas a respeito de ASV e ataques de *Replay*. Para tanto, tem-se uma seleção de técnicas e artigos já publicados tanto para o *ASVspoof challenge* 2019 quanto para as edições anteriores. Ainda, além da revisão bibliográfica, consta um breve resumo a respeito de conceitos de processamento digital de sinais que serão usados ao longo do trabalho.

Na sequência, durante a segunda etapa, os resultados obtidos com a coleta e pré-processamento de dados devem ser analisados e comparados com os guias e documentações já existentes na literatura com o objetivo de definir quais as melhores e mais eficientes práticas e técnicas.

Já na terceira etapa, com o embasamento teórico necessário adquirido, tem-se a proposta de uma estratégia para identificar, avaliar e categorizar possíveis ataques de regravação ou *Replay* para as amostras contidas no banco de dados.

Finalmente, tem-se como etapa final a comparação dos resultados obtidos por meio da nova estratégia com os demais algoritmos e técnicas já publicados para o mesmo banco de dados, com o intuito de evidenciar as potenciais contribuições alcançadas, identificar necessidades de adequações e estabelecer propostas para trabalhos futuros.

1.4 Organização da Monografia

Neste capítulo, foram introduzidos o contexto e necessidades acerca de sistemas de ASV e ataques de *Replay*, além da motivação e do objetivo do trabalho.

Os próximos capítulos estão organizados da seguinte forma:

a) Capítulo 2 - Revisão bibliográfica: estado da arte e revisão de conceitos.;
b) Capítulo 3 - Desenvolvimento: apresenta como o trabalho foi desenvolvido, e apresenta as considerações com relação a esse processo;
c) Capítulo 4 - Testes realizados e resultados obtidos são apresentados;
d) Capítulo 5 - Conclusões e considerações finais;
e) Referências - Contém todas as referências bibliográficas utilizadas nesta monografia.

2 Revisão Bibliográfica

Neste capítulo, são apresentados os temas fundamentais para a estruturação do trabalho. A primeira seção descreve os principais conceitos a respeito de processamento digital de sinais, filtros, *Encoders* e *Decoders*, além do processo de formação da fala. Em seguida, também serão abordadas algumas definições necessárias relacionadas às técnicas de *Deep Learning* utilizadas ao longo do projeto. Finalmente, o estado da arte, analisando e comparando alguns dos artigos e estratégias já existentes, será apresentado para melhor contextualização do papel deste trabalho.

2.1 Conceitos principais

Visando garantir a compreensão a respeito dos assuntos abordados, alguns dos principais conceitos necessários dentro do contexto de ASV serão descritos abaixo.

2.1.1 Formação da Voz

A voz humana é produzida pela junção de três fatores: O sistema respiratório, a laringe (que contém as pregas vocais) e ressonadores. De maneira simplificada (Ramos, 2013), o ar é inspirado para os pulmões por meio da movimentação do diafragma e expirado ativamente, passando pelas pregas vocais que podem se aproximar ou afastar para fazê-lo vibrar de diferentes formas, gerando som. O som, ainda, passa pelos ressonadores que “modificam” as vibrações antes que ele atinja, finalmente, sua forma final.

Dessa forma, como cada pessoa possui um timbre único, é necessário que os sistemas de ASV sejam preparados para lidar com os mais variados tipos de voz (principalmente os independentes de contexto), de modo a não causar falsos positivos ou negativos quando diferentes usuários forem utilizar. Portanto, de um ponto de vista computacional, nossa voz é na verdade um sinal analógico produzido pelo corpo que precisará ser ouvido pelo sistema em questão.

2.1.2 Sinais Digitais

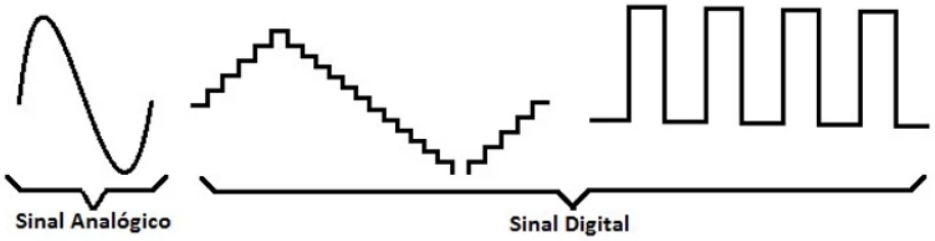
Sinais digitais são uma forma de representar sinais analógicos de maneira discreta (descontínua) em um determinado intervalo de tempo com uma amplitude dentro de um sistema computacional. Dentro do contexto aqui explorado, duas principais etapas são necessárias:

- **Amostragem:** Processo de transformar o sinal contínuo em um finito, por meio da seleção de valores com um intervalo regular. A frequência com que esses valores são obtidos, é chamada de **taxa de amostragem**.

- **Quantização:** Atribuição de valores discretos para a amplitude do sinal. A amplitude define a perturbação durante um ciclo da onda, estando relacionada diretamente com o volume do som.

O processo de transformar sinais analógicos em digitais é realizado por conversores analógico-digitais, e quanto maior a taxa de amostragem, maior a qualidade e fidelidade do sinal gerado. É possível observar uma comparação entre os dois tipos na Figura 1 .Dessa forma, a medida que um mesmo sinal passa por uma sequência de gravações e reproduções, ele pode acabar sendo modificado devido às diferentes características dos dispositivos envolvidos no processo. É possível observar esse fenômeno ao comparar o espectrograma de um sinal acústico original e sua regravação, a qual tende a possuir um mapa de calor mais acentuado nas frequências mais altas. Dentro do contexto do *ASVspoof*, considera-se arquivos de 16000 amostras por segundo, tanto para os sinais acústicos genuínos, quanto *Spoofs*.

Figura 1: Diferença entre sinal analógico e digital



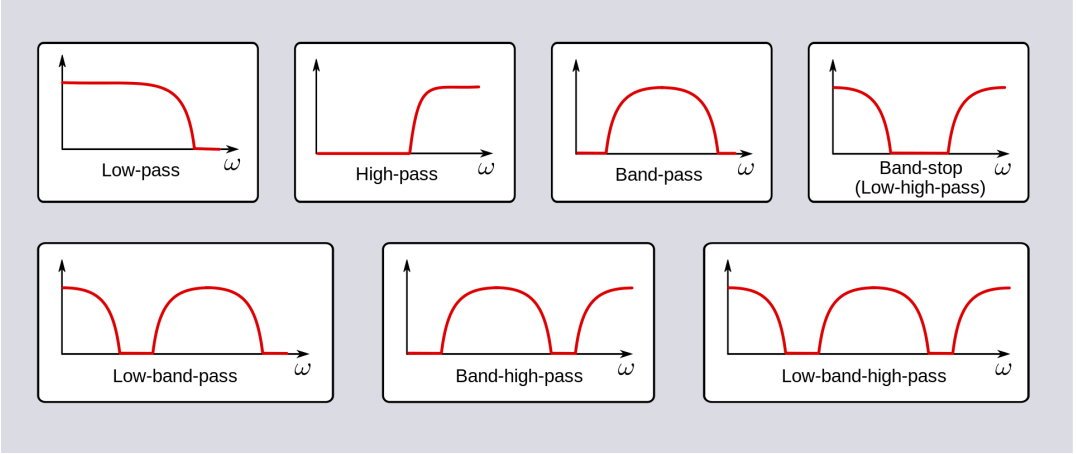
Fonte:https://eletronjun.com.br/2020/11/14/qual-e-a-diferenca-entre-eletronica-digital-e-analogica

Portanto, é somente possível operar em cima de sinais analógicos dentro de um sistema computacional por meio de sinais digitais, pois, uma vez que o primeiro possui "infinitas amostras", seria necessário um poder de processamento também infinito para trabalharmos com esses sinais.

2.1.3 Filtros

Uma vez que um sinal analógico foi devidamente representado por meio de um sistema digital, torna-se possível realizar operações em cima dele. O processo de remover ou adicionar determinados componentes ou características de um sinal, é chamado de Filtro. Existem muitos tipos diferentes de filtros que ainda podem ser operados juntos dependendo da situação, sendo os mais comuns aqueles que operam em cima das frequências dos áudios, aumentando ou reduzindo determinados intervalos. É possível observar alguns exemplos de filtros na Figura 2, na qual o eixo inferior representa as frequências e o superior a amplitude.

Figura 2: Exemplos de filtros



Fonte: www.masteringbox.com/filter-types/

Considerando que os filtros consistem em aplicar operações matemáticas em determinados valores do sinal digital (no caso da Figura 2 em determinadas frequências para bloqueá-las), eles também podem ser utilizados para reduzir a dimensão dos sinais por meio de operações de Convolução.

2.1.4 Operador de convolução

Convolução consiste em um operador linear no qual o resultado é a soma de dois sinais, por meio da soma dos produtos entre os valores dados. Dentro do contexto de processamento de sinais, são a maneira com a qual determinados filtros e *Kernels* são aplicados.

2.1.5 *Encoders* e *Decoders*

Devido a grande quantidade de informação necessária para conseguir representar digitalmente sinais analógicos, diversas técnicas de compressão de áudio (PAN, 1993) foram criadas para otimizar o armazenamento e transmissão desses tipos de dados. Esse processo consiste em aplicar uma função matemática que reduz o tamanho do arquivo em questão,

realizar as operações desejadas, e na hora de reproduzir novamente, aplicar o inverso da função que irá devolver o tamanho e informações originais do áudio.

A ferramenta responsável por aplicar essa função, é chamada de *Encoder* (ou codificador), enquanto sua outra metade é o *Decoder* (ou decodificador). Existem diversos padrões de compressão de sinais acústicos, como o "*Motion Picture Experts Group*" ou MPEG, e essa capacidade de representar dados de maneira mais enxuta, será melhor explorada ao longo deste projeto.

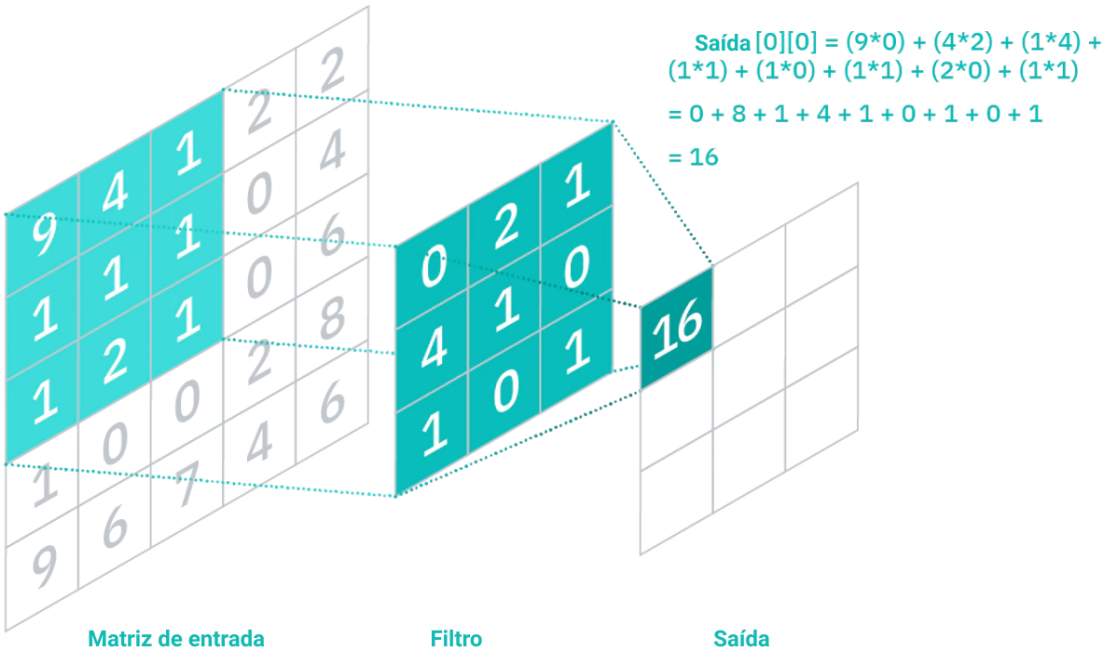
2.1.5 Machine Learning e Deep Learning

Métodos de *Machine Learning* [6] (ZHANG, 2020) são um ramo de Inteligência Artificial com foco em utilizar algoritmos e modelos matemáticos para simular a forma com a qual humanos aprendem, com cálculos para melhorar a precisão dos resultados ao longo das iterações. Dentro do contexto de análise de dados, essas estratégias são altamente utilizadas para os mais diversos casos, desde tomada de decisão em supercomputadores até monitoração de comportamento em jogos online.

Dentro desse contexto, uma subdivisão é conhecida como *Deep Learning* (GOODFELLOW, et. al. 2016), e consiste na utilização de redes neurais com três ou mais camadas, ainda com o objetivo de representar, matematicamente, a forma como o cérebro humano pensa. Para o caso de uso deste projeto, iremos priorizar os modelos conhecidos como **rede neural convolucional (CNN)**, que contém em sua arquitetura, uma ou mais camadas que aplicam a operação matemática de convolução para gerar as matrizes de multiplicação. Um exemplo de CNN pode ser visualizado na Figura 3, na qual reduz-se o tamanho da entrada por meio da aplicação de um filtro. Nesse caso, a nova saída consiste em aplicar a convolução entre um pedaço da entrada e o filtro (ou *Kernel*), para obter uma saída na qual o valor na posição (0,0) da matriz de saída, representa outros 9.

[6] https://www.ibm.com/cloud/learn/machine-learning

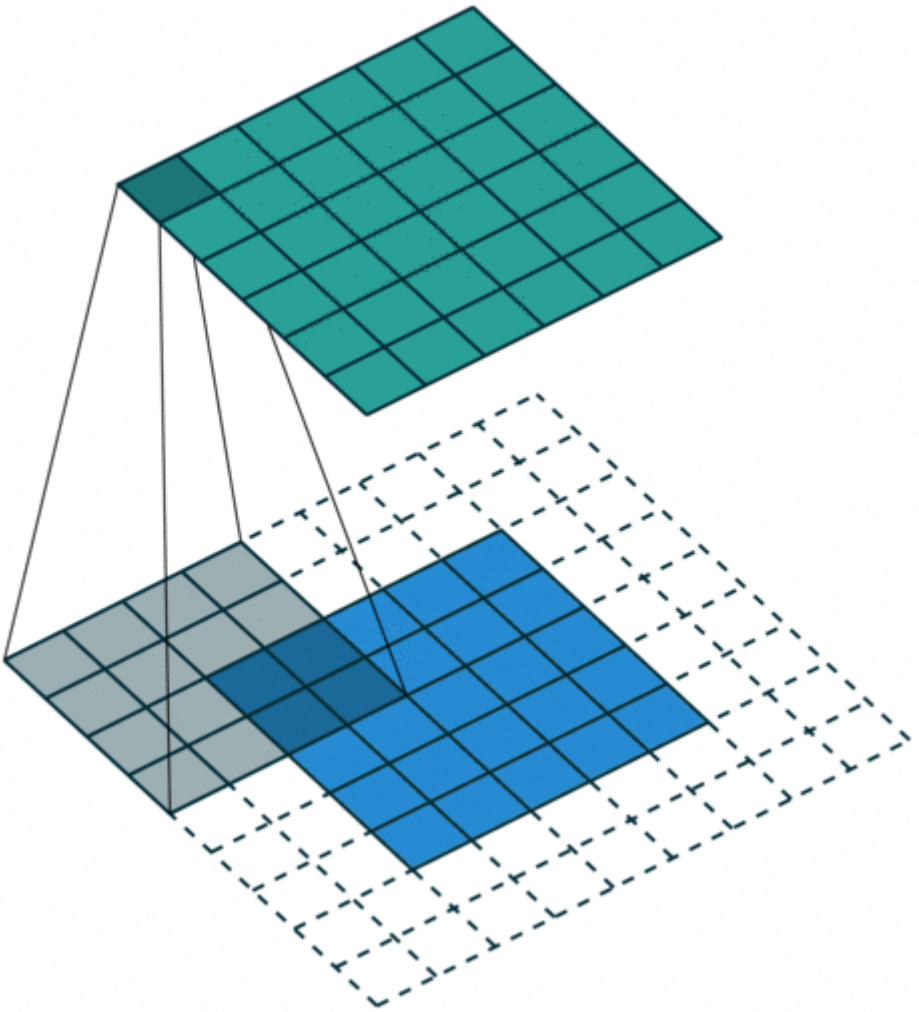
Figura 3: Exemplo de convolução



Fonte: Adaptado de www.ibm.com/cloud/learn/convolutional-neural-networks

Após esse primeiro processo, um novo pedaço da matriz de entrada será passado pelo filtro, e a distância (ou número de casas) que o *Kernel* move para pegar os próximos valores, é chamado de *Stride*. Quanto maior o *Stride*, menor a saída (pois serão puladas algumas combinações para o filtro). Ainda, nos casos os quais se deseja que os extremos da matriz passem pelo *Kernel* mas em diferentes posições, é comumente aplicado uma técnica de *Padding*, que adiciona valores ao redor da entrada para que os extremos originais possam ser manipulados, conforme exemplifica a Figura 4, na qual os espaços em branco são os valores adicionados.

Figura 4: Convolução com padding



Fonte: github.com/vdumoulin/conv_arithmetic

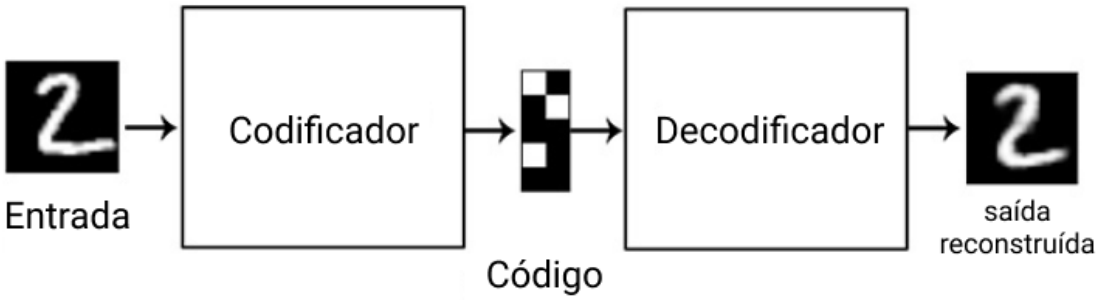
Dentro do contexto do *ASVspoof*, a utilização de modelos de **CNN** é amplamente difundida devido a necessidade de se manter consistência com relação ao tempo, juntamente da praticidade de representar modelos em uma menor dimensão.

2.1.6 Autoencoders

Uma das aplicações de redes de deep learning utilizando modelos de CNN, são os chamados autoencoders. Eles consistem em uma rede treinada para tentar copiar a entrada, contendo internamente uma camada intermediária conhecida como **código** (ou *code*) utilizada para representar o original de maneira reduzida.

Sua estrutura, então, consiste em: *Encoder -> Code -> Decoder*, conforme exemplificado na Figura 5 que utiliza uma imagem para melhor visualização (vale ressaltar que para arquivos de som, o princípio é o mesmo).

Figura 5: Modelo de autoencoder



Fonte: Adaptado de blog.curso-r.com/posts/2017-06-26-construindo-autoencoders

Apesar de, num primeiro momento, um autoencoder possa parecer irrelevante, eles podem ser utilizados para diversas finalidades, como remoção de ruídos, diminuição de dimensões, ou o que será explorado nesse projeto: a representação em menor escala de um arquivo de áudio.

2.2 Estado da arte

A primeira edição do *ASVspoof* discutiu majoritariamente os *Spoofs* de *Speech Synthesis* e *Voice Conversion*. Já na segunda, o principal tópico foi o *Replay*, dado que este é considerado o que confere a maior ameaça à segurança dos dados, pois é provavelmente a forma de ataque mais prolífica entre as opções. Nesse novo cenário, as análises se voltaram a avaliar o uso de pequenas senhas que são usadas por ASV texto-dependente. Este segundo evento foi essencial para evidenciar a maior dificuldade no estabelecimento de estratégias para o *replay* perante o speech synthesis e o voice conversation (KINNUEN *et al.,* 2017*)*.

Visando mapear os avanços e analisar os resultados obtidos até então, PATIL *et. al.* (2018), publicou uma pesquisa a respeito de ataques de repetição até o presente momento. Os principais bancos de dados validados foram o *AVspoof Database* e *ASVspoof Challenge 2017*, contendo amostras geradas/gravadas com diferentes métodos e dispositivos, além de prover dados para treinamento, desenvolvimento e avaliação. Ainda, como método de avaliação de performance, o principal utilizado é o "*Equal Error Rate (EER)"*, que consiste em balancear os possíveis erros de um sistema de ASV dadas as necessidades da aplicação. Um menor EER representa um sistema mais seguro. Essas falhas podem ser:

- **False Acceptance Rate (FAR):** Amostras de ataques ou *Spoofs* são aceitas pelo sistema;

- **False Rejection Rate (FRR):** Amostras reais/humanas são rejeitadas.

Já para as abordagens utilizadas, o sistema base utilizado para o *ASVspoof 2017* contava com *Constant Q Cepstral Coefficients (CQCC)* como classificadores e coeficientes do tipo 30-DCT e 90-D para a extração de características. Os resultados obtidos, porém, não performaram tão bem para ataques do tipo *replay*. O principal classificador utilizado pelos trabalhos avaliados, visando reduzir ainda mais o EER, foi o GMM ou *Gaussian mixture models* (REYNOLDS, 2009). Para os conjuntos de características, além do *CQCC* e suas variações, estratégias para análise espectral em maiores frequências, *Instantaneous Frequency* (IF) e técnicas de CMVN também foram aplicadas para aumentar o desempenho dos sistemas.

Font *et al.* (2017) propôs uma abordagem utilizando a fusão de vários classificadores bases com regressão logística, focando em selecionar características discriminatórias para aprimorar os resultados e reduzir o EER. Os melhores valores obtidos, utilizando *features*

individuais foram de 3.85% para o conjunto de desenvolvimento e 11.49% para avaliação. Já para a fusão de subsistemas, o melhor resultado foi um EER de l2%. Observa-se, então, que é possível otimizar o desempenho de um sistema de ASV realizando um tratamento e refinamento de classificadores já existentes.

Utilizando uma estratégia baseada em *Deep Neural Network (DNN)* ao invés de GMM„ Nagarsheth *et al.* (2017) também aplicou o *CQCC* juntamente do *HFCC* (outro classificador com ênfase em regiões de frequências mais altas) para criar uma alternativa na detecção de ataques de *Replay* para sistemas dependentes e independentes de texto. Os resultados mostraram um EER de 3.2% no dataset de desenvolvimento e 11.5% para avaliação, também somando duas ou mais técnicas para diminuir o valor.

Ainda se apoiando na extração de características em bandas de alta frequência, Witkowski *et. al.* (2017) explora o fato de que amostras de ataques de regravação, passarem por mais conversores analógico-digitais em comparação com um usuário autêntico. Para as *features*, além do *CQCC,* foram selecionadas algumas para serem usadas especificamente nas frequências maiores como *IMFCC (*variação da *MFCC*, que é uma das mais utilizadas em análises de fala) e *LPCC.* Os resultados, porém, apresentaram uma melhoria de apenas 30% com relação ao sistema base, tendo um EER de avaliação de 17.31%.

Weicheng CAI *et al.,* (2017) também utiliza *CQCC* e GMM em sua abordagem, porém com o diferencial de realizar um tratamento prévio nos dados para melhor representar as situações de regravação juntamente de uma rede de *Representation Learning.* Com isso, obteve-se uma melhora sutil: um EER de 16.39% após a fusão dos resultados no dataset de teste. Em contrapartida, o pré-processamento dos dados foi realizado manualmente e , consequentemente, depende da atuação humana.

É apresentado no quadro 1, uma relação entre os trabalhos apresentados e as principais técnicas que serão abordadas ao longo do trabalho, evidenciando uma lacuna que pode ser preenchida com uma nova estratégia.

Quadro 1: Relação entre trabalhos e metodologias

	Font et al. (2017)	Nagarsheth et al. (2017)	Weicheng CAI et al., (2017)	Witkowski et. al (2017)
***Features* individuais**	✔	✖	✔	✖
Realiza pré-processamento dos dados	✖	✖	✔	✔
Utiliza modelos de CNN	✖	✔	✖	✔
Utiliza autoencoders	✖	✖	✖	✖

Fonte: Elaborado pelo autor

Portanto, observa-se que para a elaboração de técnicas de combate a ataques de replay e criação de sistemas de ASV, a utilização de vários métodos trabalhando em conjunto, contribuem para a redução do EER. Ainda, os trabalhos com melhor desempenho bruto, foram os que utilizaram de análises espectrais e extraíram características de zonas com maior frequência. Finalmente, a utilização de algoritmos de inteligência artificial, *Machine learning* ou representation learning foi fundamental e unânime em toda documentação encontrada.

Vale ressaltar que os sistemas base disponibilizados pelo *ASVspoof Challenge 2017* possuíam um EER de 24.77% e 30.60%, e somente as 9 melhores submissões do obtiveram um resultado final com EER menor que 20%. Dos demais participantes, 24 se mantiveram no intervalo de 20% a 30% e os 14 restantes acima de 30% (sendo o pior resultado 45.55%). A média ficou em 26.01%. Já para o *ASVspoof Challenge 2019,* os dois sistemas de base para o cenário de PA, possuem um EER de 11.04% e 13.54%.

3 Desenvolvimento

O trabalho será realizado em etapas: Primeiro os dados do banco de dados disponibilizado pelo *ASVspoof,* serão avaliados e normalizados para que, em seguida, seja possível extrair suas características por meio de um modelo de *autoencoder*. Então, os valores obtidos serão aplicados em outro modelo para gerar gerar as predições que permitirão criar a relação do *EER* e, finalmente, validar a estratégia proposta.

3.1 Seleção dos dados

Os arquivos de sinais acústicos que serão utilizados, podem ser obtidos gratuitamente pelo site dos desenvolvedores do desafio[7], e já são separados entre PA e LA. Como o objetivo deste trabalho é propor uma estratégia voltada para ataque de regravação, somente o conjunto de PA será utilizado. Somando um total de aproximadamente 240000 registros, os dados possuem tamanhos que variam entre 1 e 7 segundos, e consistem em diferentes locutores falando diversas frases (o que é essencial uma vez que o sistema que será proposto será independente de contexto). Isso se aplica tanto para as amostras de *spoof* quanto para os sinais acústicos genuínos.

Os dados disponibilizados serão separados em 3 grupos: treino, desenvolvimento e avaliação, com quantias de 54000. Eles serão utilizados respectivamente para: treinar os modelos que serão descritos adiante, testar os resultados obtidos pelo treinamento e, por fim, validar o modelo comparando com o conjunto anterior. Ainda, todos os arquivos possuem nomes no formato “Tipo-de-acesso_grupo_id”, onde:

- Tipo-de-acesso: PA ou LA
- grupo: D = desenvolvimento, E = avaliação ou T = treino
- id: valor inteiro sequencial de 7 dígitos

Por exemplo: PA_D_0000010 se trata de um arquivo de PA, do grupo de desenvolvimento e id = 10. A Tabela 1 resume a separação dos dados e tipos de locutores presentes em cada um dos *datasets*.

[7] https://www.asvspoof.org/

Tabela 1: Separação dos dados por grupo e tipo

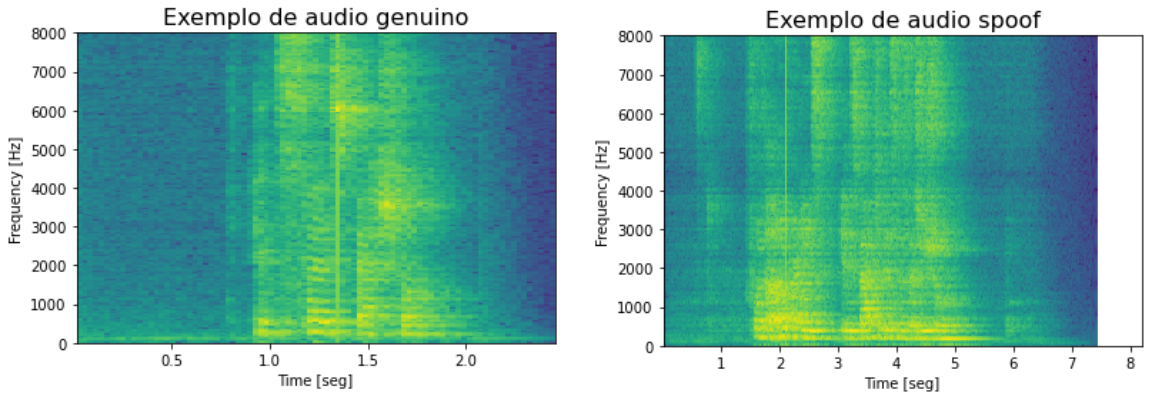
	Locutores		Tipo	
Dataset	Homem	Mulher	Genuino	*Spoof*
Treino	8	12	5400	48600
Desenvolvimento	8	12	5400	48600
Avaliação	8	12	18090	35910

Fonte: Elaborado pelo autor

3.1.1 Tratamento dos dados

Apesar do banco de dados já manter uma grande consistência, com todos os arquivos em formato ".flac" e com taxa de amostragem de 16000, para os modelos que serão apresentados, é necessário que todas as entradas possuam um mesmo tamanho, o que gera a necessidade de realizar um pré-processamento. Porém, apenas reduzir o tamanho de todos os arquivos para o menor, ocasionaria uma perda de informações muito grande, uma vez que em diversos casos, as falas acontecem principalmente no meio dos arquivos. A Figura 6 exemplifica dois desses casos por meio do espectrograma dos sinais acústicos (quanto mais claro, maior a presença de determinada frequência no instante selecionado).

Figura 6: Espectrograma de áudio spoof e genuíno

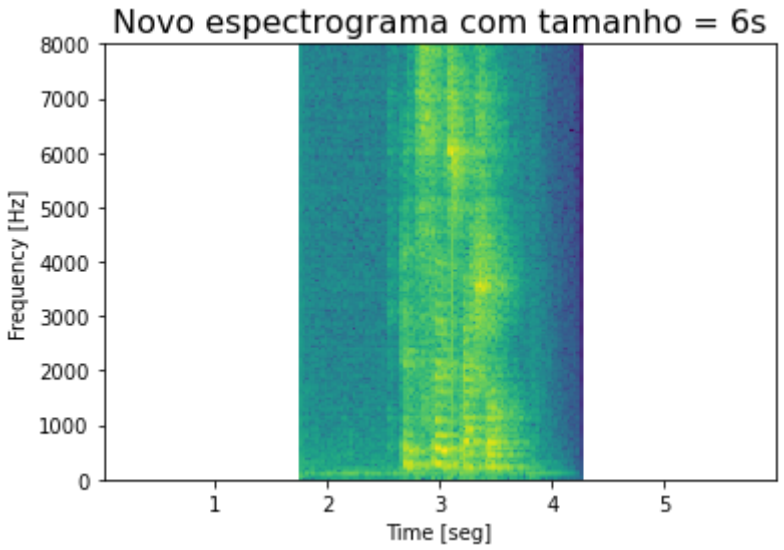


Fonte: Elaborado pelo autor

As falas se iniciam respectivamente em aproximadamente 1 e 1.5 segundos e se encerram em 2 e 5.5 segundos. Logo, ao limitar o tamanho dos arquivos, é necessário levar isso em consideração para evitar a existência de arquivos irrelevantes que possam gerar outliers durante as análises.

Para isso, todos os sinais acusticos passaram por um processo de *zero-padding*, que consiste em adicionar uma quantia igual de zeros tanto no início quanto no final dos arquivos menores do que um limite de tempo predeterminado (para este projeto, tomaremos um valor máximo de 6 segundos). Com isso, o mesmo áudio de aproximadamente 2.5 segundos utilizado no exemplo acima, após esse processo, passará de 40000 para 96000 amostras (seu novo espectrograma pode ser visualizado na Figura 7).

Figura 7: Espectrograma após zero-padding

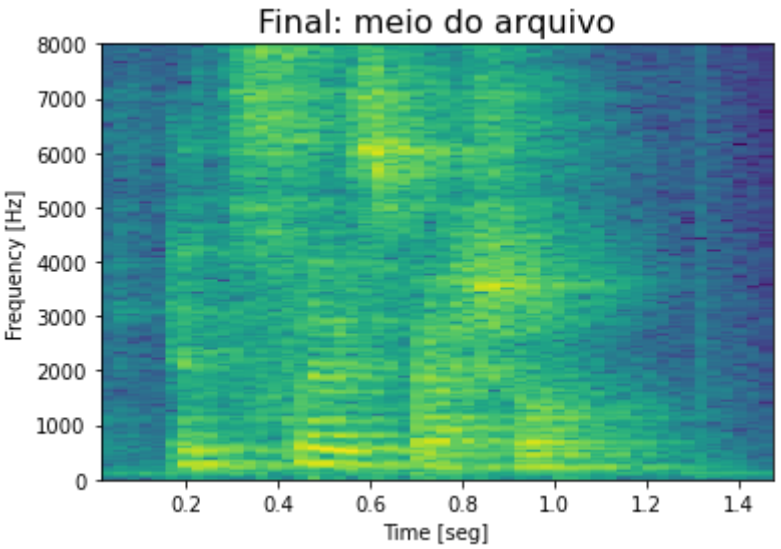


Fonte: Elaborado pelo autor

Dessa forma, torna-se possível extrair sempre uma porção de mesmo tamanho do meio de cada áudio, garantindo assim que todas as entradas possuem informações úteis. Para este

projeto, utilizaremos 1.5 segundos de cada áudio (24000 amostras) e o final de cada arquivo ficará como na Figura 8.

Figura 8: Espectrograma após pré-processamento



Fonte: Elaborado pelo autor

Finalmente, para evitar possíveis *outliers*, os vetores contendo os dados de cada áudio, são normalizados e convertidos para Tensores que serão utilizados pelos modelos de deep learning. Essa última etapa ainda adiciona duas dimensões aos dados, que serão utilizadas pelos *batches* e canais de entrada da rede.

3.2 Seleção das características

Conforme citado anteriormente, a classificação será realizada por meio de uma rede de *Deep Learning* que necessita de um conjunto de características para decidir se determinada entrada se encaixa como *Spoof* ou genuíno. Os diversos trabalhos analisados, utilizam uma ou várias features conhecidas, aqui nós iremos criar as features por meio da utilização de um **autoencoder convolucional.**

Uma vez que um *Autoencoder* é uma rede que aprende a recriar uma entrada, ao inserir um áudio no modelo em questão, obtém-se na camada intermediária um código de

menor tamanho que representa as principais características do objeto estudado. Ainda, quando trabalhamos com sinais sonoros, é necessário manter a relação de dimensionamento para que, na hora da reconstrução, a informação seja colocada na ordem certa sem perder o tempo. Com isso, a utilização de camadas de convolução se torna uma opção viável e atraente, devido a sua capacidade de reduzir o tamanho da anterior mantendo a ordem das informações.

Considerando esses fatores, a abordagem escolhida para extrair características nesta estratégia, consiste em definir um modelo de *Autoencoder* capaz de receber as entradas de tamanho 24000 (ou seja, 1.5 segundos de áudio com amostragem de 16000) e devolver um áudio recriado de mesmo tamanho. No entanto, ao invés de utilizar todos os arquivos de treinamento, passar somente aqueles referentes a *Spoofs*. Em seguida, ir reduzindo o tamanho das camadas intermediárias e balanceando os valores do *Stride* e *Kernel* de modo a não comprometer a precisão.

Finalmente, após construir um modelo capaz de identificar as principais características dos áudio de *Spoofing*, passamos novamente os arquivos (tanto de treino quanto avaliação e desenvolvimento), porém dessa vez, para cada um deles, interceptamos o fluxo de execução e retiramos como saída não a reconstrução final, mas sim o código da camada intermediária. Assim, obtemos o conjunto de características referente aquela entrada.

Para os sinais acústicos genuínos, repetimos o mesmo processo. Vale ressaltar a importância de separar um modelo para cada tipo, pois somente assim é possível obter os vetores de características de um grupo específico.

Finalizamos, então, a etapa de extração de características com duas listas de vetores, nas quais cada elemento está associado a um áudio do banco de dados, contendo os valores obtidos por meio da codificação de um trecho de cada áudio.

3.3 Rede de classificação

Com os classificadores prontos, é possível preparar o terceiro e último modelo de machine learning, dessa vez responsável por de fato classificar as entradas entre *Spoof* ou genuíno. Para isso, novamente utilizaremos de camadas de convolução (CNN), pois é necessário manter consistência com a forma que as características foram extraídas nos *Autoencoders*, permitindo uma comparação entre códigos que faz sentido.

Essa rede recebe como entrada para treinamento os trechos dos sinais acusticos juntamente das listas de características contendo o vetor respectivo. Já para predição, apenas o trecho do áudio que deseja tentar identificar.

Entretanto, o modelo de predição não se isenta do pré-processamento utilizado anteriormente para selecionar os trechos de áudio de maior relevância. Logo, todos os processos são aplicados: desde o *Padding* e corte, até a normalização e adição de dimensões aos tensores.

3.4 Tecnologias utilizadas

Todas as manipulações e códigos escritos para a implementação e testes contidos nesse projeto, foram escritos utilizando a linguagem de programação Python, devido a sua robustez, ampla documentação e permitir que todas as etapas pudessem ser realizadas dentro de um mesmo ambiente. Ainda, para os modelos de deep learning, as bibliotecas do PyTorch[8] e CUDA[9] serviram para otimizar todos os processos de treinamento e validação dos modelos. Finalmente, o ambiente do Google Colab Pro foi utilizado para executar os códigos desenvolvidos.

3.5 Considerações finais

Neste capítulo, foram apresentadas as etapas envolvidas no desenvolvimento de uma estratégia de identificação de ataques de PA do tipo regravação. Os dados utilizados foram obtidos na edição mais recente do *ASVspoof* , e a implementação dos modelos propostos foi realizada utilizando algumas das bibliotecas de manipulação de dados e *Deep Learning* mais bem avaliadas atualmente.

Os modelos apresentados, apesar de parecerem simples, possuem uma camada de complexidade implícita. Sinais de áudio digital, além de possuírem uma quantidade bruta de dados altíssima, são muito sensíveis a perdas de informação devido a sua natureza de

[8] https://pytorch.org/
[9] https://developer.nvidia.com/cuda-zone

representar ondas analógicas. Um valor *Batch* errado que o *Autoencoder* receba, pode ocasionar uma compressão anormal de todas as frequências abaixo de um limite, “abafando” os sons ou gerar anomalias como ruídos e chiados. Dessa forma, para que fosse possível diminuir efetivamente o tamanho das camadas de convolução dentro dos modelos, foram necessárias extensas rotinas de treinamento, juntamente de testes com diversas combinações de conFigurações para as camadas.

Obtém-se, portanto, uma estratégia baseada em *Deep Learning* capaz de codificar as entradas, separar as principais características e comparar com os valores previamente obtidos para validar se a entrada do sistema de ASV é genuína ou um *Spoof*.

4 Avaliação experimental

Neste capítulo, será apresentada toda a metodologia de testes utilizada para validar a estratégia proposta anteriormente, com os resultados obtidos sendo discutidos no final.

4.1 Estratégias de avaliação

Os, modelos descritos anteriormente, serão avaliados utilizando duas principais métricas: o *Equal Error Rate* (principal metodologia dentro do *ASVspoof Challenge*) e o t-DCF , que além de considerar os parâmetros de falso positivo e negativo, também leva em consideração o tempo gasto em cada um dos casos. Ambos irão considerar a acurácia do sistema, que consiste na porcentagem de acertos durante o processo de classificação.

Todos os treinamentos foram realizados com 150 *Epochs* (iterações na qual o sistema altera os pesos para melhor prever os resultados), utilizando Adam como método de otimização e algoritmo de ativação do tipo Tanh, uma vez que os valores recebidos como entrada, após o pré-processamento, encontram-se sempre entre -1 e 1. Além disso, para todos os grupos de dados, foram separados 80% para treinamento e 20% para validação, garantindo sempre que a mesma quantia de *Spoofs* e genuínos fossem utilizados durante o treino.

4.2 Resultados e discussões

Os resultados de *Equal Error Rate* (em porcentagem) e t-DCF obtidos, estão apresentados na Tabela 2

Tabela 2: Resultados

Modelo	Dataset	min-tDCF	EER (%)
CNN	Treino	0,341	20.482
CNN	Desenvolvimento	0,363	20.414
CNN	Avaliação	0,319	19.601

Fonte: Elaborado pelo autor

Observa-se que os valores de EER obtidos encontram-se pouco acima dos sistemas de base disponibilizados, assim como o t-DCF. o que indica que é possível extrair as características de *spoofing* por meio de autoencoders. Ainda, quando comparado com outros resultados submetidos ao *ASVspoof 2019*, a estratégia proposta se encaixaria junto ao modelo de posição 45, como comparado na Tabela 3.

Tabela 3: Comparação de resultados

Posição pelo ASVspoof	min-tDCF	EER (%)
#43	2.958	12.53
#44	0.3017	13.54
Modelo proposto	0,319	19.601
#45	0.3641	13.85
#46	0.4269	21.25

Fonte: Adaptado de Todisco et. al., 2019

Ainda, visando melhorar os resultados, uma segunda série de testes foi realizada, na qual os sistemas foram treinados individualmente para locutores homens e mulheres, mantendo todas as conFigurações anteriores. Houve, então, uma leve melhora bruta no desempenho, conforme observado na Tabela 3

Tabela 4: Resultados após separação

Modelo	Dataset	min-tDCF	EER (%)
CNN	Treino	0,357	18.721
CNN	Desenvolvimento	0,389	18.156
CNN	Avaliação	0,377	17.245

Fonte: Elaborado pelo autor

Contudo, apesar de um EER melhor, os valores do t-DCF mantiveram-se similares, o que se dá pelo tempo extra gasto para identificar o tipo de locutor antes de verificar se é *Spoof* ou genuíno.

4.3 Considerações finais

Os experimentos realizados validam a estratégia proposta, considerando que os modelos são capazes de operar em cima dos valores de entrada, extrair suas características e classificar com uma precisão boa para os padrões do *ASVspoof Challenge*. Ainda, a tentativa de separar os tipos de locutores, apesar de entregar resultados um pouco melhores, ainda carece de maior aprofundamento e será abordada novamente na secção referente a trabalhos futuros.

Considera-se então, duas estratégias que podem ser aplicadas para sistemas de ASV:

- A primeira utilizando o modelo de **CNN** para tomada de decisão considerando características extraídas por meio de um autoencoder
- A segunda que, primeiro decide se o locutor é homem ou mulher e, em seguida, repete o passo do anterior.

Contudo, quando se pensa em utilizar essas estratégias em cenários reais, alguns fatores devem ser considerados, como o tempo e poder de processamento necessários para cada uma, uma vez que a segunda custa mais que a primeira apesar dos melhores resultados.

Finalmente, é importante ressaltar que os modelos finais foram treinados considerando 150 *Epochs* devido a complexidade e altos custos computacionais devido ao tamanho das entradas. Dessa forma, é possível que os resultados finais com os mesmos modelos ainda

possam ser melhorados se aumentarmos o número de rodadas de treinamento e aplicar técnicas de otimização por TPU e paralelismo.

5 Conclusão

As evoluções dentro do contexto de ASV nos últimos anos não foram poucas: em 2019, o melhor classificado do *ASVspoof Challenge* possuía um EER de metade do primeiro colocado em 2017. Quando consideramos o aumento da popularidade desse tipo de problema juntamente de todas as publicações realizadas, torna-se evidente a tendência desses valores brutos diminuírem ainda mais à medida que novas técnicas mais sofisticadas surgem. Ainda, a existência de um banco de dados próprio para esse tipo de situação, que já cobre tanto situações de PA quanto LA, facilita o ingresso de novos pesquisadores na área.

Com isso, esse projeto buscou agregar com uma estratégia para não apenas classificar os valores baseados em *Features* já amplamente difundidas, como também extrair novos valores com base nos dados disponibilizados, e os resultados obtidos mostram que não apenas foi possível, como existem margens para melhoria e otimização de diversas maneiras diferentes.

5.1 Trabalhos futuros

Conforme citado anteriormente, alguns pontos ainda podem ser melhorados em trabalhos futuros, sendo eles:

1. *Ensemble* de classificadores: Diversos trabalhos da literatura utilizam mais de uma *Feature* para realizar a classificação dentro dos modelos. Logo, torna-se atraente testar a junção do classificador aqui utilizado com outros já existentes e difundidos em projetos do mesmo tipo.
2. Otimização e paralelismo: Juntamente da proposta anterior, paralelizar os modelos de treinamento visando aumentar a quantidade de *Epochs* sem a necessidade de uma excessiva quantia de poder computacional, pode ser uma das maneiras de melhorar a estratégia aqui definida.

Referências

A. PATIL, Hemant; R. KAMBLE, Madhu. **A Survey on Replay Attack Detection forAutomatic Speaker Verification (ASV) System**, 2018.

DELGADO, Héctor; TODISCO, Massimiliano; SAHIDULLAH, Md; EVANS, Nicholas; KINNUNEN, Tomi; LEE, Kong Aik; YAMAGISHI, Junichi. **ASVspoof 2017 Version 2.0: meta-data analysis and baseline enhancements**, 2017. DOI: 10.7488/ds/298. Disponível em: http://dx.doi.org/10.7488/ds/298.

FONT, Roberto; ESPÍN, Juan M.; CANO, María José. **Experimental analysis of features for replay attack detection-Results on the ASVspoof 2017 ChallengeProceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH**International Speech Communication Association, , 2017. a. DOI: 10.21437/Interspeech.2017-450.

FONT, Roberto; ESPÍN, Juan M.; CANO, María José. **Experimental analysis of features for replay attack detection-Results on the ASVspoof 2017 ChallengeProceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH**International Speech Communication Association, , 2017. b. DOI: 10.21437/Interspeech.2017-450.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. (2016). ***Deep Learning.*** MIT Press, 2016.

HANSEN, John H. L.; HASAN, Taufiq. **Speaker recognition by machines and humans: A tutorial reviewIEEE Signal Processing Magazine**Institute of Electrical and Electronics Engineers Inc., , 2015. DOI: 10.1109/MSP.2015.2462851.

KAMBLE, Madhu R.; SAILOR, Hardik B.; PATIL, Hemant A.; LI, Haizhou. Advances in anti-spoofing: From the perspective of ASVspoof challenges. **APSIPA Transactions on Signal and Information Processing**, *[S. l.]*, v. 9, 2020. DOI: 10.1017/ATSIP.2019.21.

KHALED M. ALHAWITI. **Advances in Artificial Intelligence Using Speech RecognitionApplied Data Analytics: Principles and Applications** River Publishers, , 2020. DOI: 10.1007/978-0-387-77592-0_13.

KINNUNEN, Tomi; EVANS, Nicholas; YAMAGISHI, Junichi; LEE, Kong Aik; TODISCO, Massimiliano; DELGADO, Héctor. **ASVspoof 2017: Automatic Speaker Verification Spoofing and Countermeasures Challenge Evaluation Plan ***. [s.l: s.n.]. Disponível em: http://www.spoofingchallenge.org/.

KINNUNEN, Tomi; SAHIDULLAH, Md; DELGADO, Héctor; TODISCO, Massimiliano; EVANS, Nicholas; YAMAGISHI, Junichi; LEE, Kong Aik. **The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detectionProceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH** International Speech Communication Association, , 2017. b. DOI: 10.21437/Interspeech.2017-1111.

KINNUNEN, Tomi; SAHIDULLAH, Md; DELGADO, Héctor; TODISCO, Massimiliano; EVANS, Nicholas; YAMAGISHI, Junichi; LEE, Kong Aik. **The ASVspoof 2017 Challenge: Assessing the Limits of Replay Spoofing Attack Detection**, 2017. Disponível em: https://www.iso.org/standard/53227.html.

L.R. RABINER; R.W SHAFER. **Digital Processing of Speech Signals**. [s.l: s.n.].
NAGARSHETH, Parav; KHOURY, Elie; PATIL, Kailash; GARLAND, Matt. **Replay attack detection using DNN for channel discriminationProceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH** International Speech Communication Association, , 2017. DOI: 10.21437/Interspeech.2017-1377.

PAN, Davis Yen. (1993). **Digital Audio Compression.** *Digital Technical Journal*.

PRADHAN, Swadhin; SUN, Wei; BAIG, Ghufran; QIU, Lili. Combating Replay Attacks Against Voice Assistants. **Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies**, *[S. l.]*, v. 3, n. 3, p. 1–26, 2019. DOI: 10.1145/3351258.

RAMOS, Bruno Thadeu Reis. **AS SEIS CANÇÕES TROVADORESCAS DE FRUCTUOSO VIANNA: Aspectos intertextuais e perspectivas interpretativas para voz de contratenor na canção de câmara brasileira**. 2013. Belo Horizonte, 2013.

REYNOLDS, Douglas A. **Gaussian mixture models**. Encyclopedia of biometrics, v. 741, p. 659-663, 2009.

SARFJOO, Seyyed Saeed; WANG, Xin; HENTER, Gustav Eje; LORENZO-TRUEBA, Jaime; TAKAKI, Shinji; YAMAGISHI, Junichi. **Transformation of low-quality device-recorded speech to high-quality speech using improved SEGAN model**, 2019. DOI: 10.7488/ds/1994. Disponível em: http://arxiv.org/abs/1911.03952.

SHAYEB, Ismail; ASAD, Naseem; ALQADI, Ziad; JABER, Qazem. Evaluation of speech signal features extraction methods. **Journal of Applied Science, Engineering, Technology, and Education**, *[S. l.]*, v. 2, n. 1, p. 69–78, 2020. DOI: 10.35877/454ri.asci2151.

TODISCO, M., Wang, X., Vestman, V., Sahidullah, M., Delgado, H., Nautsch, A., Yamagishi, J., Evans, N., Kinnunen, T., & Aik Lee, K. (2019). ***ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection*.**

Yang, Y., Wang, H., Dinkel, H., Chen, Z., Wang, S., Qian, Y., & Yu, K. (2019). **The SJTU robust anti-spoofing system for the ASVspoof 2019 challenge**. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, *2019-September*, 1038–1042. https://doi.org/10.21437/Interspeech.2019-2170

WANG, Xin et al. ASVspoof 2019: a large-scale public database of synthetized, converted and replayed speech. **Computer Speech and Language**, *[S. l.]*, v. 64, 2020. DOI: 10.1016/j.csl.2020.101114.

WITKOWSKI, Marcin; KACPRZAK, Stanisław; ZELASKO, Piotr; KOWALCZYK, Konrad; GAŁKA, Jakub. **Audio replay attack detection using high-frequency featuresProceedings of the Annual Conference of the International Speech Communication Association,**

INTERSPEECH International Speech Communication Association 2017. DOI: 10.21437/Interspeech.2017-776.

WU, Zhizheng; YAMAGISHI, Junichi; KINNUNEN, Tomi; HANILÇI, Cemal; SAHIDULLAH, Mohammed; SIZOV, Aleksandr; EVANS, Nicholas; TODISCO, Massimiliano; DELGADO, Héctor. **ASVspoof: The automatic speaker verification spoofing and countermeasures challengeIEEE Journal on Selected Topics in Signal Processing** Institute of Electrical and Electronics Engineers Inc., , 2017. DOI: 10.1109/JSTSP.2017.2671435.

ZHANG XD. (2020) **Machine Learning**. In: A Matrix Algebra Approach to Artificial Intelligence. Springer, Singapore. https://doi.org/10.1007/978-981-15-2770-8_6