Abordagem de Simulação de Ruídos para Avaliação do Reconhecimento de Locutores em Ambientes Ruidosos

Vinícius Almeida dos Santos
viniciusas@edu.univali.br
Laboratório de Inteligência Aplicada - LIA
Laboratory of Embedded and Distributed Systems - LEDS
Universidade do Vale do Itajaí - UNIVALI
Itajaí, Brasil

Anita Maria da Rocha Fernandes
anita.fernandes@univali.br
Laboratório de Inteligência Aplicada - LIA
Universidade do Vale do Itajaí - UNIVALI
Itajaí, Brasil

Wemerson Delcio Parreira
parreira@univali.br
Laboratory of Embedded and Distributed Systems - LEDS
Universidade do Vale do Itajaí - UNIVALI
Itajaí, Brasil

ABSTRACT

There is frequent noise that affects the performance of voice recognition and speaker recognition algorithms in a real-world environment. For voice and speaker recognition systems to be more robust, these noises must not interfere in a harmful way, causing errors in understanding commands. To evaluate signal degradation and speaker recognition when exposed to real-world environments, we explore a reverberation noise simulation environment using a specific library in this work. We tested a speaker recognition model with i-vectors and Probabilistic Linear Discriminant Analysis (PLDA). We have analyzed the impact of noise in conjunction with reverberation on its error rate. Results based on Monte Carlo simulation showed that, for the tested cases, the noise set with reverberation worsened the recognition rate by up to 24,43%.

KEYWORDS

Reverberation Models, Speaker Recognition, Noise Models, Datasets.

1 INTRODUÇÃO

Atualmente, há diversos pontos negativos no uso de senhas e códigos identificadores pessoais, como CPF (Cadastro de Pessoa Física) e RG (Registro Geral). Além da falta de políticas adequadas de senha, esses meios de autenticação tradicionais são vulneráveis, podendo ser esquecidos, perdidos ou roubados [1]. Outras alternativas para autenticação envolvem a captação de traços biométricos, mas, normalmente dependem de contato com outros indivíduos ou do uso de equipamentos custosos, trazendo outros tipos de problemas. Uma dessas alternativas para autenticação é por meio do reconhecimento de características da voz.

Sistemas de reconhecimento de voz podem ser úteis em diversas aplicações, tais como, atendimento ao cliente, busca de informações e integrações com sistemas. O potencial desse tipo de autenticação está no fato de ser um método não invasivo. Uma outra aplicação pode ser encontrada nos *smartphones* e nos assistentes pessoais de voz, que possuem inúmeros comandos, como responder mensagens, definir alarmes e fazer ligações [2]. Entretanto, para comandos que realizam ações sensíveis, como transferências bancárias, já implementadas no assistente da Google, por exemplo, ainda é necessário desbloquear ou autenticar-se fisicamente para executá-los [3].

De acordo com Nguyen et al. [4], o reconhecimento de voz em ambientes reais é um desafio. Isso se deve ao fato que, em ambientes reais, existe a frequente presença de ruídos que afetam o desempenho de algoritmos de reconhecimento de voz e de locutor. Esses ruídos podem ser tanto periódicos (como interferência elétrica) quanto não periódicos (como vozes de outras pessoas). Apesar das diversas abordagens para filtragem e redução de ruído presentes na literatura, ainda não é abordado de maneira eficiente pelos assistentes pessoais [5, 6]. Para que os sistemas de reconhecimento de voz e locutor apresentem maior robustez, é importante que esses ruídos não interfiram de maneira nociva, causando erros no entendimento de comandos. Ainda há fragilidades dos sistemas de reconhecimento de locutor utilizados atualmente. Exemplo disso é o Google assistente, que descreve em sua documentação que, se uma pessoa com voz parecida for identificada, pode fazer ações como enviar e-mail, efetuar pagamentos e visualizar a agenda pessoal [7].

Comumente, o reconhecimento de locutor pela voz é feito em ambientes reais, que apresentam ruídos [8]. Esses ruídos possuem potencial para impactar negativamente qualquer tipo de processamento de sinais. Especialmente, para dispositivos *mobile*, a acurácia do processamento de voz é reduzida por conta dos ruídos [9].

Ruídos e reverberações são comuns em ambientes reais [10]. O uso de algoritmos de *speech enhancement* para pré-processamento dos sinais pode trazer melhorias para qualquer aplicação. Assim sendo, *speech enhancement* aplicado para filtragem de ruídos e reverberação pode ser benéfico para o reconhecimento de locutor em ambientes reais.

Atualmente, são utilizadas várias abordagens para lidar com ruídos no reconhecimento de voz e locutor. A abordagem mais comum é *data augmentation*, que consiste em treinar modelos com sinais corrompidos com ruídos [11]. O foco dessa abordagem é diversificar os dados disponíveis, de maneira que a rede não apresente sobreajuste.

A maioria dos trabalhos testa corrupção de sinais por ruídos apenas de maneira aditiva [12–15]. Contudo, a simples adição de ruídos não representa ambientes reais, pois os ruídos também são afetados pela reverberação de salas.

Atualmente, trabalhos que abordam situações reais utilizam bases de dados que já fornecem dados em condições de ambientes reais [12–15], como SITW [16] e NIST 2010 retransmitted [17]. Dessa forma, modelos obtém taxas de erro abaixo de 10% em seus experimentos, e podem ser consideravelmente piores no momento de utilização de sistemas por voz [18].

A reverberação pode ser simulada por meio de *Room Impulse Response* (RIR). RIR é a simulação da resposta de reverberação de uma sala, na qual os ruídos persistem e geram interferência na captura de sinais [19]. Neste trabalho propomos o uso da biblioteca `pyroomacoustics` [19] para simulação de ambientes ruidosos com RIR, observando os impactos da simulação na acurácia do reconhecimento de locutor.

Este trabalho está dividido como se segue. Na Seção 2 é apresentada a fundamentação teórica a respeito do ambiente de simulação e do algoritmo de reconhecimento de locutor usados neste trabalho. A Seção 3 explora as base dados usadas. A Seção 4 apresenta as especificações da análise realizada. Os resultados e as conclusões são apresentados respectivamente nas Seções 5 e 6.

2 AMBIENTES RUIDOSOS E O RECONHECIMENTO DE LOCUTOR

2.1 Simulação de Ambientes Ruidosos

Ambientes fechados estão sujeitos a reverberação. A reverberação consiste na reflexão das ondas sonoras em obstáculos, paredes, etc [20]. Conforme exemplificado na Figura 1, todas as fontes de uma sala são refletidos, o que cria incontáveis caminhos de ondas. Reverberação pode tornar os sons mais agradáveis ou dificultar o entendimento, dependendo da acústica da sala. Uma das maneiras de simular reverberação é com RIR.

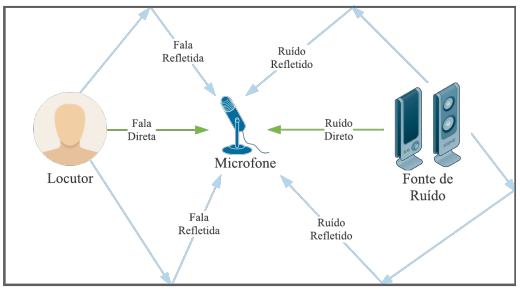

Figura 1: Propagação do som em uma sala.

Vários trabalhos da literatura já abordaram a problemática de RIR, até mesmo disponibilizando bibliotecas para uso [19]. Porém, a maioria não permite formatos de salas não retangulares, além de serem dependentes do uso de MATLAB. Com o propósito de fornecer um gerador de RIR acessível, de código aberto e flexível, Scheibler et al. [19] disponibilizaram a biblioteca Python `pyroomacoustics`. A biblioteca inclui um gerador RIR baseado em ISM (*Image Source Model*), permitindo mapear visualmente as posições de diversas fontes e microfones para simulação. Isso é feito com uma programação de alto nível, permitindo um código limpo e intuitivo.

Foram simulados ambientes ruidosos com a biblioteca `pyroomacoustics` [19]. Essa escolha se deve a diversos fatores, entre eles: (i) a biblioteca é utilizada para simulação de RIR (*Room Impulse Response*); (ii) é possível simular várias fontes, definindo a localização de cada uma; (iii) a reverberação é facilmente gerada, baseando-se nos parâmetros fornecidos; (iv) possui diversos materiais disponíveis com seus coeficientes de absorção. Dessa maneira, a biblioteca facilita a geração de simulações mais próximas de condições reais.

Em Taherian et al. [14] foram utilizadas as bases NIST SRE2010 [17] e NIST SRE2010 *retransmitted* [21]. A variação *retransmitted* é uma regravação com um auto-falante colocado em um ambiente altamente reverberante. A partir do NIST SRE2010 e fazendo uma simulação com condições similares às da base *retransmitted*, o trabalho gerou um ruído de *babble noise*, utilizando localizações aleatórias para locutores e tamanhos de salas arbitrários. As condições dos ambientes utilizadas foram as salas pequenas e médias da base OpenSLR. A base PRISM foi utilizada para o treinamento dos *i*-vector e *x*-vector.

Os *i*-vectors são a principal abordagem para um reconhecimento de locutor robusto [14]. A abordagem envolve uma redução na dimensionalidade das *features* acústicas e um classificador PLDA. Na maioria dos casos, *x*-vectors apresentam melhores resultados na presença de ruídos.

Nos resultados de Taherian et al. [14] a taxa de erro da simulação ficou bem abaixo da taxa de erro do experimento real. Essa divergência da realidade ocorreu mesmo com a simulação usando parâmetros equivalentes ao experimento real.

A Figura 2 ilustra uma abordagem para simulação de ambiente ruidoso com `pyroomacoustics`. A simulação de reverberação com RIR precisa dos parâmetros da sala, das fontes que estarão presentes no ambiente e a suas posições na sala. Dessa maneira, é possível gerar um sinal simulado mais próximo da realidade.

2.2 Reconhecimento de Locutor

A técnica de reconhecer uma pessoa pela sua voz é denominada reconhecimento automático do locutor pela voz [22]. O Reconhecimento Automático do Locutor, *Automatic Speaker Recognition* (ASR), pela voz é uma técnica que teve início a mais de 30 anos, e desde então, diversas técnicas foram apresentadas na literatura [11, 14, 23, 24].

Para avaliação do reconhecimento de locutor, foi utilizada a biblioteca Kaldi. Essa biblioteca é um *toolkit* feito em C++ que implementa a maioria das técnicas mais comuns de processamento de voz [25]. Uma vantagem da biblioteca é a existência de *recipes*, que consistem em conjuntos de *scripts* prontos cujo treinamento já foi executado e disponibilizado, facilitando sua aplicação em novos projetos, assim como melhora a aderência à biblioteca.

Neste trabalho é utilizada a *recipe* do Kaldi VoxCeleb/v1[1], que disponibiliza um modelo implementado e treinado por Snyder et al. [11]. Essa *recipe* utiliza métodos comuns para reconhecimento de locutor: *i*-vectors, e um classificador PLDA (*Probabilistic Linear Discriminant Analysis*).

São executados os seguintes estágios na fase de testes:

(i) **Geração do *Mel-Frequency Cepstral Coefficients* (MFCC)**, que extrai *features* para processamento do sinal;
(ii) **Geração do *Voice Activity Detection* (VAD)**, que detecta atividade de voz com o propósito de melhorar a taxa de erro;
(iii) **Extração dos i-vectors**, *features* específicas para locutores que servem como entrada para o classificador;

[1] https://github.com/kaldi-asr/kaldi/tree/master/egs/voxceleb/v1

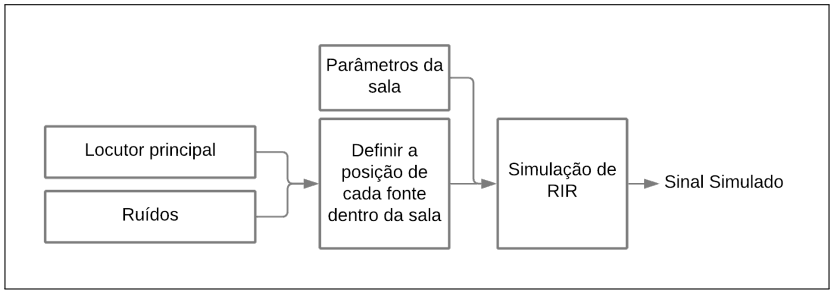

Figura 2: Fluxograma da Simulação de Ambiente Ruidoso.

(iv) ***Scoring* com PLDA**, classificando os locutores presentes em sinais; e
(v) **Cálculo do *Equal Error Rate* (EER)** e do MinDCF, as métricas para avaliação do modelo.

3 BASES DE DADOS

Para os experimentos realizados neste trabalho foram utilizadas 4 bases de dados. As seguintes bases estão descritas a seguir: VoxCeleb, Musan, NoiseX-92 e Coeficientes de Materiais.

3.1 VoxCeleb

VoxCeleb é uma base de dados aberta, permitindo usos acadêmicos e comerciais [26]. A base fornece informações áudio-visuais de locutores, com vários exemplos para cada locutor. Foi pensada especificamente para o reconhecimento de locutor, discriminando os locutores que pronunciaram cada frase.

A - Proveniência dos dados: Foi gerada a partir de vídeos enviados ao YouTube de entrevistas com celebridades [26]. Foram treinados modelos para identificar e classificar automaticamente trechos de falas nos vídeos, facilitando a criação de grandes bases de dados, sem a necessidade de classificação manual.

B - Especificações: Apresenta 148.642 amostras de 1.251 locutores no treino. Para teste, possui 4.874 amostras de áudio. 55% da base consiste em pessoas do sexo masculino, com variabilidades em etnia, sotaque, profissão e idade.

C - Aplicação: A base pode ser utilizada tanto para identificação quanto verificação de locutor. Neste trabalho, a base foi utilizada os testes de reconhecimento de locutor. Nos experimentos, foram utilizadas exclusivamente as amostras da separação de teste.

3.2 MUSAN

Snyder et al. [27] é uma base de dados que consiste em músicas, falas e ruídos. É aberto, permitindo usos acadêmicos e comerciais. Foi criado para a treinamento de modelos de VAD.

A - Proveniência dos Dados: Todas as amostras foram obtidas pela coleta em bases de domínio público. Os ruídos, que são mais relevantes para este trabalho, foram obtidos a partir de FreeSound.org e de SoundBible.com.

B - Especificações: Possui um total de $\sim$ 42 horas de música, $\sim$ 60 horas de fala e $\sim$ 6 horas de ruído. Os ruídos são diversificados, mas nenhum ruído com fala inteligível.

C - Aplicação: Neste trabalho, a base é utilizada pra o experimento global, corrompendo cada amostras de teste com um ruído aleatório da base.

3.3 NoiseX-92

É uma base de ruídos capturados em ambientes distintos [23]. Não está mais disponível para compra, mas é facilmente obtido por páginas que o disponibilizam[2].

A - Proveniência dos Dados: VARGA [23] selecionou ruídos de outra base, buscando condições que dificultassem o reconhecimento. Foram selecionados originalmente 8 ruídos, que apresentavam características não-estacionárias.

B - Aplicação: A base NoiseX-92 [23] é bastante comum para realizar testes de adição de ruídos em sinais de áudio [28, 29]. Neste trabalho, usamos 5 dos ruídos para avaliar seu impacto no reconhecimento de locutor.

3.4 Coeficientes de Absorção de Materiais

Existem várias maneiras de simular a reverberação de salas. Uma maneira simples que pode ser utilizada em `pyroomacoustics` [19] é definindo qual o material das paredes, chão e teto do ambiente. Cada material apresenta diferentes coeficientes de absorção, pois dependendo da frequência, o material pode apresentar comportamentos diferentes. A biblioteca disponibiliza[3] vários materiais para utilização.

Para a avaliação do reconhecimento de locutor, estabeleceram-se duas estratégias que serão descritas com mais detalhes na próxima seção. Uma estratégia denominada "experimento global", na qual os materiais são aplicados aleatoriamente. E outra estratégia denominada "experimentos com casos selecionados", em que cinco materiais com diferentes coeficientes de absorção foram avaliados.

4 ANÁLISE EXPLORATÓRIA

Neste trabalho, foram realizados 2 experimentos: (i) **global**, buscando a taxa de erro média para situações bastante diversificadas; e (ii) **com casos selecionados**, buscando as taxas de erro médias para algumas situações especificas.

Em ambos os experimentos, as seguintes especificações de simulação com `pyroomacousitcs`, foram utilizadas:

- classe *ShoeBox*, que facilita a criação de sala retangular;

[2] http://spib.linse.ufsc.br/noise.html
[3] https://pyroomacoustics.readthedocs.io/en/pypi-release/pyroomacoustics.materials.database.html

- parâmetro *ray_tracing* ativado, simulando a reverberação da sala;
- parâmetro *air_absoption* ativado, simulando a absorção do ar;
- sala com dimensões 10m × 10m × 5m;
- locutor sempre posicionado ao centro da sala;
- microfone sempre posicionado a 10cm do locutor;

Nos experimentos, foram utilizadas amostras de testes da base VoxCeleb [26].

4.1 Experimento Global

O experimento global serve para verificar a taxa de erro em casos com bastante diversificação. Dessa maneira, pode-se ter uma amostra mais fiel à realidade. Como ambientes reais possuem uma alta diversificação, optou-se por utilizar variáveis aleatórias para cada sinal.

Os resultados desse experimento são: (i) **sem corrupção**, utilizando a base de teste sem adulteração no sinal; (ii) **com ruído aditivo**, o método mais comum para a corrupção do sinal; (iii) **com reverberação**, simulando locutores falando em ambientes com diferentes níveis de reverberação; e (iv) **com reverberação e ruído**, simulando além de locutor, um ruído na sala.

Dessa maneira, para o resultado **com reverberação**, cada amostra de fala da base é adicionada a um ambiente com um material aleatório de `pyroomacoustics`. A seguir é feita a simulação e o sinal simulado é adicionado para uma nova base de dados.

Já para o resultado **com reverberação e ruído**, é adicionado um novo passo de adição de um sinal de ruído aleatório. O ruído tem sua amplitude normalizada para a mesma do sinal de voz e é adicionado em uma posição aleatória da sala. Esse fluxo resulta nos seguintes passos (para cada amostra de teste):

(1) criação do ambiente com material aleatório de `pyroomacoustics`;
(2) adição do locutor no centro da sala;
(3) seleção de um ruído aleatório da base MUSAN;
(4) normalização da amplitude do ruído;
(5) adição de ruído em uma posição aleatória da sala;
(6) execução da simulação; e
(7) adição à nova base de dados.

4.2 Experimento com Casos Selecionados

O objetivo deste experimento é analisar o comportamento da taxa de erro algumas situações específicas. São consideradas algumas situações reverberantes com a adição de ruído. Dessa maneira é possível observar a relação entre reverberação de ambiente e ruído.

Foram selecionados 5 materiais da biblioteca `pyroomacoustics` e 5 ruídos da base NoiseX-92. Esses materiais e ruídos selecionadas estão listados na Tabela 1. Para cada combinação de material e ruído, é gerado um experimento diferente, totalizando 25 experimentos.

Cada experimento é testado com toda a amostra de teste de VoxCeleb [26]. O ruído sendo testado é posicionado sempre a 2 metros de distância do locutor. Amplitude do ruído é normalizada para a mesma da voz.

Tabela 1: Casos selecionados para experimentos

Materiais	Ruídos
hard_surface	babble
reverb_chamber	white
brickwork	car
wooden_lining	pink
panel_fabric_covered_6pcf	factory1

4.3 Especificações dos Experimentos

A Kaldi [25] foi feita para processamento em *cluster*, e suporta facilmente o processamento em diversos computadores. Dessa maneira, na execução de processos, o *toolkit* recebe por parâmetro as seguintes variáveis: (i) número de *jobs*, que indica quantas execuções serão feitas (os *jobs* podem ser alocados em *cluster*); (ii) número de processos a cada *job*; (iii) número de *threads* para cada processo; (iv) quantidade de memória por processo. Alguns algoritmos podem se beneficiar de mais processos, *threads* ou *jobs* dependendo de sua implementação. Algumas vezes o próprio *script* possui dicas dos melhores parâmetros de execução.

Foi utilizada a *recipe* VoxCeleb/v1, introduzida na Seção 2.2. Alguns ajustes foram necessários para execução dos testes.

A - Ajustar local das bases de dados: A leitura das bases de dados pelos *scripts* funciona com base em organizações de pastas específicas. É indispensável que a base esteja organizada corretamente. No caso da *recipe* VoxCeleb/v1, um arquivo é até mesmo baixado em sua primeira execução.

B - Download de modelo treinado: Foi utilizado um modelo já treinado da *recipe* VoxCeleb. Esse modelo foi disponibilizado na página do Kaldi[4]. O modelo consiste nos parâmetros treinados do MFCC, VAD, extrator de *i*-vectors, e parâmetros do PLDA.

C - Remoção das etapas de treinamento: Para executar somente os testes, é necessário editar o código da *recipe*, mantendo somente os trechos relativos à fase de testes, e não de treinamento.

D - Cuidados com o gerenciamento de memória: Os *scripts* estão com parâmetros de execução altos. Em alguns casos, mesmo uma memória de 32GB não foi suficiente para executar com os parâmetros padrões. Foi necessário limitar o uso de memória, *jobs* e processos para adequar-se aos recursos disponiveis. Um exemplo disso é o extrator de *i*-vector: por padrão, utiliza 4GB e 80 *jobs*, e foi reduzido para 2GB com 5 *jobs*.

Todos os experimentos foram testados com o modelo já treinado. Foram utilizadas as 4874 amostras de testes de VoxCeleb nos experimentos. Cada execução do experimento demanda ∼ 40 minutos, sendo ∼ 25 minutos para simulações com `pyroomacoustics` e ∼ 15 minutos para extração de $i-vectors$ e classificação. Os experimentos com casos selecionados duraram aproximadamente 17 horas para finalizar a execução.

4.4 Métricas de Avaliação

O reconhecimento de locutor é normalmente avaliado com acurácia, EER ou *Speaker Identification Performance* (SID). Enquanto acurácia e SID são mais adequados para avaliação da identificação de locutor,

[4] https://kaldi-asr.org/models/m7

EER é mais adequado para a verificação de locutor. Neste trabalho, o classificador é voltado para a verificação de locutor.

As métricas para verificação de locutor são: *False Rejection Rate* (FRR) – Eq. (1) –, *False Acceptance Rate* (FAR) – Eq. (2) – e EER, com as seguintes definições [22]:

$$\begin{aligned} \text{FRR} &= \frac{\text{VN}}{\text{VP}+\text{VN}} = \text{Probabilidade de Erro} \\ &= \frac{\text{Total de locutores corretos rejeitados}}{\text{Total de testes com locutores corretos}}, \end{aligned} \tag{1}$$

$$\begin{aligned} \text{FAR} &= \frac{\text{FP}}{\text{FP}+\text{FN}} = \text{Probabilidade de Alarme Falso} \\ &= \frac{\text{Total de locutores impostores aceitos}}{\text{Total de testes com locutores impostores}}. \end{aligned} \tag{2}$$

Já EER se refere ao limiar de decisão no qual FAR = FRR, de maneira a manter um equilíbrio do sistema. As métricas FAR e FRR são inversamente proporcionais, então: se o FAR estiver alto e o FRR baixo, o sistema será "amigável" para o usuário, mas inseguro; se o FRR estiver alto e o FAR baixo, o sistema não será "amigável" para o usuário, mas seguro. Cabe ao projetista tomar a decisão de qual o equilíbrio o sistema deve ter.

Outra métrica utilizada para avaliação do NIST [17] é a *Decision Cost Function* (DCF) [22]. DCF é uma soma ponderada de FRR e FAR:

$$\begin{aligned} C_{Det}(\theta) = {} & C_{Miss} \times P_{\text{Miss}|\text{Target}}(\theta) \times P_{\text{Target}} + \\ & C_{\text{FalseAlarm}} \times P_{\text{FalseAlarm}|\text{Nontarget}}(\theta) \times (1 - P_{\text{Target}}) \end{aligned} \tag{3}$$

em que θ o limiar de decisão, C_{Miss} o custo da rejeição falsa, $C_{\text{FalseAlarm}}$ o custo da aceitação falsa, P_{Target} é a probabilidade de locutores alvos. Existem algumas variações da Eq. (3), de acordo com o ano do teste NIST de referência. O MinDCF, por exemplo, presente nos resultados deste trabalho, se refere ao valor mínimo de DCF.

5 RESULTADOS

Na Tabela 2 são mostrados os resultados do experimento global realizado. É importante esclarecer que o experimento com ruído aditivo apresentou um SNR menor que em reverberação e ruídos. Observa-se que após a reverberação, o EER aumentou $\sim 2,6\%$, e com ruído aditivo, aumentou $\sim 7,7\%$, indicando que ruídos são mais prejudiciais à classificação do que somente a reverberação. Após a combinação de ambos reverberação e ruído, o EER ficou maior que nas situações anteriores.

A Tabela 3 evidencia melhor a relação entre reverberação e ruído. Os materiais estão ordenados por maiores coeficientes de absorção, onde o primeiro material é feito pra ter uma absorção alta e o último é uma câmara de reverberação. Os ruídos estão ordenados pelos resultados de EER médio. Observa-se que o experimento com o menor EER teve $\sim 0.2\%$ de diferença para o experimento sem nenhuma corrupção. Para todos os ruídos, pode-se perceber que o EER aumenta conforme o coeficiente de absorção diminui. O ruído branco, por exemplo, apresentou o maior EER, chegando a $\sim 29,77\%$ no pior caso.

A Figura 3 mostra o EER de cada experimento em uma coluna, com a cor indicando o ruído e o agrupamento indicando o material testado. Fica evidenciado o impacto que a reverberação de cada ambiente tem no EER com diferentes ruídos. Os mesmos ruídos intensificaram a degradação do EER nos materiais que apresentam maior reverberação.

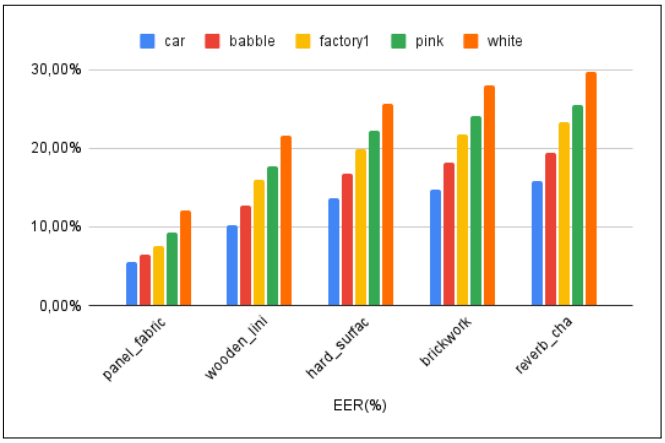

Figura 3: Resultados do experimento de casos selecionados por material em EER.

Na Figura 4 cada coluna é o EER de um experimento, indicando o material e o ruído testado. Esse gráfico evidencia as diferenças de comportamento entre os materiais. A divergência no comportamento se deve à situação que os coeficientes de absorção de cada ambiente variam de acordo com a frequência, absorvendo cada ruído de maneira diferente.

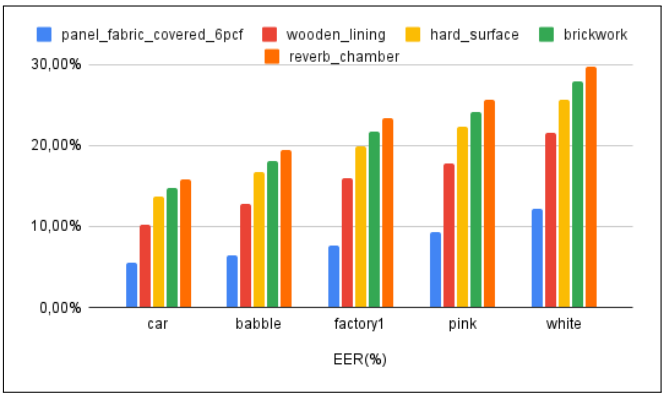

Figura 4: Resultados do experimento de casos selecionados por ruído em EER.

Entre os ruídos, o que causou maior degradação do EER foi o ruído branco. A característica do ruído branco é um espectrograma plano, ou seja, possui uma intensidade igual em todas as frequências. Ao combinar o ruído branco no ambiente menos reverberante, o EER foi de 12,12%, enquanto no ambiente mais reverberante, o EER foi 29,77%, uma perda de 17,65%. Comparando o EER sem reverberação para com reverberação, observa-se um aumento no EER de 24,43%.

6 CONCLUSÃO

Esse trabalho avaliou uma abordagem para a simulação de ruídos com reverberação, com o objetivo de fazer com que experimentos

Tabela 2: Resultados do experimento global

Experimento	EER	minDCF (10^{-2})	minDCF (10^{-3})
Sem corrupção	5,34%	0,50	0,61
Com reverberação	7,95%	0,59	0,70
Com ruído aditivo (SNR 0,3)	13,04%	0,87	*0,98*
Com reverberação e ruídos	16,76%	0,92	0,95

Tabela 3: Resultados do experimento de casos selecionados

EER (%)	car	babble	factory1	pink	white	EER Médio
panel_fabric_covered_6pcf	5,50%	6,42%	7,56%	9,25%	12,12%	8,17%
wooden_lining	10,20%	12,80%	15,98%	17,79%	21,55%	15,66%
hard_surface	13,66%	16,72%	19,96%	22,24%	25,66%	19,64%
brickwork	14,77%	18,11%	21,71%	24,05%	27,96%	21,32%
reverb_chamber	15,84%	19,50%	23,30%	25,60%	29,77%	22,80%
EER Médio	11,99%	14,71%	17,70%	19,79%	23,41%	17,52%

simulados apresentem maior semelhança com ambientes reais. Foi utilizado, nos experimentos, um modelo previamente treinado de reconhecimento de locutor, no qual foi analisado o impacto de ruídos em conjunto com reverberação em sua taxa de erro.

Somente a reverberação já piorou o EER dos modelos de classificação. A presença de ruídos aditivos aumentou o EER mais do que somente reverberação. Contudo, o conjunto de reverberação com ruídos degradou ainda mais a qualidade da classificação.

No pior caso testado, um ambiente reverberante intensificou, em média, 24,43% a degradação causada por ruído. Esse caso utilizou ruído branco, que está presente em todas as frequências, além de utilizar o material de uma câmara de reverberação. Por fim, pode-se concluir que quanto mais um ruído estiver sujeito à reverberação, maior a degradação do classificador.

Foi possível concluir que o modelo de reconhecimento de locutor avaliado não é robusto para aplicações em ambientes que apresentam ruídos. Essa falta de robustez dificulta a adesão de sistemas de reconhecimento de voz. Pode ser possível aumentar a robustez do reconhecimento de locutor por voz por meio da aplicação de métodos de filtragem ou do treinamento de novos modelos de classificação.

REFERÊNCIAS

[1] K Rao and Sourjya Sarkar. *Robust Speaker Recognition in Noisy Environments*. 2014. ISBN 978-3-319-07129-9. doi: 10.1007/978-3-319-07130-5.

[2] Douglas Vieira. 100 comandos da Google Assistente que você precisa conhecer, 2021. URL https://www.tecmundo.com.br/software/218823-100-comandos-google-assistente-voce-precisa-conhecer.htm.

[3] Leonardo Muller. Google Assistente agora permite transferir dinheiro usando comandos de voz - TecMundo, 2018. URL https://www.tecmundo.com.br/software/128478-google-assistente-permite-transferir-dinheiro-usando-comandos-voz.htm.

[4] D H H Nguyen, X Xiao, E S Chng, and H Li. Feature Adaptation Using Linear Spectro-Temporal Transform for Robust Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(6):1006–1019, 2016. ISSN 2329-9304. doi: 10.1109/TASLP.2016.2522646.

[5] Huan Feng, Kassem Fawaz, and Kang G Shin. Continuous Authentication for Voice Assistants. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, MobiCom '17, pages 343–355, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349161. doi: 10.1145/3117811.3117823. URL https://doi.org/10.1145/3117811.3117823.

[6] Xinyu Lei, Guan-Hua Tu, Alex X Liu, Chi-Yu Li, and Tian Xie. The Insecurity of Home Digital Voice Assistants - Amazon Alexa as a Case Study. *CoRR*, abs/1712.0, 2017. URL http://arxiv.org/abs/1712.03327.

[7] Google. Vincular sua voz aos dispositivos com o Voice Match. URL https://support.google.com/assistant/answer/9071681#zippy=%2Cvoice-match-e-resultados-personalizados.

[8] Lawrence R. Rabiner and Ronald W Schafer. *Theory and applications of digital speech processing*. Pearson/Prentice Hall, 2011. ISBN 0136034284; 9780136034285.

[9] J.-S. Park, G.-J. Jang, J.-H. Kim, and S.-S. Yeo. Unsupervised noise reduction scheme for voice-based information retrieval in mobile environments. *Multimedia Tools and Applications*, 75(9):4981–4996, 2016. ISSN 13807501 (ISSN). doi: 10.1007/s11042-013-1788-y. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-84888803076&doi=10.1007%2Fs11042-013-1788-y&partnerID=40&md5=8e6c926bbef88525ef89c1f2b7d76a89.

[10] Philipos C Loizou. *Speech enhancement: theory and practice*. Second edition, first issued in paperback edition, 2017. ISBN 9781138075573; 1138075574; 9781466504219; 1466504218.

[11] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333, April 2018. doi: 10.1109/ICASSP.2018.8461375.

[12] Sefik Emre Eskimez, Peter Soufleris, Zhiyao Duan, and Wendi Heinzelman. Front-end speech enhancement for commercial speaker verification systems. *Speech Communication*, 99:101–113, 2018. ISSN 0167-6393. doi: https://doi.org/10.1016/j.specom.2018.03.008. URL https://www.sciencedirect.com/science/article/pii/S0167639317302480.

[13] Waad Ben Kheder, Driss Matrouf, Moez Ajili, and Jean-Francois Bonastre. A Unified Joint Model to Deal With Nuisance Variabilities in the I-Vector Space. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 26(3):633–645, 2018. ISSN 2329-9290. doi: 10.1109/TASLP.2018.2789399. URL https://doi.org/10.1109/TASLP.2018.2789399.

[14] Hassan Taherian, Zhong-Qiu Wang, Jorge Chang, and DeLiang Wang. Robust Speaker Recognition Based on Single-Channel and Multi-Channel Speech Enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1293–1302, 2020. ISSN 2329-9290. doi: 10.1109/TASLP.2020.2986896. URL https://doi.org/10.1109/TASLP.2020.2986896https://ieeexplore.ieee.org/document/9064910/.

[15] Ali Bou Nassif, Ismail Shahin, Shibani Hamsa, Nawel Nemmour, and Keikichi Hirose. CASA-based speaker identification using cascaded GMM-CNN classifier in noisy and emotional talking conditions. *Applied Soft Computing*, 103:107141, 2021. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2021.107141. URL https://www.sciencedirect.com/science/article/pii/S1568494621000648.

[16] Mitchell McLaren, Luciana Ferrer, Diego Castán, and Aaron D Lawson. The Speakers in the Wild (SITW) Speaker Recognition Database. In *INTERSPEECH*, 2016.

[17] NIST. The nist year 2010 speaker recognition evaluation plan, 2010. URL https://www.nist.gov/sites/default/files/documents/itl/iad/mig/NIST_SRE10_

evalplan-r6.pdf.

[18] Jesús Villalba, Nanxin Chen, David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Jonas Borgstrom, Leibny Paola García-Perera, Fred Richardson, Réda Dehak, Pedro A Torres-Carrasquillo, and Najim Dehak. State-of-the-art speaker recognition with neural network embeddings in NIST SRE18 and Speakers in the Wild evaluations. *Computer Speech & Language*, 60:101026, 2020. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2019.101026. URL http://www.sciencedirect.com/science/article/pii/S0885230819302700.

[19] Robin Scheibler, Eric Bezzam, and Ivan Dokmanić. Pyroomacoustics: A Python Package for Audio Room Simulation and Array Processing Algorithms. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 351–355, 2018. doi: 10.1109/ICASSP.2018.8461310.

[20] Matti Karjalainen Ville Pulkki. *Communication Acoustics: An Introduction to Speech, Audio and Psychoacoustics*. Wiley, 1 edition, 2015. ISBN 9781118866542; 1118866541. URL libgen.li/file.php?md5=1efcb76d31c03ce17bf11c7f5c51113e.

[21] Ondřej Novotný, Oldřich Plchot, Ondřej Glembek, Jan "Honza" Černocký, and Lukáš Burget. Analysis of DNN Speech Signal Enhancement for Robust Speaker Recognition. *Computer Speech Language*, 58:403–421, 2018. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2019.06.004. URL http://www.sciencedirect.com/science/article/pii/S0885230818303607.

[22] Man-Wai Mak and Jen-Tzung Chien. *Machine Learning for Speaker Recognition*. Cambridge University Press, 2020. doi: 10.1017/9781108552332.

[23] A. VARGA. The noisex-92 study on the effect of additive noise on automatic speech recognition. *ical Report, DRA Speech Research Unit*, 1992. URL https://ci.nii.ac.jp/naid/10006472430/en/.

[24] Najim Dehak, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, May 2011. ISSN 1558-7924. doi: 10.1109/TASL.2010.2064307.

[25] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. IEEE Catalog No.: CFP11SRW-USB.

[26] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

[27] David Snyder, Guoguo Chen, and Daniel Povey. MUSAN: A music, speech, and noise corpus. *CoRR*, abs/1510.08484, 2015. URL http://arxiv.org/abs/1510.08484.

[28] M C A Korba, H Bourouba, and D Rafik. Text-Independent Speaker Identification by Combining MFCC and MVA Features. In *2018 International Conference on Signal, Image, Vision and their Applications (SIVA)*, pages 1–5, 2018. ISBN VO -. doi: 10.1109/SIVA.2018.8661138.

[29] Z Wang, S Duan, C Zeng, X Yu, Y Yang, and H Wu. Robust Speaker Identification of IoT based on Stacked Sparse Denoising Auto-encoders. In *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, pages 252–257, 2020. ISBN VO -. doi: 10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00056.