Estudo da variabilidade em testes na verificação de locutores com redes neurais profundas

Victor Costa Beraldo
Pós-Graduação em Engenharia da Informação (PPGInf)
Universidade Federal do ABC
Santo André, Brasil
victor.beraldo@ufabc.edu.br

Murilo Bellezoni Loiola
Pós-Graduação em Engenharia da Informação (PPGInf)
Universidade Federal do ABC
Santo André, Brasil
murilo.loiola@ufabc.edu.br

***Resumo*—A verificação automática de locutores revela-se de grande importância para segurança na autenticação de pessoas. Concebida através de dados de voz, seu papel como ferramenta de autenticação tem sido feito, com melhores desempenhos, por meio do aprendizado profundo de redes neurais utilizando d-vectors. Entre seus benefícios está a desnecessidade de treinar novos modelos para verificar locutores inexistentes nas bases de treinamento. Neste contexto, notou-se a necessidade de comparar diferentes modelos baseados nos d-vectors, em situações onde temos dados para treinamento que não foram obtidos pela mesma origem que as possíveis bases de teste, representando um problema real devido a diferentes fontes de variabilidade nos dados de voz, como diferentes idiomas, áudios gravados com dispositivos diferentes e ruídos de fundo diferentes, onde necessita-se escolher um modelo, porém não há dados de treinamento e teste com as mesmas características. As comparações foram realizadas entre os modelos SincNet, GE2E, redes ResNet Triplet Loss e o modelo proposto neste trabalho SincNet + GE2E, cujo desempenho supera o a rede GE2E original, porém até então não superou o desempenho da SincNet original.**

***Palavras-chave*—Redes Neurais, D-Vectors, Verificação de Locutores**

I. Introdução

A verificação de identidade de uma pessoa é um requisito essencial para controle de acessos para proteger recursos. A identidade pessoal é geralmente requisitada pela apresentação de uma propriedade pessoal única, como uma chave ou senha [1].

Com o rápido crescimento da internet móvel e smartphones, problemas com a segurança de dados criaram a necessidade de uma autenticação mais robusta. A metodologia de número de identificação do usuário e senha existente, embora tolerável para algumas aplicações como uso de desktops e laptops, é inadequada para o uso de dispositivos móveis, pela possibilidade de estar na mão de pessoas não autorizadas, por exemplo. Conforme a percepção de que a quantidade de informações sensíveis que nossos dispositivos podem conter aumenta, será essencial que a autenticação biométrica seja parte integrante do acesso a informações e arquivos existentes no dispositivo. Sendo a fala o meio mais natural da comunicação humana, é possível supor que uma autenticação automática nesses dispositivos sejam baseada na voz com o passar do tempo, assim como outros tipos de biometria. Modelos recentes de smartphones já utilizam esta tecnologia [2].

Além de aplicações de autenticação pessoal para controle de acessos, o reconhecimento de locutor é uma ferramenta importante na aplicação da lei, segurança e análise forense em geral, devido ao fato dos telefones celulares estarem se tornando o principal meio de comunicação para o público em geral e também para criminosos. Essas pessoas mal intencionadas, geralmente, não querem ser reconhecidas e muitas vezes tentam alterar sua voz para que não sejam identificadas. Este problema introduz novos desafios para desenvolvedores de tecnologias de reconhecimento de locutor [2].

A modelagem de verificadores de locutores, antes dominada pela utilização dos i-vectors [3], que são baseados em modelos de mistura de gaussianas, atualmente tem sido realizada cada vez mais com a aplicação de redes neurais profundas [4]–[6], utilizando uma estratégia análoga, apelidada de d-vectors [7]. Esta abordagem se beneficia da alta capacidade de generalização das redes neurais, com um método escalável de verificação, sem necessidade de treinar um modelo por falante e que também é capaz de realizar a verificação de maneira satisfatória apenas com o uso da distância cosseno entre vetores de cadastro e de teste.

Levando em consideração os recentes estudos deste tema, compreende-se a fundamental importância da utilização das novas técnicas de aprendizado profundo para verificação de locutores. Notou-se também a insuficiência de estudos que comparassem essas técnicas em bases de dados preparadas nas mesmas condições, utilizando d-vectors como ferramenta principal para verificação das vozes.

Dado este cenário, o objetivo deste trabalho é realizar uma análise comparativa de diferentes arquiteturas de redes neurais artificiais profundas, e que utilizam os d-vectors, nas mesmas condições de dados para treinamento e teste. Especificamente, são considerados cenários nos quais os dados a serem testados não foram coletados da mesma forma que os dados utilizados para treinamento, simulando problemas reais nos quais dispomos de áudios para treinamento gravados em uma condição e necessita-se testar o modelo treinado em outra base de áudios.

Este trabalho está divido da seguinte forma: a Seção II, apresenta os conceitos básicos sobre a verificação de locutores, suas principais etapas e complicações vindas das fontes de variabilidade na verificação de locutores. A Seção III apresenta os detalhes sobre as diferentes formas de modelagem utilizadas

e a criação das bases de dados. Os resultados das simulações são apresentados e discutidos na Seção IV. Por fim, a Seção V conclui o artigo.

II. Verificação do locutor

A tarefa de verificação é uma forma de reconhecimento de locutor que considera poucas comparações, geralmente de 1:1, onde é comparada apenas a voz a ser testada com a voz cadastrada por aquele locutor na base de dados. Logo, se o tamanho da população aumenta, a capacidade computacional se mantém constante. Esta característica é responsável pela sua popularidade, sendo considerada uma técnica simples se comparada à identificação do locutor, onde as comparações são feitas na forma 1:N, onde N é o número de locutores cadastrados na base de dados, com objetivo de identificar quem é o locutor e não apenas verificá-lo.

A. Etapas de um sistema de verificação de locutores

O diagrama representado pela Figura 1 exemplifica como um sistema verificação de locutor funciona. Este sistema pode variar entre as metodologias, porém algumas etapas são comuns a maioria delas, como as de cadastramento e de teste dos locutores.

A etapa de modelagem dos locutores é conhecida por realizar o treinamento do modelo universal (UBM, do inglês *Universal Background Model*) [8]. Este modelo contempla dados de vozes de diversos locutores diferentes. A ideia por trás dele é poder comparar o locutor teste com características presentes em diversas vozes. Os atributos presentes nos dados de voz são extraídos geralmente através de segmentos da fala, denominados *frames*, por meio de técnicas como MFCCs (do inglês *Mel-Frequency Cepstral Coefficients*) [2] ou mesmo diretamente dos áudio por meio de redes neurais profundas [5].

A realização do cadastramento dos locutores é feita a partir da extração de atributos da voz do locutor de modo a criar um modelo dependente do locutor ou uma marca vocal que será cadastrada, comumente gerada como uma adaptação de um modelo universal. O teste, por sua vez, também é feito utilizando a extração de atributos da voz de um locutor teste, de modo a poder comparar as informação presentes nessa voz a ser testada com o modelo cadastrado do mesmo locutor. Essa comparação é feita no processo de pontuação, onde é gerado um valor escalar, medindo a diferença entre as vozes e, caso supere um certo limite τ, confirma a verificação.

B. D-vectors

Redes neurais profundas, do inglês *Deep Neural Networks* (DNNs), estão conseguindo com sucesso desempenhar tarefas de processamento de voz, grande parte em aplicações de reconhecimento de fala [2]. Aplicações em reconhecimento de locutores também estão sendo testadas, superando técnicas tradicionais como os i-vectors [7], [9], [10].

Uma das metodologias mais conhecidas atualmente em aplicações de verificação de locutores é denominada de d-vectors [7]. Esta abordagem é feita a partir da utilização

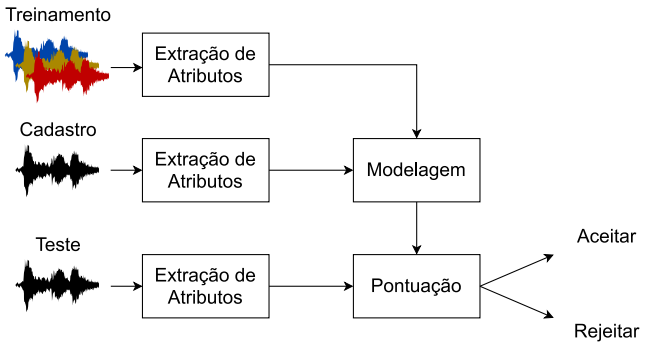

Figura 1. Arquitetura de um sistema de verificação de locutor simplificado.

de um DNN para obtenção de um modelo universal, capaz de modelar diretamente as características de voz de diversos falantes, como um extrator de vetores de características. Esta DNN é treinada utilizando atributos de frames da voz com objetivo de discriminar os locutores, conforme representado na Figura 2. Após o treinamento, a etapa de cadastramento é feita a partir da obtenção da média das ativações da última camada escondida da DNN, chamada de vetor profundo ou d-vector. A etapa de pontuação, por sua vez, é feita a partir da comparação do d-vector do locutor alvo com o d-vector extraído da voz de um locutor teste.

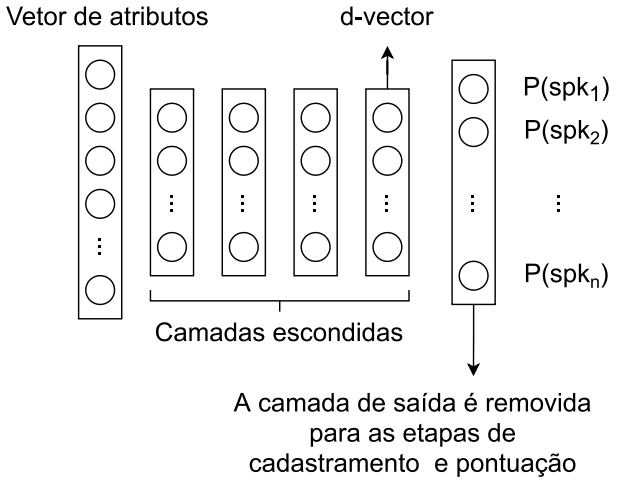

Figura 2. Arquitetura de uma DNN usada para verificação de locutores.
Fonte: Adaptado de [7]

A saída da última camada escondida é escolhida no lugar da camada softmax por diversos motivos. Ao reduzir o tamanho da rede neural retirando a camada de saída, além de simplificar o modelo, o torna capaz de ser utilizado como um extrator de características da voz que represente o locutor e não classificar classificá-lo diretamente. Essa estratégia de modelagem é útil mesmo para novos falantes, sem necessidade de retreino. A utilização da penúltima camada tende a obter melhor

generalização do que se fosse utilizada a última camada, em casos onde os falantes testados não estavam presentes na etapa de treinamento.

C. Fontes de variabilidade na verificação de locutores

Diferente de outras formas de biometria, como a impressão digital, palma da mão, iris e características da face, a voz humana apresenta várias formas de variabilidade. A mesma pessoa pode repetir as mesmas palavras de diversas formas, representando a variabilidade intra-locutor, ou mesmo a variedade de dispositivos de gravação ou métodos de transmissão. Uma pessoa pode ter sua voz alterada se for gravada através de uma ligação telefônica, ou mesmo em situações de alterações na voz causada por algum problema de saúde. Em geral, 3 tipos de variabilidades podem dificultar o reconhecimento de locutores: variabilidade de locutor, conversacional e de tecnologia [2], [11]. Exemplos de cada uma são mostrados na Figura 3 usando setas nas cores verde, azul e vermelho, respectivamente.

A variabilidade de locutor reflete as mudanças em como o locutor fala e como isso afeta o desempenho de um sistema de reconhecimento de locutores. Ela pode ser vista como a variabilidade intrínseca do locutor, baseada em alguns fatores como a emoção, problemas de saúde, idade, aumento do esforço vocal na presença de ruído (efeito Lombard [12]), cansaço e o efeito de outras tarefas sendo executadas enquanto fala. A variabilidade conversacional, por outro lado, reflete cenários nos quais a diferença na voz é criada a partir da interação de pessoas ou de pessoas com sistemas. Ela é encontrada quando seres humanos alteram a sua forma de falar ao interagirem com uma pessoa ou um grupo de pessoas, mudança de sotaque, idioma ou dialeto e quando se conversa com máquinas como sistemas de reconhecimento de fala. Por último, tem-se a variabilidade causada pela tecnologia, representada por fatores externos como a forma com que o áudio foi gravado, presença de ruídos do ambiente, relação sinal-ruído (SNR, do inglês *signal-to-noise ratio*), distância do microfone e atributos da qualidade do áudio, incluindo taxa de amostragem, duração e compressão.

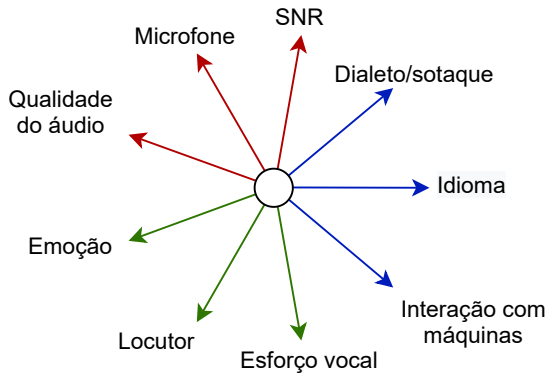

Figura 3. Fontes de variabilidade.
Fonte: Adaptado de [2]

As formas de variação da voz são consideradas um dos maiores desafios para sistemas de reconhecimento de falantes. Sua tarefa consiste em discriminar se a variabilidade da voz é do mesmo locutor (variabilidade intra-locutor) ou diferente (variabilidade inter-locutor). Tais variabilidades são representadas na Figura 4.

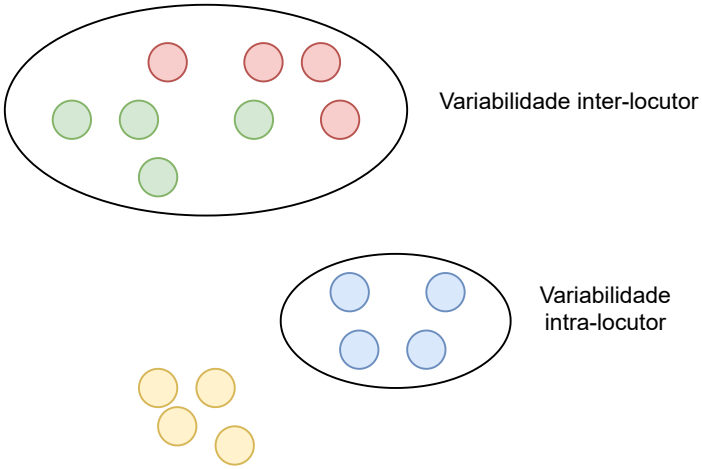

Figura 4. Variabilidade inter-locutor e intra-locutor.
Fonte: Adaptado de [2]

Este trabalho realizou experimentos com foco em explorar bases de dados contendo várias combinações das variabilidades explicadas nesta seção e em entender qual o efeito dessas variações na avaliação de um grupo específico de modelos de verificação de locutores texto-independentes.

III. Modelagem

A. BASES DE DADOS

Os dados utilizados neste trabalho foram organizados de modo a simular uma situação onde os dados de treinamento disponíveis não contém áudios com as mesmas características dos áudios que serão testados. Com este propósito, as bases de dados foram extraídas de três bases de dados públicas: LibriSpeech [13], MLS (do inglês *Multilingual LibriSpeech*) [14] e TIMIT (do inglês *Texas Instruments Massachussets Institute of Technology*) [15].

A LibriSpeech é uma base de dados contendo vozes em inglês presentes em áudio-livros provenientes do projeto LibriVox, contendo 1000 horas de falas amostradas em 16 kHz [16]. A Base B1 foi feita através desta base, a qual foram selecionados áudios, subamostrados a 8kHz de 1500 locutores aleatórios para treinamento e outros 500 para teste, sendo de 4 a 10 trechos por falante sem presença de trechos silenciosos.

Os dados da base TIMIT são gravados em 16 kHz com 16 bits por amostra, contendo vozes faladas especificamente por 630 locutores, de oito dialetos principais do inglês americano. A Base B2 foi gerada a partir da repartição padrão indicada em sua documentação e a subamostragem dos áudios em 8kHz. A repartição de treino contem 463 falantes diferentes e a de teste 168, ambas com 10 segmentos de áudio por locutor.

Tabela I
ESPECIFICAÇÕES DAS BASES DE DADOS UTILIZADAS

Nome	# Locutores	Segmentos de Áudio por Locutor	Origem	Função	Tempo por Segmento (s)
B1 (Treino)	1500	4-10	LibriSpeech Inglês	Treino	3.8 (2.8)
B1 (Teste)	500	4	LibriSpeech Inglês	Teste	4.2 (3.3)
B2	168	4	TIMIT	Teste	3.2 (0.9)
B3	55	4	MLS Librispeech Português	Teste	15.2 (2.9)
B4	40	4	VoxCeleb	Teste	8.8 (5.5)

A MLS, criada em 2020, contém mais áudios de mesma procedência da Librispeech, porém com mais de 50.000 horas de fala entre oito idiomas. A base de dados B3 foi criada com áudios em português da base MLS subamostrados em uma taxa de amostragem de 8kHz contendo 60 locutores diferentes. Utilizou-se apenas 4 segmentos de áudio por locutor. Esta base foi criada apenas para etapa de teste, para entender o comportamento dos modelos testados em outra língua, no caso o português.

A base de dados VoxCeleb [17] foi criada de forma automatizada a partir de videos do YouTube, utilizando Redes neurais convolucionais treinadas para verificarem tanto a voz quanto a face de mais de 1000 celebridades. Esta base de dados tem sido a mais utilizada para comparação de modelos de verificação de locutores, contendo vozes de pessoas de diferentes nacionalidades. A Base de dados criada utilizando esta base de origem é a B4. Obtida a partir da base VoxCeleb utilizando a divisão padrão de teste de sua documentação com áudios subamostrados em 8kHz, contendo 40 locutores diferentes, com 4 segmentos de áudio por locutor. Esta base foi criada a fim de testar uma base com uma variabilidade muito grande, uma vez que áudios de YouTube são gravados com os mais diversos dispositivos, as celebridades falam línguas diferentes, e os áudios são prejudicados por sons de risadas e músicas.

As novas bases criadas B1, B2, B3 e B4 têm suas especificações detalhadas nesta seção e resumidas através da Tabela I. Em resumo, as bases de dados de voz descritas nesta seção exercem papel de simular problemas reais ao treinar e avaliar modelos de verificação de locutores em diversos contextos, treinando sempre com a mesma base B1(Treino) e variando as bases de teste. A situação mais comum é representada pelo teste em B1(Teste), uma vez que as características das bases de treinamento e teste são semelhantes. Os testes realizados com as bases B2, B3 e B4 são mais complexos, pois são bases de teste com características diferentes, variando principalmente o canal (B2), a língua (B3) e diversas formas de variabilidade juntas (B4).

B. Modelos

Esta seção apresenta as técnicas utilizadas e as adaptações feitas em modelos de verificação de locutor, utilizando arquiteturas diferentes de redes neurais artificiais profundas como a Sincnet [5], a arquitetura baseada no trabalho GE2E (*Generalized end-to-end loss for speaker verification*) [4] e Redes Triplet Loss [6]. Todos os modelos utilizados foram adaptados para serem utilizados como um método de extração de características através das camadas mais profundas da rede neural, assim como os d-vectors. Seguindo a modalidade de verificação de locutores independente de texto (não necessita que as frases contidas nos áudios sejam as mesmas). Todos os modelos foram treinados utilizando a técnica de parada antecipada (do inglês *Early Stopping*), onde uma parte da base de treinamento, de 20%, foi separada como mesma base validação para todos os modelos.

1) GE2E: O modelo GE2E foi criado com o objetivo de melhorar o treinamento utilizando uma nova função de custo para a verificação de locutores, chamada de GE2E (*Generalized end-to-end*) [4]. Este treinamento é realizado utilizando um grande número de trechos de voz de uma vez, contendo N falantes diferentes e M trechos por falante. Cada vetor de características $\boldsymbol{x}_{ji}$ ($1 \leq j \leq N$ e $1 \leq i \leq M$) representa as características de um falante j e um trecho de voz i. As características extraídas são introduzidas em uma rede neural LSTM (do inglês *Long Short-Term Memory*) [18] com uma camada linear conectada ao final desta rede. Sendo a saída da rede dada por $f(\boldsymbol{x}_{ji}; w)$, onde $\boldsymbol{w}$ representa os parâmetros da rede toda, o d-vector é definido como a normalização L_2 da saída da rede, conforme (1):

$$\boldsymbol{e}_{ji} = \frac{f(\boldsymbol{x}_{ji}; \boldsymbol{w})}{||f(\boldsymbol{x}_{ji}; \boldsymbol{w})||_2}, \tag{1}$$

na qual $\boldsymbol{e}_{ji}$ representa o d-vector do j-ésimo locutor em seu i-ésimo trecho de voz.

O centroide $\boldsymbol{c}_k$ dos d-vectors de um locutor k é representado pela média dos d-vectors de seus M trechos de voz. A matriz de similaridade $\boldsymbol{S}_{ji}$ é definida como as similaridades cosseno (s_{cos}) entre o d-vector e todos os centroides $\boldsymbol{c}_k$ ($1 \leq j, k \leq N$ e $1 \leq i \leq M$) e está representada pela Equação (2):

$$\boldsymbol{S}_{ji,k} = \alpha \cdot s_{cos}(\boldsymbol{e}_{ji}, \boldsymbol{c}_k) + b, \tag{2}$$

onde α e b são parâmetros que podem ser aprendidos. Utiliza-se a restrição nos peso $\alpha > 0$, para que a similaridade entre os d-vectors aumente, enquanto a similaridade cosseno cresce.

A utilização desta função de custo na etapa de treinamento tem como objetivo obter d-vectors similares aos respectivos centroides dos d-vectors do mesmo locutor, mas ao mesmo tempo distantes de d-vectors de falantes diferentes. A implementação desta estrategia se dá pela utilização de uma camada Softmax em $S_{ji,k}$ para $k = 1, ..., N$. A definição da função de custo para cada d-vector é representada pela Equação (3):

$$L(\boldsymbol{e}_{ji}) = -\boldsymbol{S}_{ji,j} + \log \sum_{k=1}^{N} \exp(\boldsymbol{S}_{ji,k}). \quad (3)$$

A função de custo total, L_G , é a soma do custo (3) para todos os d-vectors da matriz de similaridade por meio da Equação (4):

$$L_G(\boldsymbol{S}) = \sum_{j,i} L(\boldsymbol{e}_{ji}). \quad (4)$$

O treinamento do modelo GE2E foi feito utilizando os parâmetros $M = 5$ e $N = 4$ e uma taxa de aprendizado de 0,01. Os demais parâmetros foram os mesmos que foram utilizados no artigo original na tarefa de verificação de locutores independente de texto [4]. Os atributos utilizados também foram os mesmos, a saber banco de filtros na escala mel com 40 dimensões para cada frame.

2) SincNet: A SincNet é uma arquitetura de redes neurais convolucionais (CNN, do inglês *Convolutional Neural Networks*), que utiliza como entrada o sinal de voz bruto, deixando que a rede aprenda atributos importantes para a discriminação das vozes dos locutores [5]. Grande parte de trabalhos passados utilizavam atributos extraídos manualmente, como MFCCs e bancos de filtros [4], [6], [7], [19]. Estas características extraídas, no entanto, não têm garantias de serem ótimas para todas as tarefas de modelagem da voz.

A estratégia que a SincNet executa é a utilização de camadas convolucionais na entrada da rede, com convoluções do sinal puro no domínio do tempo através de funções sinc parametrizadas como filtros passa-banda. As frequências de corte altas e baixas são os únicos parâmetros que são aprendidos nesta etapa. Esta arquitetura tem se mostrado mais eficiente em relação as CNNs convencionais, treinando mais rápido e atingindo melhor desempenho em testes. A arquitetura total da SincNet é apresentada na Figura 5 e será melhor explicada a seguir.

A arquitetura SincNet destaca-se pela primeira camada convolucional, que executa uma série de convoluções entre o sinal de voz e filtros retangulares passa-banda, cujas frequências de corte são parâmetros aprendidos pela rede neural. Após a aplicação das convoluções utilizando os filtros sinc, outras operações comuns em CNNs são empilhadas, como Pooling, Layer Norm, Dropout e, por fim, camadas convolucionais e densas são empilhadas para realizar a classificação dos locutores utilizando a função Softmax.

O treinamento do modelo SincNet foi realizado utilizando os mesmos parâmetros que foram utilizados no artigo original, alterando apenas o número de neurônios da última camada de acordo com o número de locutores, dependendo da base de treino utilizada.

3) SincNet + GE2E: O modelo Sincnet + GE2E é proposto neste trabalho com o objetivo de aproveitar as vantagens da utilização da arquitetura SincNet com função de custo GE2E. A Figura 6 expõe a estrutura do modelo. Ela utiliza o sinal de voz puro como entrada para o modelo SincNet, que extrai representações da voz dos N locutores com M segmentos

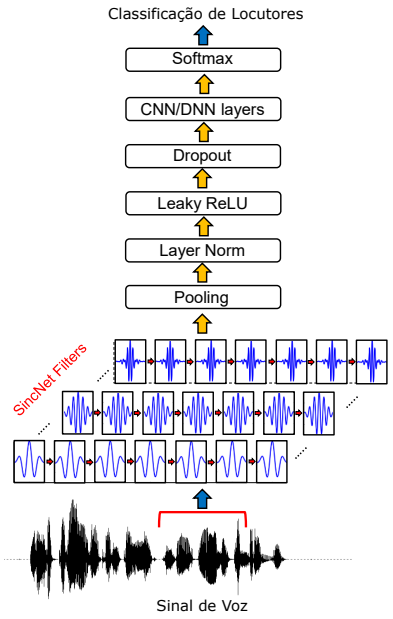

Figura 5. Arquitetura da SincNet.
Fonte: [5]

cada um. Através dos d-vectors, é construída uma matriz de similaridade, a partir da equação (2), entre cada segmento e o respectivo centroide (média dos d-vectors do mesmo locutor) no cálculo da função de custo GE2E.

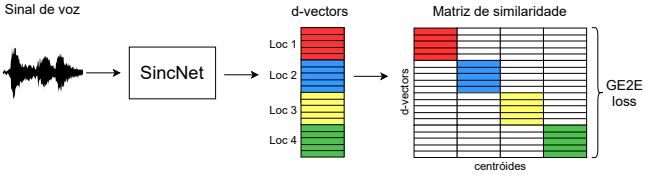

Figura 6. Arquitetura da SincNet + GE2E.

A construção deste modelo utilizou toda a arquitetura SincNet padrão, porém retirando a última camada Softmax e utilizando os d-vectors extraídos da penúltima camada para serem utilizados para o cálculo da função de custo GE2E. Desta forma, a rede é forçada a aprender representações das vozes com d-vectors distantes entre si para áudios de pessoas diferentes e próximos para áudios da mesma pessoa.

Para treinamento deste modelo, foram utilizados os parâmetros $M = 5$ e $N = 4$ e a alteração do número de neurônios das 3 últimas camadas densas da rede de 1024, 1024 e 1024 para 1024, 512 e 256. Esta alteração foi realizada, pois ao testar diferentes parâmetros, notou-se que essa diminuição de neurônios não alterava o desempenho final e tornava o treinamento mais rápido. Todos os demais parâmetros foram mantidos iguais aos que foram utilizados no treinamento da SincNet padrão.

4) Redes Triplet Loss: As redes Triplet Loss foram inspiradas nas redes siamesas, que são arquiteturas de redes

neurais introduzidas para a tarefa de reconhecimento de assinaturas [20] e que têm gerado grandes avanços na tarefa de reconhecimento de imagem [21], [22], causando interesse também na modelagem com dados de voz [6], [17]. Trabalhos recentes com redes siamesas têm utilizado a técnica *One Shot Learning*. Esta abordagem visa obter maior generalização dos dados, mesmo quando não há muitos dados rotulados da mesma classe. Situação semelhante é observada nas bases de treinamento de verificação de locutores, onde há muitos dados de locutores diferentes, porém poucos dados do mesmo locutor.

As redes Triplet Loss [23], assim como as siamesas, são projetadas através de redes com uma mesma arquitetura e pesos compartilhados, porém com três entradas diferentes: âncora, negativa e positiva, sendo a âncora e a positiva da mesma classe e a negativa, de classe diferente. A Figura 7 ilustra essa abordagem, adaptada para a tarefa de verificação de locutores, onde classes diferentes são representadas por locutores diferentes e é utilizada a distância cosseno para calcular a função de custo Triplet Loss, assim como foi realizado em [6].

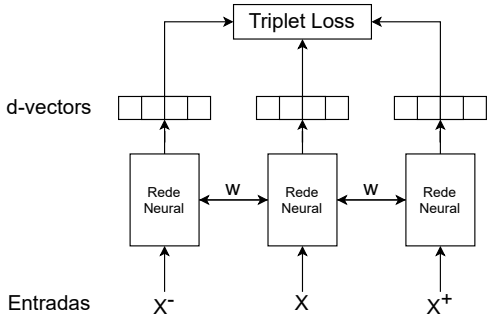

Figura 7. Arquitetura da Rede Triplet Loss adaptada para verificação de locutores.

A função de custo Triplet tem como objetivo maximizar a distância entre a âncora e a entrada negativa, enquanto minimiza a distância entre a âncora e a entrada positiva, sendo expressa por (5):

$$L_{triplet} = \max\left(d(a,p) - d(a,n) + m, 0\right), \qquad (5)$$

onde $d(a,p)$ e $d(a,n)$ representam as distâncias cosseno entre os d-vectors da âncora e das entradas positiva e negativa, respectivamente. A margem é representada por m, aumentando o custo para casos onde a âncora é próxima tanto da entrada positiva, quanto da negativa.

A modelagem através desta arquitetura foi baseada no artigo [6], com o treinamento de uma rede neural convolucional ResNet [24], utilizando um bancos de filtros de tamanho 64 como entrada para cada *frame* de áudio de 25 ms com 10 ms de sobreposição. Foram utilizados 4 blocos residuais contendo 32, 64, 128 e 256 canais, respectivamente. Utilizam-se também camadas convolucionais antes de cada bloco residual, garantindo que a dimensão permaneça constante em todas camadas convolucionais.

O treinamento foi realizado seguindo a estratégia da utilização da rede ResNet com função de custo entropia cruzada como um modelo pré-treinado. Após o treinamento, utiliza-se esta rede com os seus pesos aprendidos, e é treinada novamente, utilizando a função de custo Triplet Loss com o valor de margem de 0,2. Essa modelagem com retreino da rede ResNet com a função de custo Triplet Loss é chamada neste artigo de PResNet Triplet Loss.

IV. Resultados e Discussão

Os resultados que serão apresentados nesta seção foram obtidos utilizando a mesma base de treinamento B1(Treino) e variando as bases de teste, para todos os modelos, comparados utilizando a métrica EER [1]

A metodologia de avaliação dos modelos foi realizada de forma padronizada, a partir das etapas de cadastro e teste, representadas pelas figuras 8 e 9, respectivamente. Na etapa de cadastro, foram processados áudios de cada uma das bases de teste separadamente, de modo a gerar um d-vector por áudio. Em seguida, são separados os 4 áudios de cada um dos locutores, formando a base de d-vector de cadastro DCadastro a partir da média dos 3 d-vectors por locutor, enquanto a base de d-vector de teste DTeste é obtido diretamente de 1 d-vector por locutor, obrigatoriamente diferente dos 3 usado no cadastro. A etapa de pontuação é feita a partir da comparação entre cada um dos DTeste com todos os DCadastro, utilizando a distância cosseno, gerando uma tabela com vários testes de verificação e suas respectivas distâncias. A métrica EER é, enfim, calculada. Todo esse processo é repetido 16 vezes para cada base de testes diferente, com todas as combinações possíveis entre os 4 d-vectors de cadastro e 4 d-vectors de teste de cada locutor.

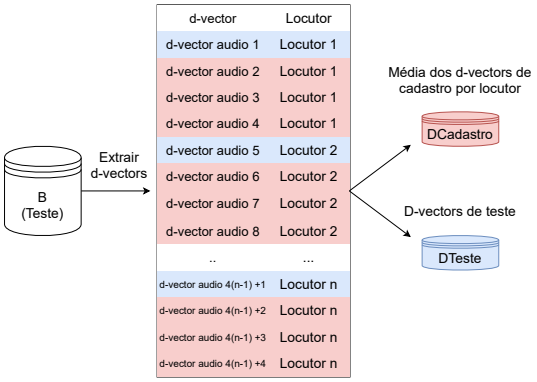

Figura 8. Descrição do procedimento de Cadastro.

Os resultados dos experimentos são expostos por um Diagrama de Caixas pela Figura 10 e resumidos através da Tabela II, com as médias e desvios-padrão entre parênteses, mostram que o modelo SincNet obteve melhores desempenhos para as bases de teste B1, B2 e B4. Já o modelo PResNet Triplet

[1]O *Equal Error Rate* (EER) é uma das métrica mais utilizadas para avaliar verificação de locutores. Ela é obtida pelo ponto onde os erros por falsa aceitação e falsa rejeição são iguais, conforme o limiar usado na etapa de pontuação do sistema varia.

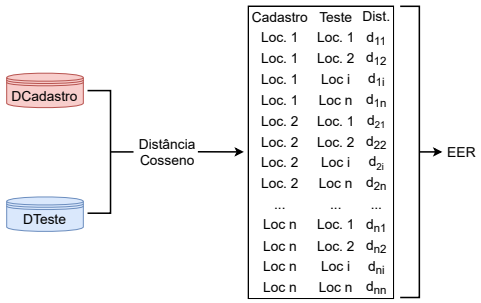

Figura 9. Descrição do procedimento de Pontuação

Loss foi melhor para a base de testes B3. O modelo proposto SincNet + GE2E obteve melhores resultados em todas as bases em relação ao modelo GE2E, porém não superou o desempenho do modelo SincNet para nenhuma base. O modelo PResNet Triplet Losss obteve um desempenho melhor que o modelo proposto SincNet + GE2E para todas as bases de teste, menos B1.

O ganho de desempenho do modelo proposto SincNet + GE2E em relação ao modelo GE2E sugere a vantagem na utilização da SincNet como extrator dos d-vectors, na qual os atributos extraídos do sinal utilizados na modelagem são aprendidos pela rede através da sua camada convolucional, diferente dos bancos de filtros utilizado no modelo GE2E.

Os resultados médios de EER em B1 foram os melhores para quase todos os modelos, com exceção do modelo PResNet Triplet Loss, como esperado, pois são dados de teste semelhantes aos que foram utilizado para treinamento dos modelos oriundos de B1 (treino e teste). A B4, por sua vez, obteve os piores resultados médios de EER para todos os modelos, conforme esperado, pois as suas fontes de variabilidade das vozes é a maior entre as bases de teste.

A análise dos desvios-padrão mostradas na Tabela II expõe a variação de desempenho ao escolher áudios que serão utilizados para cadastro e teste. O modelo SincNet apresentou o menor desvio padrão médio entre as bases, perdendo apenas na base B3 para o modelo PResNet Triplet Loss.

Na Tabela III são apresentadas as relações entre os resultados testados em B1 e as demais bases. Esses resultados expõe o problema real estudado neste artigo, pela perda de generalização dos modelos quando testados em áudios de origens diferentes. Neste contexto, foi observado que a base B3 teve a menor perda média de 169% em relação aos resultados testados em B1. Esse resultado sugere que a variabilidade causada por fatores tecnológicos influenciam mais que causas conversacionais entre os idiomas português e inglês, para a maioria dos modelos de verificação de locutores independente de texto estudados. o Modelo SincNet foi o único que obteve uma perda relativa maior em B3 do que B2.

V. Conclusões

Os experimentos desenvolvidos neste trabalho mostraram o efeito de diversas fontes de variabilidade nos dados de voz, com diferentes estratégias de modelagem para tarefa de

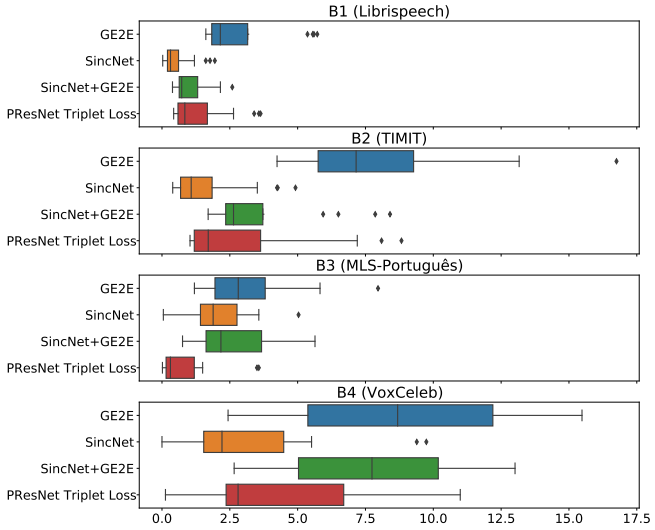

Figura 10. Resultados de EER em Diagrama de Caixa. Resultados estão divididos em 4 bases de teste B1, B2, B3 e B4. As bases que as originaram estão descritas entre parenteses

Tabela II
Resultados medidos em EER(%)

	B1	B2	B3	B4
GE2E	2.15 (1.61)	7.16 (3.61)	2.81 (1.89)	8.69 (4.07)
SincNet	**0.31 (0.64)**	**1.08 (1.56)**	1.89 (1.29)	**2.21 (3.0)**
SincNet+GE2E	0.73 (0.7)	2.64 (2.2)	2.17 (1.46)	7.74 (3.33)
PResNet Triplet Loss	0.84 (1.18)	1.7 (2.69)	**0.31 (1.15)**	2.8 (3.1)

verificação de locutores. O melhor modelo foi o SincNet em relação aos modelos utilizando a PResNet Triplet Loss, GE2E e à combinação (SincNet + GE2E) para quase todas as bases de teste, mesmos B3, onde o modelo PResNet Triplet Loss obteve o melhor resultado.

Os testes de modelos treinados em B1 e testados em B3 indicam que a variabilidade entre as língua inglês e português não causa uma perda de desempenho para os modelos testados neste artigo, uma vez que a perda de desempenho relativa entre B2 e B1 (canais diferentes e línguas semelhantes) é maior que a perda de desempenho relativa entre B3 e B1 (canais semelhantes e línguas diferentes). Estudos como este, sobretudo em relação a bases de dados em português não

Tabela III
EER (%) relativo à base B1

	B2	B3	B4	Média por modelo
GE2E	-233	-31	-304	-142
SincNet	-248	-510	-613	-343
SincNet+GE2E	-262	-197	-960	-355
PResNet Triplet Loss	-102	63	-233	**-68**
Média por base	-211	**-169**	-528	

são realizados com frequência, conforme novos modelos são criados. Trabalhos mais aprofundados deverão ser realizados para obter respostas mais precisas e confiáveis em relação a esses e outros tipos de fontes de variabilidade.

REFERÊNCIAS

[1] J. M. Naik, "Speaker verification: A tutorial," *IEEE Communications Magazine*, vol. 28, no. 1, pp. 42–48, 1990.

[2] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[3] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[4] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[5] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 1021–1028.

[6] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, vol. 650, 2017.

[7] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4052–4056.

[8] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[9] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 1695–1699.

[10] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification." in *Interspeech*, 2017, pp. 999–1003.

[11] J. H. Hansen and H. Bořil, "On the issues of intra-speaker variability and realism in speech, speaker, and language recognition tasks," *Speech Communication*, vol. 101, pp. 94–108, 2018.

[12] J. H. Hansen and V. Varadarajan, "Analysis and compensation of lombard speech across noise type and levels with application to in-set/out-of-set speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 2, pp. 366–378, 2009.

[13] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[14] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," *arXiv preprint arXiv:2012.03411*, 2020.

[15] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *STIN*, vol. 93, p. 27403, 1993.

[16] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[17] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[20] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a"siamese"time delay neural network," *Advances in neural information processing systems*, vol. 6, pp. 737–744, 1993.

[21] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2. Lille, 2015.

[22] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[23] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International workshop on similarity-based pattern recognition*. Springer, 2015, pp. 84–92.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.