Estudo sobre a robustez de técnicas de verificação de locutores com D-vectors

Victor Costa Beraldo e Murilo Bellezoni Loiola

***Resumo*— Recentemente, técnicas de verificação de locutores vêm sendo concebidas por meio de redes neurais profundas por meio dos d-vectors. Neste trabalho, foram realizados experimentos para comparar modelos, em situações onde temos dados para treinamento que não foram obtidos da mesma origem que a base de teste, representando um problema real, onde necessita-se escolher um modelo sem dados de treinamento semelhantes aos de teste. As comparações foram feitas entre os modelos SincNet, GE2E e Redes Triplet Loss. Além disso, foi proposto neste trabalho o modelo SincNet + GE2E, cujo desempenho supera o a rede GE2E original. A SincNet, no entanto, obteve os melhores resultados nas condições presentes.**

***Palavras-Chave*— Verificação de Locutores, Redes Neurais, D-vectors.**

***Abstract*— Recently, speaker verification techniques have been conceived through deep neural networks using d-vectors. In this work, we made experiments to compare different models in situations where we have data for training that were not obtained from the same source as the test data, representing a real problem, where it is necessary to choose a model, without having training data similar to the test data. Comparisons were made with the SincNet, GE2E and Triplet Loss Network models. This work also proposes the SincNet + GE2E model, whose performance is superior to the original GE2E. The SincNet, however, obtained the best performance results in the current scenarios.**

***Keywords*— Speaker Verification, Neural Networks, D-Vectors.**

I. Introdução

A verificação da identidade de uma pessoa é um requisito essencial para controle de acessos. A identidade é 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. 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. Neste trabalho, a biometria de voz é estudada por meio de técnicas de verificação de locutores.

A modelagem de verificadores de locutores, antes dominada pela utilização dos *i-vectors* [2], que utilizam modelos de mistura de gaussianas, atualmente tem sido realizada com a aplicação de redes neurais profundas [3]–[5], utilizando uma estratégia análoga, apelidada de d-vectors [6]. 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, entretanto, 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 para verificação de locutores independente de texto. Incluindo um modelo proposto neste artigo denominado SincNet + GE2E, que combina características dos modelos [3], [4] e apresentou EER menor, comparado ao [3]. 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 são dispostos 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 e trabalhos anteriores. A Seção III apresenta os detalhes sobre a modelagem 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

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*) [7]. 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 por meio de segmentos da fala, denominados *frames*, por meio de técnicas como MFCCs (do inglês *Mel-Frequency Cepstral Coefficients*) [8] ou mesmo diretamente dos áudio por meio de redes neurais profundas [4].

Victor Costa Beraldo, Universidade Federal do ABC (UFABC), Santo André-SP, e-mail: victor.beraldo@ufabc.edu.br; Murilo Bellezoni Loiola, Universidade Federal do ABC (UFABC), Santo André-SP, e-mail: murilo.loiola@ufabc.edu.br.

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, no qual é gerado um valor escalar, medindo a semelhança entre as vozes e, caso supere um certo limite τ, confirma a verificação, ou seja a voz do locutor teste é aceita.

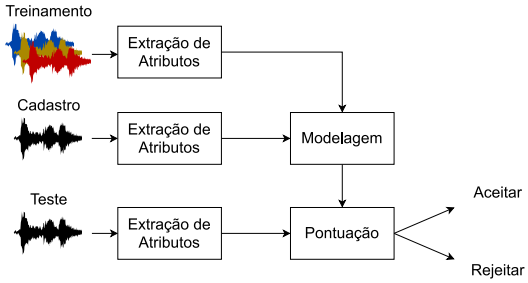

Fig. 1. Arquitetura de um sistema de verificação de locutor simplificado.

III. Modelagem

A. BASES DE DADOS

Os dados utilizados neste trabalho foram organizados de modo a simular uma situação em que 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 duas bases de dados públicas: LibriSpeech [9], TIMIT (do inglês *Texas Instruments Massachussets Institute of Technology*) [10].

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 [9]. A Base B1 foi feita com esta base, a qual foram selecionados áudios subamostrados a 8kHz de 1500 locutores aleatórios, 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 B3 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. A base de treinamento B2 foi criada pela união das bases B1 e B3.

As novas bases criadas B1, B2 e B3 têm suas especificações detalhadas nesta seção e resumidas na 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. A situação padrão é representada pelo treinamento em B3 (Treino), por ser uma base de mesma origem (TIMIT) que a base de Teste, utilizada para todos os experimentos. A base B1 ilustra o caso principal do artigo, no qual há mais dados que a situação padrão, porém extraídos de uma origem diferente (LibriSpeech). A B2 ilustra o contexto no qual há dados da mesma origem do teste, porém utilizando mais dados de outras origens.

TABELA I
ESPECIFICAÇÕES DAS BASES E DADOS UTILIZADAS

Nome	# Locutores	Segmentos de Áudio por Locutor	Origem	Função
B1	1500	4-10	LibriSpeech	Treino
B2	1963	4-10	B1 e B3	Treino
B3 (Treino)	463	10	TIMIT	Treino
B3 (Teste)	168	10	TIMIT	Teste

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 [4], a arquitetura baseada no trabalho GE2E (*Generalized end-to-end loss for speaker verification*) [3] e Redes Triplet Loss [5]. Todos os modelos utilizados foram adaptados para serem utilizados como um método de extração de características por meio 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*), na qual 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*) [3]. 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 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 (so inglês *Long Short-Term Memory*) [11] com uma camada linear conectada ao final desta rede. Sendo a saída da rede dada por $f(x_{ji}; w)$, em que 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 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 S_{ji} é definida como as similaridades cosseno

(s_{cos}) entre o d-vector e todos os centroides c_k ($1 \leq j, k \leq N$ e $1 \leq i \leq M$) e está representada pela Equação (2):

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

na qual w e b são parâmetros que podem ser aprendidos, com restrição no peso $w > 0$. A contribuição de cada d-vector para função de custo para é obtida substituindo a equação (2), na equação Softmax, tornando a saída igual a 1 se $k = j$, e 0 caso contrário (3):

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

A função de custo total, L_G , é a soma de (3) para todos os d-vectors da matriz de similaridade.

$$L_G(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i,j} L(\boldsymbol{e}_{i,j}). \tag{4}$$

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 [3]. 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 [4]. Grande parte de trabalhos passados utilizavam atributos extraídos manualmente, como MFCCs e bancos de filtros [3], [5], [6], [12]. 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 por meio 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 2 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.

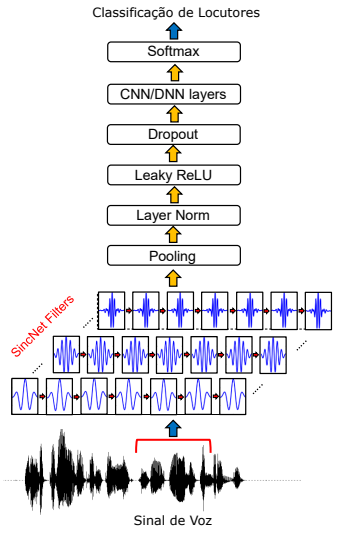

Fig. 2. Arquitetura da SincNet.

Fonte: [4]

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 3 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 cada um. Por meio 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.

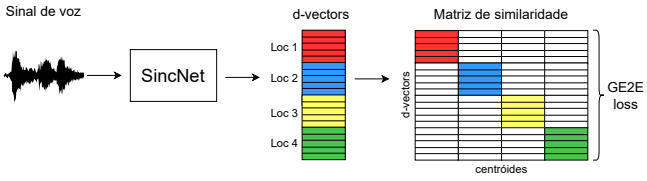

Fig. 3. Arquitetura da SincNet + GE2E.

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 [13] e que têm gerado grandes avanços na tarefa de reconhecimento de imagem [14], [15], causando interesse também na modelagem com dados de voz [5], [16]. 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, em que há muitos dados de locutores diferentes, porém poucos dados do mesmo locutor.

As redes Triplet Loss [17], assim como as siamesas, são projetadas por meio 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 4 ilustra essa abordagem, adaptada para a tarefa de verificação de locutores, na qual 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 [5].

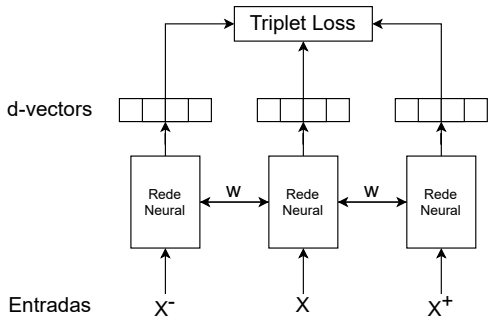

Fig. 4. Arquitetura da Rede Triplet Loss adaptada para verificação de locutores.

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):

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

na qual $d(a,p)$ 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 em que a âncora é próxima tanto da entrada positiva, quanto da negativa.

A modelagem por meio desta arquitetura foi baseada no artigo [5], com o treinamento de uma rede neural convolucional ResNet [18], 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 utilizando esta arquitetura foi dividido em três experimentos: Primeiro a rede foi treinada tradicionalmente com a função de custo entropia cruzada (ResNet EC), depois a mesma rede foi treinada utilizando a fução de custo Triplet Loss (ResNet T) e, no final, foi utilizada também a função de custo Triplet loss, porém com a rede pré-treinada ResNet EC (PResNet T). Para os modelos que utilizaram a função de custo Triplet, foi utilizado um valor de margem de 0,2.

IV. Resultados e Discussão

Todos os resultados apresentados nesta seção foram obtidos utilizando a amostra de teste da base B3 e a métrica EER [1] para avaliação dos modelos.

A metodologia de avaliação dos modelos foi realizada de forma padronizada, a partir das etapas de cadastro e teste, representadas pelas Figuras 5 e 6, respectivamente. Na etapa de cadastro, foram processados áudios da base B3 (Teste), de modo a gerar um d-vector por áudio. Em seguida, são separados 4 áudios, 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 e 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.

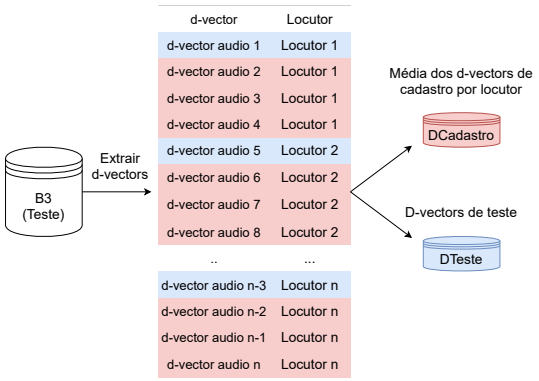

Fig. 5. Descrição do procedimento de Cadastro.

Os resultados, organizados na Tabela II, mostram que o modelo SincNet obteve melhores desempenhos que os modelos GE2E, variando entre as bases de treinamento. A melhor base de dados utilizada para treino foi a B2 para o modelo SincNet, com um EER de 1,8% com dados de áudios tanto da base TIMIT, quanto da LibriSpeech.

O modelo SincNet obteve melhores desempenhos conforme foi utilizado maior número de dados e origens diferentes, dado que o treinamento com a B2, que tem dados de origem semelhante aos dados de teste e também de origem diferente (Librispeech) foi o melhor. Além disso, o modelo treinado com

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

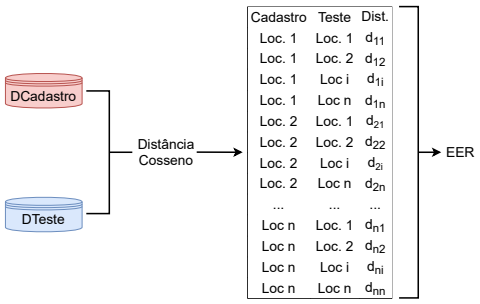

Fig. 6. Descrição do procedimento de Pontuação.

TABELA II
RESULTADOS DE EER PARA OS MODELOS

Treino	SincNet	GE2E	SincNet +GE2E	ResNet CE	ResNet T	PResNet T
B1	0,036	0,107	0,060	0,060	0,101	0,054
B2	0,018	0,077	–	–	–	–
B3	0,042	0,071	–	–	–	–

Experimentos com treino em B1 foram priorizados, uma vez que ele representa o treino com dados de origem diferente do teste. Resultado do treinamento em B2 e B3 para os demais modelos poderão ser feitos em trabalhos futuros.

B1 (mais dados, porém origem diferente) obteve EER de 3,6%, valor menor que o modelo treinado com B3 (menos dados e mesma origem), de 4,2%. O modelo GE2E, de forma diferente, obteve melhores resultados utilizando a base B3, com EER de 7,1%, em relação ao treinamento em B1, que teve EER de 10,7%, indicando ser um modelo em que a origem dos dados semelhantes ao da base de teste é mais importante do que ter mais dados de origens diferentes, para os experimentos deste trabalho.

O modelo SincNet + GE2E obteve uma redução de cerca de 44% de EER em relação ao modelo GE2E e um aumento de aproximadamente 65% de EER em relação ao SincNet. Esse melhor desempenho em relação ao modelo GE2E indica 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 por meio de sua camada convolucional, diferente dos *Filter Banks* utilizado no modelo GE2E.

Os modelos baseados em Redes Triplet Loss mostram um desempenho ligeiramente melhor que o modelo proposto SincNet + GE2E quando, em relação ao mesmo treinamento em B1. O pré-treinamento contribuiu para um melhor desempenho da PResNet T em relação a ResNet T, assim como foi documentado em [5], porém neste experimento a mesma rede treinada com Triplet Loss sem pré-treino (EER de 10,1%) obteve pior desempenho do que a versão tradicional treinada com função de custo Entropia Cruzada (EER de 6,0%).

V. CONCLUSÕES

Os experimentos desenvolvidos neste trabalho mostraram diferentes estratégias de treinamento de modelos de verificação de locutores. O melhor modelo foi o SincNet em relação aos modelos utilizando a Triplet Loss, GE2E e à combinação (SincNet + GE2E).

O modelo SincNet mostrou menos dependência da origem da base de dados, pois conseguiu obter melhor resultado treinando inclusive com base de outra origem (B1) e melhor ainda quando treinado com B2 (B1 + B3). Já o modelo GE2E se mostrou bastante dependente da origem dos dados.

Experimentos utilizando arquiteturas diferentes como a SincNet + GE2E foram testados com objetivo de aproveitar ainda mais a generalização da SincNet e obter melhores desempenhos. Trabalhos futuros poderão ser realizados, desta vez modificando a Triplet Loss com a SincNet pré treinada e aplicando estas arquiteturas propostas nas bases B2 e B3.

REFERÊNCIAS

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

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

[15] 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.

[16] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

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

[18] 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.