UNIVERSIDADE FEDERAL DE SANTA CATARINA

Rede Neural para Identificar Nível de Estresse na Voz: uma abordagem testando parâmetros

Eduardo Kohler

Florianópolis
2020

Eduardo Kohler

Rede Neural para Identificar Nível de Estresse na Voz: uma abordagem testando parâmetros

Trabalho de conclusão de curso submetido como parte dos requisitos para obtenção do título de Bacharel, do curso de Ciências da Computação da Universidade Federal de Santa Catarina.

Orientador: Rafael de Santiago

Florianópolis, Abril de 2021

Resumo

O desenvolvimento de métodos para análise de estresse na voz é relevante para diversas áreas, e envolve conceitos da psicologia, computação e análise comportamental. Ao decorrer do tempo, diferentes sistemas de classificação foram projetados. Desses, os mais antigos contam com a captura de características como respiração e pressão sanguínea, seguida da análise de um operador, responsável por interpretar os dados e sintetizar o resultado. Entretanto, nos últimos anos, fez-se relevante a análise não intrusiva de características extraídas da voz, assim como o uso de classificadores para resultados automáticos. A fim de se obter melhores apurações, como eliminar a necessidade de intervenção humana no processo, considera-se o uso de rede neurais, as quais a partir de treinamento são capazes de detectar padrões e tomar decisões. Esse projeto tem o objetivo de avaliar o uso de LSTMs na tarefa de detecção de estresse na voz, através da análise de diferentes bases de dados e características extraídas da voz. Para isso, foram utilizadas variações de configurações de redes neurais LSTM e o *software* OpenSMILE para extração. Com base nesses procedimentos, foi possível a análise de fatores importantes, como a eficácia de diferentes características, os processos de treinamento para diferentes *datasets* e as consequências da natureza e quantidade de dados utilizados. A partir dos resultados, percebeu-se que características como as potências logarítmicas das bandas de frequência de Mel e os MFCCs são relevantes para a detecção de estresse, e ficou evidenciada a importância de aspectos como quantidade e variabilidade nos dados do processo de treinamento.

Palavras-chave: redes neurais, detecção de estresse, características da voz.

Sumário

1 Introdução

A detecção de estresse na voz se tornou uma importante ferramenta em áreas relacionadas à psicologia para identificar emoções na fala como raiva e felicidade, assim como na computação, onde pode ser utilizada em sistemas de reconhecimento de voz e aplicações com controle por fala (COSETL; LOPEZ, 2011).

Diversos estudos e equipamentos foram desenvolvidos com o objetivo de detectar micro tremores na voz, ou outros fenômenos fisiológicos resultantes de situações de estresse. Dos dispositivos desta área, o mais conhecido é o polígrafo, o qual registra atividade cardiovascular, padrão respiratório e condutividade da pele, a partir da presunção de que respostas verdadeiras e falsas apresentarão diferenças nessas características. Apesar de ser frequentemente reconhecido como uma ferramenta de detecção de mentira, há muitas controvérsias em relação a sua efetividade (SONDHI et al., 2016). Além disso, o polígrafo é incapaz de tomar decisões, sendo necessária a análise humana dos dados registrados. Desta forma, expõe-se a pessoa avaliada ao julgamento de outro ser humano, sujeito a falhas e quebra de sigilo.

Desta maneira, faz-se relevante o desenvolvimento de métodos alternativos para detecção de estresse na voz. Um desses é chamado de VSA (do inglês *Voice Stress Analysis*), o qual consiste em procedimentos para medições, sem contato corporal, de respostas psicológicas involuntárias da voz de uma pessoa que está sob estresse (COSETL; LOPEZ, 2011). A partir disso, é possível treinar modelos para que detectem certos padrões emocionais e consigam inferir previsões sobre os dados analisados, como explicitado no artigo “A neural network approach for human emotion recognition in speech”, de Bhatti, Yongjin Wang e Ling Guan (2004).

Uma das aplicações recentes de detecção de estresse na voz foi desenvolvida por Marcolla, Santiago e Dazzi (2020). Trata-se de um trabalho no qual a partir da extração de características de sinais de voz, foi possível treinar modelos de redes neurais para decidir se uma fala era verdadeira ou não. Os resultados apresentados foram promissores, e evidenciaram a relação entre o estresse na voz e o ato de contar uma mentira, assim como a relevância dos MFCCs como uma característica da voz. No entanto, apesar de ter apresentado metodologias concisas, o trabalho fez uso de poucas amostras, que não eram muito diversas entre si, e utilizou apenas uma característica extraída da voz.

Esse trabalho objetiva avaliar a utilização de LSTMs e diferentes parâmetros extraídos da voz na tarefa de detecção de estresse na voz, a partir de base de dados públicas e os dados coletados no trabalho de Marcolla, Santiago e Dazzi (2020).

1.1 Objetivos

O presente trabalho tem como objetivo geral avaliar o uso de LSTMs na tarefa de detecção de estresse na voz, através da análise de diferentes bases de dados e características extraídas da voz.

Os objetivos específicos são:

- Especificar as características da voz e base de dados utilizadas;
- Obter os dados para treinamento, validação e testes e realizar pré-processamento;
- Desenvolver as redes neurais LSTM;
- Realizar experimentos com as características extraídas e as bases de dados especificadas;
- Analisar os resultados obtidos nos experimentos;
- Divulgar os resultados.

1.2 Método de Pesquisa

Este trabalho seguiu as seguintes etapas:

1. Realização de pesquisas a partir de artigos e livros da área de redes neurais e aprendizado de máquina, de maneira a formar uma base de referências e exemplos, para então ser possível abordar assuntos mais específicos.

2. Estudo com o objetivo de definir as características da voz a serem utilizadas como entradas nas redes, e também como obtê-las a partir das gravações de áudio.

3. Análise de ferramentas e bibliotecas para a implementação das redes neurais e seus distintos modelos, assim como para o tratamento, processamento e normalização dos dados de entrada do sistema.

4. Obtenção dos dados de entrada.

5. Implementação dos modelos e o treinamento das redes neurais.

6. Realização de comparações e considerações relevantes, inclusive em relação ao trabalho de Marcolla, Santiago e Dazzi (2020).

7. Documentação do trabalho desenvolvido.

1.3 Estrutura do Documento

O capítulo 1 desse trabalho apresenta uma introdução ao tema de detecção de estresse na voz, assim como os objetivos e metodologia de pesquisa. O segundo capítulo explicita conceitos da análise de sinais da voz para detecção de estresse, além de métodos e definições da área de inteligência artificial. Já no capítulo 3, são citados trabalhos que utilizaram de inteligência artificial para detectar estresse na voz, com um comparativo sobre as técnicas e dados utilizados em cada artigo. Em seguida, no quarto capítulo são descritos os principais conceitos desenvolvidos como base para as implementações. No capítulo 5 estão os experimentos realizados e os resultados obtidos, com discussões e análises pertinentes ao contexto do trabalho. Finalmente, no capítulo 6 estão as conclusões e sugestões para trabalhos futuros.

2 Fundamentação Teórica

Nesse capítulo são apresentados conceitos da detecção de estresse pela voz pertinentes ao presente trabalho, mais especificamente, fundamenta-se quais as características da voz pode-se extrair para eventual análise, e quais os métodos de classificação presentes na literatura. Além disso, são conceituados métodos de inteligência artificial, e discutidas as configurações das bases de dados públicas de vozes anotadas.

2.1 Estresse

"Estresse" é uma palavra que geralmente se refere a experiências que causam sentimentos de ansiedade e frustração. Os causadores de estresse mais comuns são aqueles que operam cronicamente, geralmente em níveis baixos, como acontecimentos do dia a dia. No entanto, há também situações extremas que evocam a resposta de "luta ou fuga" no corpo humano, as quais, ao contrário dos aborrecimentos diários, são estressores agudos, portanto suas consequências podem ser mais facilmente detectadas (MCEWEN, 2006).

O responsável pelas mudanças comportamentais e fisiológicas geradas pelo estresse é o sistema nervoso autônomo, o qual pode ser dividido em duas seções, os sistemas simpático e parassimpático. É através do sistema simpático que o corpo humano ativa glândulas e órgãos para defender o corpo de ameaças, ocasionando também reações como aumento da frequência cardíaca, fluxo sanguíneo rápido para os músculos, ativação das glândulas sudoríparas e aumento da frequência respiratória (KURNIAWAN; MASLOV; PECHENIZKIY, 2013).

Um dos modelos mais comumente usado para definir essas mudanças é o de Selye, o qual define o estresse como uma perturbação da homeostase causada por estímulo físico ou psicológico (SELYE, 1951). O modelo também propõe três diferentes estágios na resposta por estresse. O primeiro estágio é denominado "estágio de alerta", no qual o corpo identifica o estressor e entra em estado de alarme, acarretando a liberação de adrenalina e cortisol na corrente sanguínea, responsáveis por levar fluxo sanguíneo aos maiores músculos do corpo e aumentar a pressão sanguínea, respectivamente. O segundo estágio é o de "resistência", nesse, o corpo utiliza de recursos para lidar com o estresse a partir de adaptação. Após essa utilização de recursos, o corpo entra, então, no terceiro estágio, chamado de "exaustão", no qual esse é incapaz de manter o funcionamento normal. O primeiro e segundo estágios são particularmente relevantes, uma vez que causam tensão muscular, um importante fator para classificação e identificação do estresse (ZHANG et al., 2009).

Como explicitado no artigo "Voice analysis for detection of deception", de Sondhi et

al. (2016) um exemplo de estressor é o ato de contar uma mentira, cujos efeitos geralmente se manifestam no psicológico e no físico, em forma de expressões faciais, linguagem corporal ou voz.

2.1.1 Estresse na voz

A voz humana pode efetivamente sinalizar o estado psicológico de um indivíduo, seja emocional, físico, intencional ou inconsciente (SONDHI et al., 2016). Embora a fala seja uma atividade vocal da qual muito seja verbal, há várias vocalizações humanas que são essencialmente não linguísticas, como entonação, qualidade da voz, prosódia, ritmo e pausa. Esses fenômenos representam um sistema de sinalização não verbal, que se entrelaça com o sistema verbal ou linguístico, trazendo entre outras coisas, informações sobre os aspectos fisiológicos e estado psicológico do falante (ROTHKRANTZ; WEES; VARK, 2004). São esses fenômenos que sofrem as alterações mais relevantes em situações de estresse. Como já citado, umas das respostas do corpo humano a estressores é a tensão muscular, a qual influencia também as cordas vocais, podendo então alterar direta ou indiretamente a produção de fala (ZHANG et al., 2009).

Streeter et al. (1977), no artigo "Pitch Changes During Attempted Deception", evidencia um tipo de manifestação do estresse na voz. No experimento expresso no artigo, mudanças de tom na voz de entrevistados foram observadas entre afirmações falsas e verdadeiras. As medições foram possíveis através da análise da frequência fundamental da voz, a qual se refere à vibração das cordas vocais por unidade de tempo, e pelo fato de refletir a eficiência do sistema fonatório, é considerado um importante parâmetro na avaliação anatômica desta área (TEIXEIRA; FERREIRA; CARNEIRO, 2011). A partir disso, observou-se que quando as afirmações eram falsas, a frequência fundamental da voz do entrevistado era maior, além de proporcional à quantidade de estresse vinculada ao ato de mentir. Enquanto reconhece-se que mentir nem sempre resulta em estresse vocal (devido a condições sociopatas, estresse silenciado por certas substâncias químicas e assim por diante), acredita-se que esse ainda está presente na maioria dos casos (HARNSBERGER et al., 2009).

Outra manifestação relevante são as perturbações na voz chamadas de *jitter* e *shimmer*, as quais podem ser medidas a partir de variações na frequência e amplitude, respectivamente (CARROLL, 2011). Os experimentos realizados por Mendoza e Carballo (1998) no artigo "Acoustic Analysis of Induced Vocal Stress by Means of Cognitive Workload Tasks" registraram variações nesses parâmetros quando os entrevistados se encontravam em situações de estresse, como a realização de tarefas rapidamente e trava-línguas. Os dados coletados evidenciaram que os jitter e shimmer diminuem em contextos de estresse.

O fenômeno chamado de *jitter* também pode ser referenciado como microtremor, sendo esse um parâmetro bastante discutido em estudos e, portanto, um potencial indicador

de estresse. Zhang et al. (2009) afirmam que através do sinal da voz e métodos de decomposição empíricos adaptativos foi possível detectar variações em tais microtremores da voz, analisando falas de base de dados públicas e entrevistas específicas do estudo. No entanto, no artigo "Voice Stress Detection: A method for stress analysis detecting fluctuations on Lippold microtremor spectrum using FFT", de (COSETL; LOPEZ, 2011), após a demodulação da voz e processamento com FFT (do inglês *Fast Fourier Transform*), concluiu-se que não foi possível observar variações claras de microtremores nos dados analisados. Apesar de estar relacionado com controle muscular e produção de fala, ainda não é certo de que forma ou em qual grau o estresse pode se manifestar em microtremores na voz (SONDHI et al., 2016).

2.2 Métodos para Detecção de Estresse

O estresse pode ser detectado através de vários métodos e técnicas, uma vez que sua manifestação pode resultar em diferentes fatores. Alguns dos parâmetros considerados nesse quesito são: expressão facial, mudanças na voz, manifestações comportamentais e emocionais, características físicas e sinais fisiológicos (Shanmugasundaram et al., 2019). A maioria desses efeitos podem ser medidos e classificados através de sensores e tecnologias modernas, de formas não intrusivas, as quais já foram investigadas extensivamente nas décadas passadas (KURNIAWAN; MASLOV; PECHENIZKIY, 2013).

Para a detecção a partir de manifestações comportamentais, por exemplo, utiliza-se de características extraídas de vídeos. Já a detecção por características físicas geralmente se dá por monitoramento e reconhecimento facial. Em relação aos sinais fisiológicos, pode-se citar a detecção de alteração nos batimentos cardíacos, na temperatura do corpo, na umidade da pele e pressão sanguínea, os quais são analisados através de um sensor específico para cada parâmetro (KURNIAWAN; MASLOV; PECHENIZKIY, 2013).

No que se refere à detecção de estresse na voz, há mais de um parâmetro relevante, e diversas técnicas de análise surgiram com o passar do tempo e avanço tecnológico. No passado as principais características analisadas eram as relacionadas à prosódia, como entonação, duração e intensidade. Nos últimos 20 anos, a maioria dos estudos utilizam de LLDs (do inglês *Low Level Descriptors*) para extrair informações da fala, como a frequência fundamental, jitter e shimmer, energia, e outros (SCHULLER; STEIDL; BATLINER, 2009).

Há muitos dispositivos comerciais que afirmam detectar variações causadas por estresse, sendo o polígrafo o mais conhecido. Esse usa eletrodos, manguitos de pressão arterial e medidores pneumáticos para registrar atividades cardiovasculares, frequência respiratória e condutância da pele. Embora o polígrafo seja aceito em alguns países como uma ferramenta de detecção de fraude, há muitas controvérsias em relação à sua eficácia

(SONDHI et al., 2016).

Existem também aparelhos que buscam analisar somente parâmetros da voz, sem intrusão, com o objetivo de detectar estresse e potenciais mentiras em falas analisadas. Esses são chamados de dispositivos de VSA (do inglês *Voice Stress Analysis*), os quais utilizam de técnicas como LVA (do inglês *Layered Voice Analysis*) com o objetivo de detectar variações nos microtremores da voz. No artigo "Stress and Deception in Speech: Evaluating Layered Voice Analysis", Harnsberger et al. (2009) fez uma avaliação de dispositivos que afirmam usar essa técnicas e, a partir de entrevistas gravadas, concluiu que os resultados não foram bons o suficiente para comprovar a eficiência prometida.

Apesar de os dispositivos de VSA e o polígrafo funcionarem de maneiras diferentes, ambos utilizam de técnicas de entrevista semelhantes para obtenção de dados, através de vários tipos de testes. Independente da tecnologia utilizada, é importante a utilização de protocolos específicos para obtenção de padrões nas entrevistas realizadas. Isso significa que cada sessão de teste deve seguir um padrão específico, com o objetivo de realizar um estudo realista dos sinais de estresse em um ambiente controlado. Se nenhuma diretriz for seguida, as análises serão expostas a erros causados por outros estressores, como medo de reprovar nos testes, fornecendo resultados incorretos (COSETL; LOPEZ, 2011).

2.2.1 Detecção de estresse na voz

A detecção de estresse pela voz é um caso específico da detecção de emoções, e para isso, há mais de um método adequado, e diversos parâmetros a serem analisados. Dentre esses parâmetros, o com maior consenso e mais utilizado é a frequência fundamental, ou F0 (JULIAO et al., 2015). Entretanto, métricas para medição de energia e frequências como os MFCC (do inglês *Mel-frequency cepstrum coefficients*) também foram propostas e analisadas, como no artigo de Marcolla, Santiago e Dazzi (2020). Outro parâmetro relevante é o TEO (do inglês *Teager Energy Operator*), o qual, de acordo com Zhou, Hansen e Kaiser (2001), é o parâmetro que melhor consegue refletir a estrutura da fala sob condições de estresse. Apesar de serem usados para o mesmo propósito, os diferentes métodos de análise de características da voz podem derivar de diferentes modelos. Métricas como MFCC, por exemplo, derivam de um modelo de produção de fala linear, enquanto de acordo com a teoria por trás dos TEO, a origem dos sons no contexto da fala são interações não lineares (ZUO; FUNG, 2011).

Um dos conceitos adequados para a análise da voz reside na utilização de métodos para transformar o sinal da fala em um domínio de frequências, que então podem ser comparadas de acordo com o estado do indivíduo analisado. No artigo "Voice Stress Detection: A method for stress analysis detecting fluctuations on Lippold microtremor spectrum using FFT", de Cosetl e Lopez (2011), usou-se FFT sobre o sinal da voz demodulado e um algoritmo para detecção de frequências dominantes, e através desses foi

possível observar que os componentes de frequência entre 8 e 12 Hz apresentaram uma diminuição de magnitude quando uma pessoa estava sob estresse.

Zuo e Fung (2011) no estudo "A Cross Gender And Cross Lingual Study On Acoustic Features For Stress Recognition In Speech", observaram parâmetros como MFCCs, TEOs e F0 (frequência fundamental) e compararam seus comportamentos na voz humana sob estresse. Após análise e classificações, os autores chegaram à conclusão que os parâmetros mais precisos para medição de estresse foram os MFCCs e TEOs. Além disso, concluiu-se que os índices de acurácia na classificação aumentaram em sistemas dependentes do gênero do entrevistado.

2.2.2 Métodos automatizados

Após a extração de sinais e parâmetros da voz, o próximo passo para a detecção de estresse é a classificação e eventual decisão sobre a presença ou não de estresse na voz. Para isso, existem diferentes métodos e algoritmos, a seguir serão citados alguns dos mais abordados em estudos relacionados.

Artigos como os de Juliao et al. (2015), Kurniawan, Maslov e Pechenizkiy (2013) e Zuo e Fung (2011) utilizaram de SVMs (do inglês *Support Vector Machine*) para classificar e obter decisões sobre as características extraídas da voz. SVM é um conceito desenvolvido no âmbito da teoria de aprendizado estatístico, e é utilizado em áreas como reconhecimento facial e processamento de dados biológicos para diagnóstico médico, por exemplo. O funcionamento de uma SVM é o seguinte: a partir de um conjunto de dados de treinamento, e da medição da disparidade entre os valores esperados e os previstos pela máquina, busca-se por uma função que minimize essa diferença (EVGENIOU; PONTIL, 2001). No caso da detecção de estresse na voz, utiliza-se dos parâmetros extraídos de falas para o treinamento da SVM, a qual é responsável por classificar e inferir sobre os áudios analisados.

Outro método de classificação é exposto nos artigos de Marcolla, Santiago e Dazzi (2020) e Han, Byun e Kang (2018), os quais fazem uso de redes neurais artificiais com arquitetura LSTM (do inglês *Long Short-Term Memory*) para tirar conclusões sobre os dados extraídos. As redes neurais artificiais funcionam a partir de treinamento com dados, assim como SVMs, mas diferem em alguns fatores, como estrutura e quantidade de dados necessária, por exemplo.

Além disso, pode-se citar o método aplicado no trabalho de Gulhane, Rode e Ladhake (2011), no qual utilizou-se de Lógica Fuzzy e uma estrutura semelhante a uma rede neural, também com objetivo de classificar a presença de estresse na voz, nesse caso a partir de áudios obtidos em entrevistas. Através do mapeamento de entradas e saídas por funções de associação e parâmetros associados, e do processo de aprendizado do sistema,

observou-se que a Lógica Fuzzy também é viável para esse tipo de classificação.

2.2.3 Considerações

A partir das informações presentes na literatura, pode-se observar que os chamados *low-level descriptors* são importantes fatores na detecção de emoção na voz. Alguns desses parâmetros são: MFCCs, TEOs e F0, por exemplo. Em relação à classificação, métodos como SVMs e RNAs são amplamente discutidos, com resultados relevantes com ambas as implementações.

2.3 Redes Neurais Artificiais

O desenvolvimento dos sistemas denominados redes neurais artificiais surgiu a partir da observação do funcionamento do cérebro humano, o qual processa informações de maneira totalmente diferente de um computador convencional. A partir da organização e conexão das unidades de processamento chamadas "neurônios", o cérebro é capaz de adquirir informação e acumular conhecimento com o tempo. Com isso, desenvolve-se habilidades como reconhecimento de padrões e coordenação motora, por exemplo (HAYKIN, 2008).

Uma das características mais importantes das redes neurais é a plasticidade, ou seja, a capacidade de se desenvolver de acordo com as exigências do ambiente, e essa característica faz-se presente também nas redes neurais artificiais (HAYKIN, 2008). Sistemas computacionais atualmente lidam com grandes volumes de informação constantemente, com isso, faz-se relevante o aprendizado a partir de dados distribuídos. A habilidade de aprender continuamente com o passar do tempo, junto com a retenção de conhecimento adquiridos previamente é um dos maiores desafios na área de aprendizado de máquina e redes neurais artificiais (PARISI et al., 2018).

Para atingir esse aprendizado, as redes neurais contam com a interconexão massiva de unidades de processamento, ou "neurônios". No caso do cérebro humano estima-se que há cerca de 10 bilhões de neurônios e 60 trilhões de conexões ou sinapses, o que gera um sistema extremamente eficiente. Apesar de o volume de estruturas ser menor nas redes neurais artificiais, e o modelo de neurônio ser primitivo se comparado ao do cérebro, há similaridades no que se refere a como o conhecimento é adquirido, através do processo de aprendizado, e como o conhecimento é armazenado, a partir de pesos sinápticos (HAYKIN, 2008).

O modelo de neurônio de uma rede neural artificial é composto por três elementos: um conjunto de pesos sinápticos, um somador para as entradas e uma função de ativação para restringir a amplitude da saída. Os pesos sinápticos se referem aos valores pelos quais as entradas são multiplicadas, de forma a modelar um peso para cada entrada em cada neurônio e, diferente de no cérebro humano, os pesos sinápticos das redes neurais artificiais

podem estar em intervalos positivos ou negativos. O somador pode ser expresso como um combinador linear, com os sinais de entrada ponderados pelos pesos sinápticos. A função de ativação é responsável por restringir a saída a um valor finito, geralmente representada como um intervalo fechado [0,1] ou [-1,1] (HAYKIN, 2008).

A partir dessas estruturas, o processo utilizado para a aprendizagem é chamado "algoritmo de aprendizagem", o qual é responsável por modificar os pesos sinápticos de forma a alcançar o objetivo da rede neural artificial. O tipo de algoritmo utilizado está diretamente relacionado com a arquitetura da rede neural, referente ao número de camadas de neurônios e como são organizadas e conectadas. Redes neurais artificiais podem ter uma única camada de unidades processadoras assim como múltiplas, no último caso, as camadas intermediárias são denominadas "camadas ocultas"(HAYKIN, 2008). À medida em que cresce o número de camadas, cresce também a complexidade e profundidade da rede e, com isso, obtém-se mais níveis de representação de dados e maior extração de características. Esse comportamento é conhecido como *deep learning* (SALEHINEJAD et al., 2017). De modo geral, pode-se citar três estruturas de redes neurais, as alimentadas adiante com camada única, as alimentadas diretamente com múltiplas camadas, e as redes recorrentes (HAYKIN, 2008). Nesse trabalho será utilizada a arquitetura de redes neurais recorrentes, mais especificamente a LSTM.

2.3.1 Redes Neurais Recorrentes

As redes neurais recorrentes se destacam por conter laços de realimentação, a partir dos quais saídas de neurônios podem fazer parte das entradas de neurônios de camadas inferiores. A presença desse tipo de conexão tem uma influência profunda no desempenho e capacidade de aprendizagem (HAYKIN, 2008). Os neurônios com esse comportamento se encontram nas camadas ocultas das redes neurais desse padrão, formando a estrutura que age como a memória e o estado do sistema, a qual é dependente do estado anterior. Isso permite que a rede armazene e memorize sinais anteriores por um longo período de tempo (SALEHINEJAD et al., 2017). Um tipo de rede neural que utiliza dessa arquitetura é a LSTM.

2.3.2 Long Short-Term Memory (LSTM)

As redes neurais LSTM foram propostas por Hochreiter e Schmidhuber (1997) com o objetivo de evitar que sinais importantes desapareçam da memória do sistema com o passar do tempo, o que pode comprometer a rede neural caso essa necessite de uma informação de um passado distante. Esse é um problema comum nas redes neurais recorrentes, denominado “problema do desaparecimento do gradiente”. Para mitigar esse comportamento, as LSTM utilizam de unidades de processamento chamadas de "células de memória", as quais através de “portões” que manipulam a memória, lembram ou esquecem

informações ao longo do tempo. Cada célula contém três tipos de portões em sua estrutura, um *forget gate*, um *input gate* e um *output gate*. O *forget gate* é responsável por remover as informações que não são mais úteis ao estado da célula, o *input gate* realiza a adição de informação ao estado, e o *output gate* extrai informações úteis do estado da célula para formar um sinal de saída (HOCHREITER; SCHMIDHUBER, 1997).

2.4 Bases de Dados sobre Voz Anotadas

Atualmente há diversos laboratórios de pesquisa e empresas com o objetivo de desenvolver soluções na área detecção de emoções na fala. Para que esse desenvolvimento aconteça, a existência de bases de dados de vozes é de grande importância (VERVERIDIS; KOTROPOULOS, 2012). Além disso, é relevante a análise sobre a modelagem e qualidade de tais bases de dados, uma vez que conclusões incorretas podem aparecer caso uma base de má qualidade seja usada. Alguns dos critérios para o julgamento sobre uma base de dados são a natureza das emoções expressas, as características das pessoas entrevistadas, como idade, gênero e nacionalidade e quais os tipos de frases ou palavras utilizadas (El Ayadi; KAMEL; KARRAY, 2011).

Há três tipos de bases de dados para sistemas de reconhecimento de emoções na fala: as com emoções naturais, as com emoções atuadas e as com emoções induzidas. As bases naturais são aquelas que contém falas reais e espontâneas, como gravações de *callcenters*, interações entre paciente e médico e gravações durante situações anormais. As bases com emoções atuadas são aquelas que possuem gravações coletadas a partir de atores profissionais, os quais reproduzem emoções nas falas de maneira simulada. Por último, as bases com emoções induzidas utilizam de métodos para influenciar os entrevistados a expressarem certas emoções. As emoções básicas geralmente usadas na literatura são: raiva, medo, tristeza, prazer sensorial, diversão, satisfação, contentamento, excitação, nojo, desprezo, orgulho, vergonha, culpa, constrangimento e alívio (SWAIN; ROUTRAY; KABISATPATHY, 2018). Além disso, sinais complementares como pressão sanguínea, batimentos cardíacos e respiração também podem ser documentados (VERVERIDIS; KOTROPOULOS, 2006).

Alguns exemplos de base de dados de falas são: Belfast Natural Database, Kids' Audio Speech Corpus NSF/ITR Reading Project, Magdeburger Prosodie Korpus, SUSAS (Speech Under Simulated and Actual Stress), SAVEE (Surrey Audio-Visual Expressed Emotion) e RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song). A Tabela 1 apresenta as principais características de cada base de dados citada.

Tabela 1 – Comparação entre bases de dados

Base de dados	**Idioma**	**Nº de amostras**	**Tipo de amostra**	**Origem**
Belfast Natural Database	Inglês	239	Entrevistas e conversas	Queen's University
Kids' Audio Speech Corpus	Inglês	1.000	Expressões	University of Colorado
Magdeburger Prosodie Korpus	Alemão	4.200	Expressões	Leib-niz Institute of Neurobiology
SUSAS	Inglês	16.000	Expressões	University of Colorado Boulder
SAVEE	Inglês	480	Expressões	University of Surrey
RAVDESS	Inglês	1.440	Expressões	Ryerson University

Fonte: o Autor

3 Trabalhos Relacionados

Os trabalhos similares referenciados nesse projeto foram reunidos através da ferramenta SCOPUS, utilizando os termos *voice/speech stress analysis*, *voice/speech stress detection* e *voice/speech stress* como palavras-chave para busca. Os critérios para escolha foram: a utilização de redes neurais para classificação e a detecção de estresse exclusivamente através da voz.

3.1 Stress Detection Through Speech Analysis

No artigo de Tomba et al. (2018) foram analisados parâmetros para a detecção de estresse na voz, assim como métodos para a classificação. Os parâmetros escolhidos foram características do sinal de áudio gerado pela voz, uma vez que é um processo não intrusivo e extensamente abordado na literatura relacionada. As características analisadas foram energia média, intensidade média e MFCCs, e as estratégias de classificação testadas foram SVMs e RNAs.

Para a extração da energia média e da intensidade média, utilizou-se um software de processamento de voz, chamado Praat, e uma biblioteca Python. De acordo com o autor, os resultados encontrados com os dois métodos não foram iguais, porém seguiam uma tendência para todos os exemplares de áudio testados. Em relação aos MFCCs, usou-se as seguintes etapas para extração: segmentação dos arquivos de áudio em tamanhos menores (*windowing* e *framing*), aplicação de FFT, aplicação de bancos de filtros e então a etapa de MFCC. A saída desse processo é uma matriz NxM, com N sendo o número de segmentos obtidos na etapa de segmentação e M o número de coeficientes, nesse caso, 13. Apesar de o método utilizado permitir a obtenção de um número maior coeficientes, somente os primeiros 13 foram utilizados, uma vez que os restantes eram referentes a detalhes não relevantes para a detecção de estresse. Essas etapas de extração aconteceram a partir de bibliotecas em Python.

Os áudios utilizados no projeto foram obtidos a partir de três bases de dados, a Berlin Emotional Database (EmoDB), a Keio University Japanese Emotional Speech Database (KeioESD) e a Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). A primeira contem cerca de 500 arquivos de áudio, com duração variando entre 2 e 4 segundos, gravados por 10 atores expressando 7 emoções diferentes em alemão. A segunda se trata de um conjunto de 19 palavras com 47 emoções distintas expressas por um orador masculino em japonês, com duração média de 0.5 segundos por arquivo. A última contem cerca de 1000 amostras de frases faladas por 24 oradores masculinos e femininos, com uma média de aproximadamente 3 segundos por arquivo.

A partir dos dados obtidos das bases de dados, foram preparados 4 conjuntos de parâmetros, com o objetivo de determinar qual a maneira mais apropriada de se lidar com os MFCCs. Os conjuntos formados foram:

- Conjunto 1: média de intensidade e média de energia;
- Conjunto 2: média de intensidade, média de energia e MFCCs;
- Conjunto 3: MFCCs;
- Conjunto 4: media de intensidade, média de energia, média de MFCCs e desvio padrão de MFCCs.

Apesar de se tratar de uma análise para classificação de estresse, testou-se conjuntos com cinco emoções (felicidade, desgosto, tristeza, ansiedade/estresse e raiva) além de conjuntos com duas emoções (ansiedade/estresse e ausência de ansiedade/estresse). Para a classificação, foram utilizadas RNAs e SVMs, com 75% dos dados utilizados na etapa de treinamento e 25% para os testes. A partir dos testes, observou-se que com os dados da base KeioESD e classificação multi classe, o conjunto 3 foi o mais eficiente no caso das SVMs, com uma acurácia de 83,3%, e o conjunto 4 foi o mais eficaz com as RNAs, com 70,83% de acurácia. Com os dados das outras duas bases, o último conjunto foi o de maior acurácia, tanto na classificação multi classe, quanto na binária.

Dessa forma, concluiu-se que os MFCCs são bons parâmetros para medição de estresse por voz, principalmente quando considerados a média e desvio padrão. Além disso, os autores propuseram como melhorias para o trabalho exposto o aumento do número de características da voz, assim como a utilização de bases de dados mais amplas, incluindo casos reais de estresse.

3.2 A Deep Learning-based Stress Detection Algorithm with Speech Signal

Nesse artigo, Han, Byun e Kang (2018) propuseram um algoritmo para detecção de estresse pela voz utilizando *deep learning*. Para isso, combinou-se uma rede neural LSTM e dois tipos de classificadores, utilizando de coeficientes *mel-filterbank* extraídos da voz como os parâmetros a serem analisados. Os classificadores usados foram uma camada SVM e uma camada *softmax*, as quais tiveram os resultados comparados posteriormente.

Em relação à aquisição de dados anotados, os autores realizaram um processo de entrevista com 56 pessoas, capturando voz, video e sinais biológicos. Para os excertos sem a presença de estresse, os entrevistados estavam em um ambiente confortável, e após assistir um vídeo relaxante, liam um roteiro em coreano. Para a captura de fragmentos com a

presença de estresse, antes da leitura do roteiro, os entrevistados tiveram que responder perguntas em inglês feitas por entrevistadores não coreanos.

O conjunto de dados utilizado nos experimentos foram fragmentos de 4 segundos das gravações de 25 dos 56 entrevistados. Foram utilizados 5 fragmentos com presença de estresse e 5 fragmentos sem, totalizando 10 por entrevistado. Com isso, a extração das características da voz consistiu em aplicar métodos de pré-processamento nos arquivos de áudio, e então a utilização de bancos de filtros para a obtenção dos coeficientes *mel-filterbank*.

O modelo proposto é formado por duas camadas LSTM e uma camada totalmente conectada. As camadas LSTM são responsáveis por calcular a saída em nível de *frames* de 25ms, e as camadas totalmente conectadas calculam a saída em nível do fragmento inteiro. Mais especificamente, a análise de um trecho de áudio acontece a partir do processamento de *frames* de 25ms, cujos coeficientes *mel-filterbank* são introduzidos na camada LSTM e então convertidos para um parâmetro referente ao trecho inteiro. Esse parâmetro pode ser a média das saídas da camada LSTM, ou o valor da saída da LSTM referente ao último frame analisado. Esse valor é então alimentado à camada totalmente conectada, e saída dessa é consumida pelo classificador, que pode ser uma camada softmax ou uma camada SVM.

Os experimentos foram realizados em quatro categorias:

- Categoria 1: LSTM-Softmax e média dos frames;
- Categoria 2: LSTM-Softmax e último frame;
- Categoria 3: LSTM-SVM e média dos frames;
- Categoria 4: LSTM-SVM e último frame.

Dentre esses, o que obteve maior acurácia foi o LSTM com SVM e média dos frames, com 66,4% de precisão. Observou-se também que a utilização da média dos frames, em vez do último frame, é a alternativa mais eficiente. Para aumentar a acurácia em trabalhos futuros, os autores propuseram uma abordagem multimodal.

3.3 Speech-Based Stress Classification based on Modulation Spectral Features and Convolutional Neural Networks

O trabalho exposto nesse artigo, de Avila et al. (2019), tem como principal objetivo validar a utilização de uma rede neural convolucional (CNN, do inglês *convolutional neural network*) para detecção de estresse na voz. Para isso, usou-se características espectrais de modulação e a base de dados Speech Under Simulated and Actual Stress (SUSAS).

Os resultados obtidos pelo modelo proposto foram comparados com outros dois sistemas, um desses com classificador SVM, e outro com uma rede neural profunda (DNN, do inglês *deep neural network*). Ambos utilizaram características extraídas com a ferramenta OpenSMILE.

Para a extração das características espectrais de modulação, primeiramente normalizou-se o sinal de áudio com o objetivo de eliminar variações de energia indesejadas. Em seguida, os sinais foram sujeitos a um filtro de bancos que simula o processamento realizado na cóclea, e após o dimensionamento de *frames* de 256ms, utilizou-se de mais um banco de filtros baseado no sistema auditivo humano. A partir desse processo, e com a utilização de conceitos de agrupamento estatístico, os autores chegaram a 5 conjuntos de características do sinal da voz.

O modelo usado foi a rede neural convolucional. Esse padrão de rede neural normalmente é composto por uma camada de convolução, responsável por mapear características detectadas nas conexões locais das camadas anteriores, e uma camada de agrupamento, a qual reduz o dimensionamento agrupando características similares.

A realização dos testes se deu a partir de dados da base SUSAS, a qual contém gravações com estresse simulado e estresse real. Nove condições de estresse estão documentadas na base: neutro, zangado, alto, suave, lento, efeito Lombard (ruído rosa), rápido e fala produzida em dois níveis de carga de trabalho: baixa e alta. Em relação ao dados utilizados no artigo em questão, optou-se por utilizar somente áudios com estresse simulado, uma vez que as gravações com estresse real tratam-se de aúdios de pilotos de helicóptero, ou de pessoas em montanhas russas, portanto o nível de ruído e som ambiente são muito elevados.

Os experimentos foram feitos com três tipos de classificação: duas classes (zangado e neutro), quatro classes (neutro, zangado, suave e rápido) e nove classes (as nove condições de estresse documentadas na base). O modelo proposto teve o desempenho comparado com os classificadores SVM e rede neural profunda, utilizando características do OpenSMILE e também as características espectrais de modulação extraídas pelos autores. Os resultados encontrados estão na Tabela 2.

Pode-se observar que o classificador CNN obteve bons resultados, e foi o menos afetado com o aumento do número de classificações. Outro fator relevante foi que o DNN superou a SVM em todas as categorias, e na classificação com 9 classes, foi mais eficiente utilizando os coeficientes espectrais de modulação em comparação aos recursos do OpenSMILE. Com isso, os autores concluíram que os coeficientes espectrais de modulação podem ser úteis para a detecção de estresse, principalmente com a utilização de redes neurais convolucionais e redes neurais profundas.

Tabela 2 – Resultados do trabalho de Avila et al. (2019) comparados

Classifier	Features	2 Classes	4 Classes	9 Classes	Average
SVM	OpenSMILE	68 %	57 %	58 %	61 %
	$\mathcal{E}_1$	61 %	53 %	44 %	52 %
	$\mathcal{E}_2$	63 %	53 %	46 %	54 %
	$\mathcal{E}_3$	63 %	54 %	49 %	56 %
	$\mathcal{E}_4$	63%	53 %	50 %	54 %
DNN	OpenSMILE	**83 %**	**77 %**	58 %	**72 %**
	$\mathcal{E}_1$	67 %	62 %	55 %	52 %
	$\mathcal{E}_2$	69 %	63 %	58 %	61 %
	$\mathcal{E}_3$	74 %	67 %	57 %	63 %
	$\mathcal{E}_4$	75 %	67 %	62 %	68 %
CNN	$\mathcal{E}_5$	76 %	71 %	**70 %**	**72%**

Fonte: Avila et al. (2019)

3.4 Novel Lie Speech Classification by using Voice Stress

O trabalho exposto no artigo de Marcolla, Santiago e Dazzi (2020) analisou a utilização de redes neurais para a detecção de mentiras pela voz. A arquitetura considerada foi a rede neural recorrente LSTM, e as características extraídas da voz foram os MFCCs.

Os dados utilizados no projeto foram coletados a partir de entrevistas com 10 pessoas do sexo masculino. Os entrevistados foram submetidos a uma série de perguntas, as quais foram respondidas com respostas verdadeiras e falsas. Os áudios foram então processados e segmentados para conterem apenas as respostas, totalizando 220 gravações, com 110 declarações verdadeiras e 110 falsas. 180 dessas foram utilizadas para treinamento das redes neurais, e as 40 restantes foram utilizadas nos testes de classificação.

Para a extração das características da voz nos áudios, utilizou-se uma biblioteca em Python chamada Librosa. A partir dessa, pode-se definir a quantidade de MFCCs extraídos por arquivo. Inicialmente esse valor foi setado como 13, depois como 20 e finalmente como 40, obtendo-se três tipos de conjunto de dados. Cada gravação processada pela biblioteca gerou uma matriz de dados, com tamanho relativo ao tamanho do arquivo e a quantidade de MFCCs extraídos. Após essa etapa, um método de normalização chamado *padding* foi utilizado, com isso, as sequências de MFCC obtidas passaram a ter o mesmo tamanho.

A partir disso, foi possível então realizar o treinamento das redes neurais. A arquitetura de rede neural utilizada foi a LSTM, com a função de perda *cross-entropy*, acessada pelo TensorFlow, e a função de ativação *softmax*, a qual é eficiente para problemas envolvendo classificação. Os testes foram feitos considerando a variação de alguns parâmetros para cada experimento. Esses parâmetros foram os seguintes: taxa de aprendizado, número de camadas ocultas, número de células por camadas oculta, tamanho do lote, número de iterações, e número de MFCCs por sequência.

Os resultados foram expressos na Tabela 3. Somente os conjuntos mais eficientes de todos os conjuntos testados foram apresentados. Pode-se observar que o Modelo 1 foi o mais eficiente, com uma acurácia de 72,5%. Das 40 entradas, o modelo classificou 29 corretamente e 11 incorretamente, com 6 falsos positivos e 5 falsos negativos. Outro ponto relevante é a variação entre os parâmetros de cada modelo. Fatores como número de camadas ocultas (*layers*), número de células por camada oculta (*cells*) e tamanho do lote (*b atch*) mostraram pouca diferença entre os experimentos, uma vez que valores maiores ou menores não apresentaram resultados com significância. O valor de número de iterações se manteve no intervalo entre 100 e 200 pois com o aumento desse, a rede neural desenvolveu um comportamento de *overfitting*. A taxa de aprendizado foi a medida com maior alteração, variando entre 0.01 e 0.001. Em relação ao número de MFCCs, notou-se que os modelos com 13 e 20 MFCCs foram mais eficientes que os com 40, visto que a maior acurácia de um modelo com 40 MFCCs foi 55%.

Tabela 3 – Resultados do trabalho de Marcolla, Santiago e Dazzi (2020) comparados

Nº Model	Layers	Cells	Nº MFCC	Batch	Iterations	Learn. Rate	Accuracy (%)
1	3	300	13	64	150	0.01	72.5
2	4	300	20	64	100	0.003	70.0
3	3	300	13	64	80	0.003	67.5
4	4	200	20	64	100	0.003	65.0
5	3	300	40	64	180	0.001	55.0

Fonte: Marcolla, Santiago e Dazzi (2020)

Com isso, concluiu-se que há a possibilidade de detecção de mentiras pela voz a partir de redes neurais, e que para a obtenção de resultados mais relevantes nessa área, pode-se considerar a utilização de bases de dados mais volumosos, assim como a consideração de diferentes idiomas e sotaques.

3.5 Comparativo

As diferenças mais relevantes entre os trabalhos analisados foram: os parâmetros extraídos da voz, a base de dados utilizada e o classificador implementado. A Tabela 4 a seguir explicita os conceitos abordados por cada artigo citado.

Percebe-se que as redes neurais são uma importante classe de classificadores para esse tipo de problema, visto que as principais soluções da literatura utilizam tais estruturas. Outro ponto importante é o extenso uso de LLDs, dos trabalhos relacionados analisados, todos concluíram que esses parâmetros são um importante avanço para a detecção de estresse na voz. Em relação aos dados utilizados, as publicações citadas conseguiram resultados relevantes com ambos dados de base públicas e dados próprios.

Tabela 4 – Resultados dos trabalhos citados

Trabalho	**Parâmetros**	**Classificadores Analisados**	**Base de Dados**	**Acurácia máxima (%)**
Tomba et al. (2018)	EM, IM e MFCCs	SVM e RNA	EmoDB, KeioESD e RAV	83,33%
Han, Byun e Kang (2018)	MFCCs	LSTM-Softmax e LSTM-SVM	Própria	65,2%
Avila et al. (2019)	CEM e openSMILE	CNN, DNN e SVM	SUSAS	72%
Marcolla, Santiago e Dazzi (2020)	MFCCs	LSTM	Própria	72,5%

Fonte: o Autor

O presente trabalho utilizará de redes neurais LSTM para classificação, a partir de uma base de dados pública e de parâmetros extraídos a partir do *software* OpenSMILE.

4 Desenvolvimento

Nesse capítulo são descritas as etapas de desenvolvimento desse trabalho, das extrações das características até os experimentos realizados com as redes neurais implementadas.

4.1 Extração de características da voz

Para a extração de características da voz, usou-se o software de código aberto OpenSMILE (Open-Source Media Interpretation by Large feature-space Extraction), o qual é utilizado amplamente na área de análise de sinais e machine learning. Através de métodos de processamento de sinais, o OpenSMILE é capaz de realizar diversos cálculos e transformações sobre arquivos de áudio, de forma a extrair uma grande variedade de LLDs.

A captura de características no OpenSMILE funciona a partir de arquivos de configurações, os quais explicitam quais parâmetros serão extraídos, qual o tamanho de cada *frame* de áudio processado, e o formato de saída dessas informações, por exemplo. Dentre os arquivos disponibilizados pelo próprio OpenSMILE, há arquivos de configuração específicos para reconhecimento de emoção na voz. Nesse trabalho, foram utilizados dois arquivos de configuração, os quais focam em LLDs relevantes para a detecção de emoção na voz, conforme apresentado nos artigo de Schuller, Steidl e Batliner (2009). Os LLDs extraídos foram:

- MFCCs: Coeficientes cepstrais de frequência de Mel (do inglês *Mel-frequency cepstral coefficients*);
- logMelFreqBand: Potência logarítmica das bandas de frequência de Mel (do inglês *Logarithmic power of Mel-frequency bands*);
- lspFreq: 8 Frequências espectrais computadas a partir de 8 coeficientes de predição (do inglês *8 line spectral pair frequencies computed from 8 LPC coefficients*);
- F0finEnv: Envelope do contorno suavizado da frequência fundamental (do inglês *Envelope of the smoothed fundamental frequency contour*);
- voicingFinalUnclipped: A probabilidade de vocalização do candidato final da frequência fundamental (do inglês *The voicing probability of the final fundamental frequency candidate*);

Com essa estrutura definida, o OpenSMILE faz a extração das características após a aplicação de um *moving filter* à onda do áudio, e dependendo da configuração definida no

arquivo, aplica funções estatísticas aos valores encontrados nos *frames* do áudio, ou expõe os valores referentes a cada *frame* separadamente. Caso utilize-se a primeira opção, a saída do processo de extração para cada áudio são 384 valores, independente da duração da mídia. Esses valores são referentes a funções estatísticas, como máximo, mínimo e média aritmética aplicadas aos LLDs extraídos nos *frames* do áudio analisado. Com a segunda opção, a saída consiste nos valores dos LLDs referentes à cada *frame* do áudio, portanto, com tamanho proporcional ao tamanho do arquivo de mídia. Nesse trabalho, utilizou-se a solução *frame* por *frame* da segunda opção, uma vez que redes neurais recorrentes apresentam bons resultados com dados sequenciais.

A utilização do openSMILE se dá por uma interface de linha de comando, a qual permite a especificação do arquivo de configuração a ser utilizado, assim como um arquivo de entrada e um de saída. Para o processamento de todos os áudios do *dataset* desse trabalho, foi desenvolvido um *script* que executa o comandos referente a cada arquivo automaticamente. Com isso, após a execução desse *script*, obtem-se um arquivo .csv para cada áudio, com os valores dos LLDs por *frame* observado.

4.2 Pré-processamento das características extraídas

Com o objetivo de organizar os dados extraídos em um formato compatível com redes neurais recorrentes LSTM, foram aplicados processos de ordenamento e normalização. Primeiramente, os dados contidos nos arquivos .csv são lidos de acordo com as características escolhidas para teste, em seguida, essas características são carregadas em um uma matriz tridimensional com dimensões nº de áudios x nº de frames x nº de características.

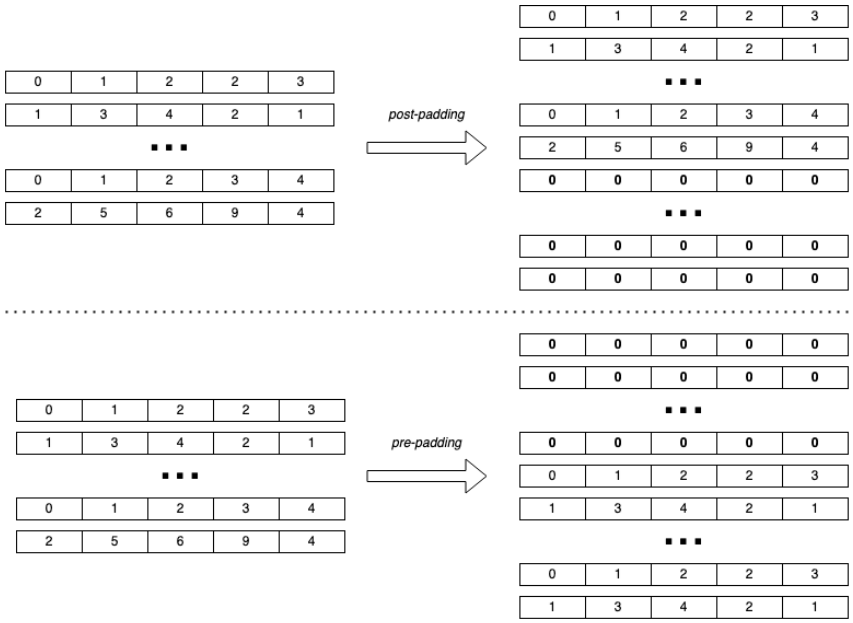
Devido ao caráter temporal dos dados extraídos, áudios com diferentes durações geram conjuntos de dados de diferentes tamanhos. Um áudio de 4 segundos, por exemplo, terá mais *frames* que um áudio de 2 segundos, e portanto, um conjunto maior de dados. Para contornar essa irregularidade e fornecer entradas com dimensões fixas para a rede neural, utilizou-se uma técnica chamada *padding*, a qual consiste na aplicação de valores constantes para a normalização de dados. No caso desse trabalho, todos os conjuntos de características foram preenchidos com valor 0 até atingir o tamanho do conjunto do maior áudio do *dataset*. Dessa maneira, todos os conjuntos de características apresentam o mesmo tamanho no fim da etapa de extração.

De acordo com o artigo de Dwarampudi e Reddy (2019), a forma como o *padding* é aplicado no *dataset* pode implicar em diferentes comportamentos em uma LSTM. O trabalho citado fez testes com dois tipos de *padding*, denominados *pre-padding* e *post-padding*. O primeiro consiste na aplicação de valores constantes no começo da sequência a ser normalizada, já o segundo implica na inserção dos valores no fim da sequência. Os testes foram realizados em uma rede neural LSTM e em uma rede neural convolucional, e

a partir dos resultados obtidos, observou-se que o tipo de *padding* aplicado foi relevante na redes LSTM, enquanto na convolucional não houve diferenças significantes. Para o problema abordado no artigo, o uso de *pre-padding* acarretou em resultados melhores nas redes LSTM, dessa forma, concluiu-se que esse comportamento está relacionado com o funcionamento das redes neurais recorrentes. Com isso, para evitar a influência do *padding* no comportamento das redes neurais desenvolvidas, uso-se um método chamado *masking*, o qual permite informar às camadas de processamento que certos passos de tempo em uma entrada se tratam de *padding*, e portanto devem ser ignorados.

Na Figura 1 estão representações de dados do presente trabalho após a aplicação de *pre-padding*, *post-padding* e *masking*.

Figura 1 – Exemplos de *post-padding* e *pre-padding*



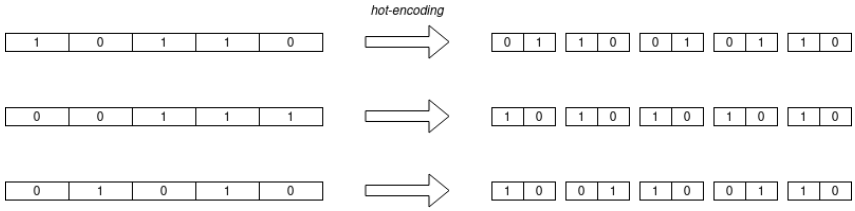
Fonte: O Autor

Outra etapa importante no pré-processamento de dados é o tratamento das informações chamadas *labels*, as quais representam o tipo de cada áudio do *dataset*. Originalmente, esse conjunto de dados consiste em uma sequência de valores 0 ou 1 ordenados de acordo com o *dataset*. O valor 0 representa áudios sem estresse na voz, e o valor 1 representa áudios com a presença de estresse. Após a definição da estrutura de *labels*, essa sequência foi submetida a um processo chamado *hot-one encoding*, o qual consiste em transformar o valores inteiros de *labels* em uma combinação de bits. Esse é um método amplamente utilizado em classificações multi classe, e apesar do presente trabalho se tratar de uma classificação binária, a aplicação do *hot-one encoding* apresentou vantagens no processo de treinamento da rede neural. Dessa forma, as *labels hot-encoded* passaram a consistir em

uma sequência de dois bits, com o primeiro bit indicando se o áudio não contém estresse, e o segundo representando áudios com presença de estresse.

A Figura 2 representa o processo de *hot-encoding* utilizado no conjunto de *labels* desse trabalho.

Figura 2 – Exemplos de *hot-encoding*



Fonte: O Autor

4.3 Datasets

As etapas de treinamento, validação e teste das redes neurais desenvolvidas foram realizadas a partir de dados adquiridos de base de áudios públicas especializadas em emoções. As bases utilizadas foram a SAVEE (Surrey Audio-Visual Expressed Emotion) e a RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song), as quais apresentam dados sobre diferentes emoções. Durante a busca por dados, não foram encontradas bases públicas que lidassem com estresse especificamente. Dessa forma, fez-se necessário o aprofundamento em conceitos de psicologia e neurociência, com o objetivo de correlacionar as emoções encontradas nas bases de dados com o estresse.

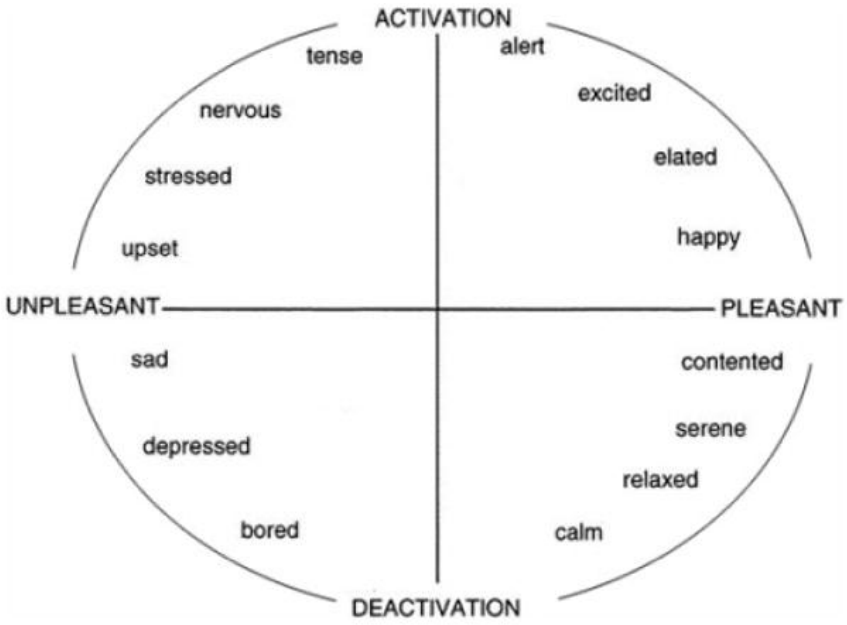
A primeira base encontrada foi a SAVEE, a qual contém dados em formato de áudio e texto, com somente a parte de áudio sendo relevante para esse trabalho. Nessa, os dados de áudio foram gravados por quatro falantes nativos da língua inglesa do sexo masculino, com idades entre 27 e 31 anos. As emoções de cada gravação foram descritas em seis categorias distintas: raiva, nojo, medo, felicidade, tristeza e surpresa. Além disso, foi adicionada uma categoria denominada "neutra", referente a áudios sem nenhuma emoção específica. Essa divisão de emoções é apoiada pela literatura na área de psicologia, portanto estudos de reconhecimento automático de emoções tendem a se concentrar em reconhecê-las dessa forma.

A segunda base utilizada, RAVDESS, contém dados em áudio e vídeo, com gravações de 24 atores profissionais (12 mulheres e 12 homens) proferindo frases e músicas com emoções na língua inglesa. Há também uma versão com apenas o áudio de cada gravação. As gravações de fala incluem as emoções calma, alegria, tristeza, raiva, medo, nojo e

surpresa, e assim como na SAVEE, foi incluído um subgrupo neutro. Além disso, cada expressão foi produzida em dois níveis de intensidade: normal e forte.

Durante o desenvolvimento dos estudos na área das emoções e psicologia humana, diversos modelos de representação foram propostos. Dentre esses, um modelo relevante atualmente é a abordagem dimensional denominada de "modelo circumplexo do afeto", evidenciado no artigo de Russell (1980). Nesse, propõe-se que todos os estados emocionais humanos surgem de dois sistemas neurofisiológicos fundamentais, um relacionado à valência (prazer ou desprazer) e o outro à excitação, ou estado de alerta. Com isso, cada emoção pode ser entendida como uma combinação linear dessas duas dimensões, ou como vários graus de valência e excitação. Essa organização está representada na Figura 3.

Figura 3 – Modelo circumplexo do afeto



Fonte: Posner, Russell e Peterson (2005)

A partir desses conceitos, foi possível montar os *datasets* do presente trabalho utilizando as emoções contidas nas bases de áudio. Para o conjunto de dados sem estresse, utilizou-se os áudios anotados como neutros, e no caso da base RAVDESS, as gravações com a emoção calma também foram selecionadas para esse caso, uma vez que essa emoção mostra valores opostos ao estresse em relação à valência e excitação. Já para o conjunto de dados com estresse, foram selecionados áudios das emoções raiva e medo, as quais, assim como o estresse, são formados por valores positivos de excitação e negativos de valência.

Na base de dados SAVEE, há um total 480 arquivos de áudio, divididos igualmente entre os 4 atores. Para cada falante, há 15 gravações diferentes para cada emoção e 30 áudios da categoria neutra, totalizando 120. Durante o processo de formação do *dataset*,

utilizou-se os 15 áudios disponíveis da emoção raiva, os 15 da emoção ódio e os 30 das falas neutras de cada ator. Como resultado, obteu-se um *dataset* com 240 arquivos de áudio, igualmente dividido entre as categorias sem e com presença de estresse.

Através da base de dados RAVDESS, outro *dataset* foi construído de maneira semelhante. Nessa base há um total de 1440 arquivos, separados igualmente entre 24 atores, 12 do sexo masculino e 12 do sexo feminino. Para cada falante, a base disponibiliza 4 áudios neutros e 8 de cada emoção abordada. Dessa forma, foi possível a obtenção de 12 áudios da categoria sem estresse para cada ator, totalizando 288 áudios. Para a categoria com a presença de estresse, selecionou-se 12 áudios das emoções raiva e medo, divididos igualmente entre as duas emoções e os níveis de intensidade, gerando um total de 288 áudios para essa categoria, e 576 para todo o *dataset*. As características de cada *dataset* estão expostas na Tabela 5.

Tabela 5 – Comparação entre *datasets*

Dataset	**Idioma**	**Nº de amostras**	**Nº de atores**	**Emoções**	**Tipo de amostra**	**Base de dados**
Dataset 1	Inglês	240	4	Neutro, raiva e medo	Expressões	SAVEE
Dataset 2	Inglês	576	24	Neutro, calma, raiva e medo	Expressões	RAVDESS

Fonte: o Autor

Durante a etapa de treinamento e testes, os *datasets* foram divididos em subconjuntos de treinamento, validação e teste. O primeiro subconjunto se trata do utilizado para o aprendizado da rede neural, com o ajuste de pesos das unidades de processamento. Já o segundo é usado para a análise da performance da rede durante o treinamento, possibilitando melhor refinamento de hiperparâmetros. Por fim, o subconjunto de teste possibilita avaliar o desempenho do classificador totalmente treinado. Para o *datasets* utilizados, a separação desses subconjuntos se deu da seguinte forma: do total de áudios, 25% foram utilizados para testes, e dos 75% restantes, 80% formaram o conjunto de treinamento e 20% o conjunto de validação.

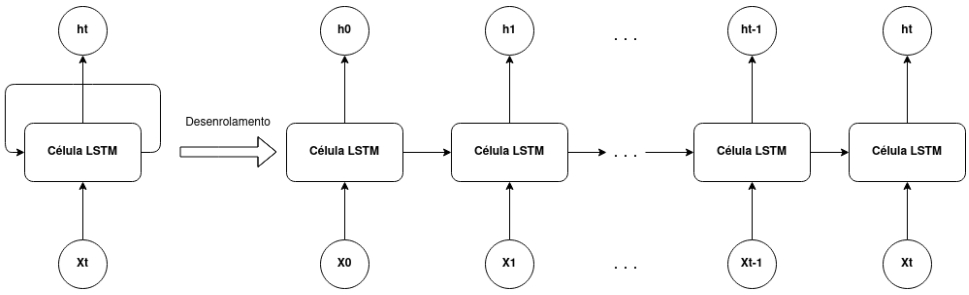
Além dos dados dos *datasets* citados, foram utilizados os dados coletados no trabalho de Marcolla, Santiago e Dazzi (2020), os quais foram obtidos através de gravações de voluntários respondendo perguntas com respostas falsas e verdadeiras. Levando em consideração a relação entre o estresse e o ato de contar uma mentira apontada no trabalho citado, utilizou-se desses dados para comparações com os desempenhos obtidos pelos *datasets* das bases públicas.

4.4 Redes Neurais LSTM

As classificações realizadas nesse trabalho foram realizadas por redes neurais recorrentes do tipo LSTM. Para isso, utilizou-se a API de *deep learning* Keras, a qual disponibiliza uma interface produtiva e completa para soluções envolvendo *machine learning*, desenvolvida como uma camada acima da plataforma *open-source* TensorFlow. Através dessa API, foi possível a definição de diferentes topologias de redes neurais para testes, assim como a customização de hiperparâmetros.

As redes neurais LSTM desenvolvidas são compostas por uma camada de entrada, camadas LSTM e uma camada de saída. A primeira camada é a responsável por alimentar os dados de entrada na rede neural, os quais tem dimensões definidas pelo tamanho das *batches*, da quantidade de sequências de tempo e número de características (LLDs) analisadas (tamanho das *batches* x nº de sequências de tempo x nº de características). As camadas intermediárias são compostas por células LSTM, com a quantidade de células em uma camada dependendo do número de sequências de tempo da entrada. Isso pode ser interpretado como um "desenrolamento" de uma célula LSTM ao decorrer do processamento das sequências de tempo, como representado na Figura 4. Por fim, a última camada se trata de uma camada de dois neurônios, a qual, a partir de uma função de ativação, fornece um resultado para a classificação.

Figura 4 – Desenrolamento de uma célula LSTM



Fonte: o Autor

O objetivo das redes LSTM desenvolvidas é aprender sobre as entradas fornecidas, e então classificá-las entre duas categorias: presença de estresse e não presença de estresse. Para isso, a rede neural precisa consumir os dados de entrada iterativamente, e a cada iteração, avaliar e atualizar seus pesos internos. A avaliação de tais pesos acontece a partir de uma função *loss*, a qual calcula o erro do modelo nas iterações de treinamento, fornecendo um valor que descreve a eficácia da configuração de pesos analisada. A função utilizada nas redes desse trabalho foi a *binary cross-entropy* (entropia cruzada binária), que utiliza de logaritmos e distribuições probabilísticas para o valor de *loss* em classificações

binárias. A partir disso, o otimizador da rede neural fornece um método de minimizar o valor encontrado pela função *loss*.

As redes neurais desenvolvidas utilizaram do otimizador Adam, o qual se trata de um método de descida de gradiente estocástico para a atualização dos pesos das redes de forma iterativa, a partir dos dados de entrada. Esse algoritmo foi exposto no artigo de Kingma e Ba (2014) e desde então se tornou popular na área de aprendizado de máquina, principalmente pelo fato de atingir bons resultados rapidamente. O otimizador Adam utiliza de uma taxa de aprendizado para controlar o quanto os pesos da rede são atualizados a cada iteração do treinamento, influenciando na velocidade com que o modelo aprende sobre as entradas a serem classificadas. Dessa forma, a taxa de aprendizado se faz um importante hiperparâmetro a ser configurado durante o refinamento de uma rede neural.

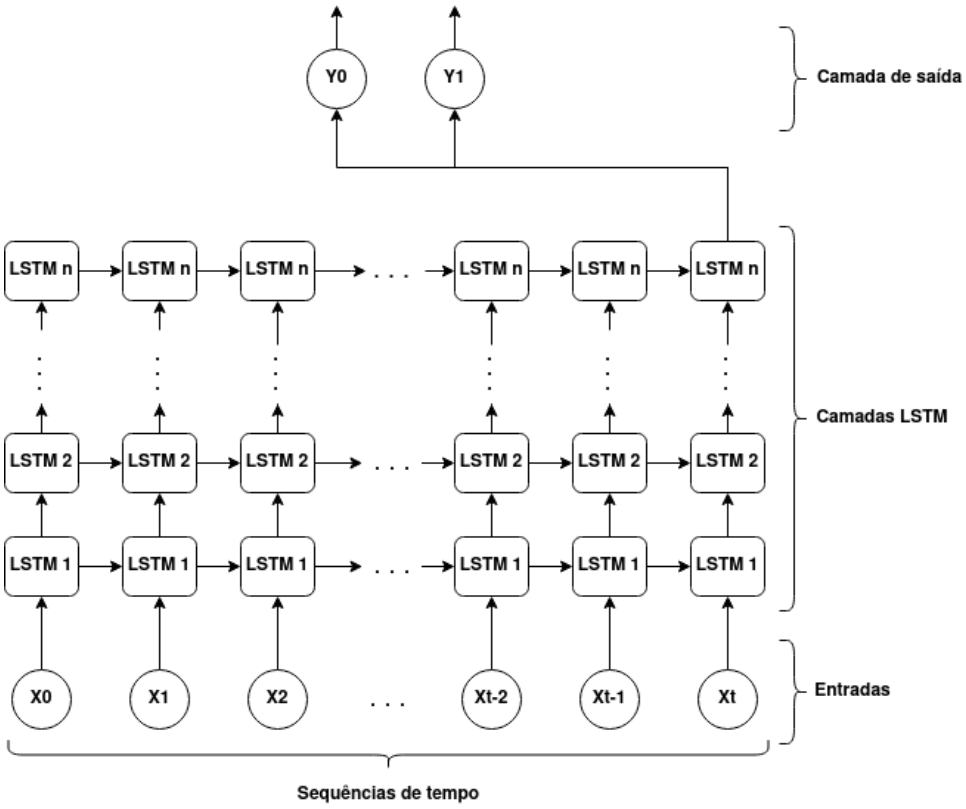
Outro fator relevante da arquitetura de uma rede neural é a função de ativação, a qual é responsável por definir a saída de um neurônio. Nesse trabalho, a função de ativação utilizada nos neurônios da camada de saída foi a *softmax*. Essa função consiste numa generalização da função *sigmoid*, com ambas produzindo valores entre 0 e 1 que representam probabilidades. Uma diferença importante entre essas funções é o fato das probabilidades geradas pela *sigmoid* serem independentes, ou seja, não necessariamente a soma dos valores gerados é 1, enquanto na *softmax* as saídas são relacionadas, com sua soma sempre sendo 1. É comum a função *sigmoid* ser utilizada em classificações binárias, enquanto a *softmax* é amplamente usada em problemas multiclasse. Nesse trabalho, testou-se as duas opções de saída:

- 1 neurônio na camada de saída com função de ativação *sigmoid*;
- 2 neurônios na camada de saída com função de ativação *softmax* e *hot-encoding*;

A solução utilizando *hot-encoding* e dois neurônios na camada de saída com função *softmax* apresentou resultados levemente melhores no geral, portanto, foi a escolhida na definições das redes neurais LSTM a serem refinadas. Dessa forma, a saída da última camada consiste nas probabilidades calculadas de um áudio conter ou não estresse. A Figura 5 representa a topologia da rede neural LSTM desenvolvida nesse trabalho.

Durante o processo de refinamento das redes desenvolvidas, os hiperparâmetros manipulados foram: número de unidades de processamento das células LSTM, taxa de aprendizado, *dropout*, tamanho de *batch* e épocas de treinamento. Além disso, foram testadas diversos tipos de entrada a partir das características (LLDs) extraídas dos áudios. O processo de treinamento e refinamento das redes neurais testadas foi documentado e discutido no capítulo seguinte.

Figura 5 – Exemplo de LSTM



Fonte: o Autor

5 Experimentos

Os experimentos realizados nesse trabalho consistiram nas seguintes etapas:

1. Testes de entradas e configurações de LSTM com o *dataset* da base RAVDESS;
2. Testes de entradas e configurações de LSTM com o *dataset* da base SAVEE;
3. Teste da LSTM com maior acurácia da primeira etapa com dados da base da segunda etapa;
4. Teste da LSTM com maior acurácia da segunda etapa com dados da base da primeira etapa;
5. Teste de ambas as LSTMs com maior acurácia com os dados coletados no trabalho de Marcolla, Santiago e Dazzi (2020);

5.1 Entradas e configurações de LSTM

As duas primeiras etapas foram executadas a partir de uma série de testes envolvendo configurações de redes neurais LSTM e os dados contidos nos *datasets*. Através do conjunto de características extraídas de cada áudio e da customização dos hiperparâmetros das redes, foi possível a análise de combinações de diferentes entradas com configurações de LSTMs. As características da voz analisadas em cada *dataset* foram as seguintes:

- MFCCs: Coeficientes cepstrais de frequência de Mel (do inglês *Mel-frequency cepstral coefficients*);
- logMelFreqBand: Potência logarítmica das bandas de frequência de Mel (do inglês *Logarithmic power of Mel-frequency bands*);
- lspFreq: 8 Frequências espectrais computadas a partir de 8 coeficientes de predição (do inglês *8 line spectral pair frequencies computed from 8 LPC coefficients*);
- F0finEnv: Envelope do contorno suavizado da frequência fundamental (do inglês *Envelope of the smoothed fundamental frequency contour*);
- voicingFinalUnclipped: A probabilidade de vocalização do candidato final da frequência fundamental (do inglês *The voicing probability of the final fundamental frequency candidate*);

Os hiperparâmetros das LSTMs considerados foram:

- Número de camadas LSTM;
- Número de unidades de processamento das células LSTM;
- Taxa de aprendizado;
- *Dropout*;
- Tamanho dos *batches*;
- Número de épocas de treinamento;

O processo de desenvolvimento das redes neurais LSTM foi o seguinte: para cada característica extraída, montou-se uma entrada para uma rede neural LSTM base, e então determinou-se a melhor topologia da rede a ser utilizada, a partir da variação dos hiperparâmetros. Para a análise dos efeitos da variação de cada hiperparâmetro, o estado inicial das redes foi fixado, e com as configurações mais promissoras foram realizados testes com maior volume. A qualidade das redes LSTM encontradas foi definida a partir dos valores de acurácia e generalização, observados nas iterações da etapa de treinamento e validação e nos testes. Nas metodologias usadas no processo de refinamento das redes, foram utilizados conceitos como os apresentados no artigo de Reimers e Gurevych (2017), no qual são avaliados os impactos da variação de hiperparâmetros em redes LSTM. No entanto, de forma geral as escolhas foram feitas de forma empírica.

5.2 Refinamento

Durante o período de treinamento, validação e teste, analisou-se o comportamento das redes neurais construídas, e dependendo dos resultados de acurácia e generalização, refinamentos e modificações foram feitos iterativamente.

Na etapa de treinamento, o número de épocas define a quantidade de vezes que a rede neural interage com todo o conjunto de dados de treinamento. Esse número normalmente é maior que 1, ou seja, a rede itera sobre todo o conjunto de treinamento mais de uma vez. Isso acontece devido à natureza iterativa de otimizadores de descida de gradiente. A quantidade de iterações por época é definido pela divisão da quantidade de dados de treinamento pelo número de dados por *batch*:

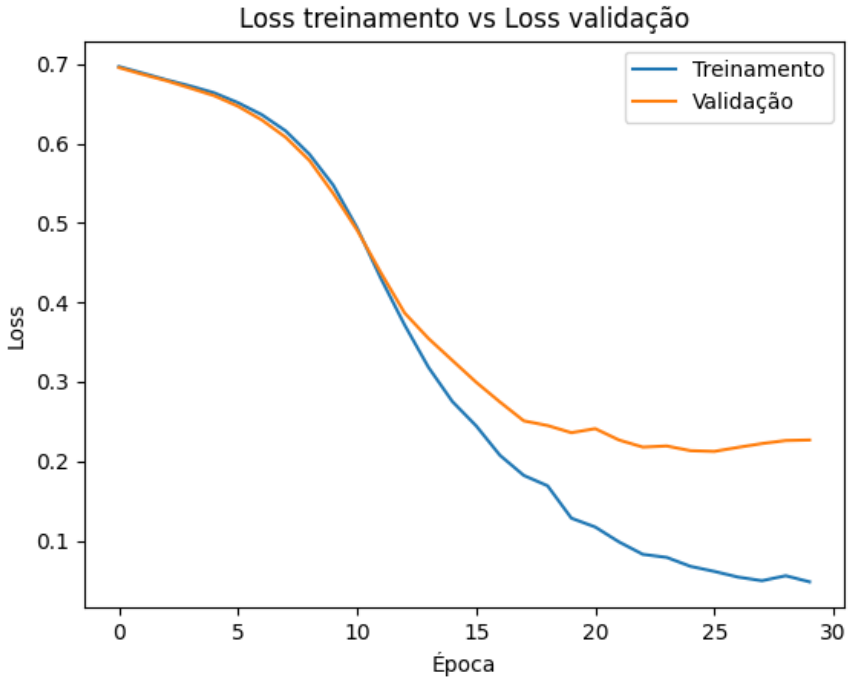
$$n^o \text{ de iterações por época} = \text{total de dados de treinamento} \;/\; \text{dados de treinamento por batch}$$

A cada iteração, ou seja, a cada *batch* processado, a rede neural ajusta seus pesos internos a partir do erro calculado pela função *loss*, e após o processamento de todos os *batches*, uma época é concluída, obtendo-se um valor de acurácia e *loss* relativo ao estado

da rede nessa época. Nesse momento, o conjunto de validação é passado como entrada na rede neural, de forma a se obter um valor de acurácia e *loss* relativos ao conjunto de validação, o qual contém dados diferentes do conjunto de treinamento. Isso é útil para a análise do desempenho da rede neural pois indica como o estado da rede se comporta com dados externos, isto é, diferentes dos dados de treinamento. A capacidade de uma rede neural classificar corretamente dados diferentes dos de treinamento é chamada de generalização.

Um fenômeno relevante no treinamento de redes neurais é o *overfitting*, o qual ocorre quando a generalização da rede neural começa a degenerar, ou seja, a rede se especializa no conjunto de dados de treinamento e perde a capacidade de generalização. Esse comportamento pode ser observado durante o treinamento a partir do conjunto de validação, uma vez que, quando a rede começa a desenvolver *overfit*, o valor de *loss* do treinamento cai continuamente enquanto o *loss* da validação para de diminuir ou aumenta em um algum ponto do treinamento. O gráfico na Figura 6 abaixo apresenta um treinamento executado nesse trabalho que apresentou *overfitting*.

Figura 6 – Gráfico de um treinamento com *overfitting*



Fonte: o Autor

Existem técnicas para diminuir o *overfitting* no treinamento. Uma delas é a definição de um critério de parada alternativo para o treinamento, isto é, considerando o valor de *loss* a cada época, pode-se interromper o treinamento da rede antes do *overfitting*

surgir, quando a rede apresentar uma boa capacidade de generalização e a taxa de *loss* for suficientemente pequena.

Outro método existente é o *dropout*, o qual consiste em omitir unidades de processamento da rede neural durante o processo de treinamento de forma aleatória. Isso evita que as camadas da rede desenvolvam padrões que estimulam o *overfitting*, tornando a rede mais robusta. Esse artifício foi apresentado pela primeira vez no artigo de Hinton et al. (2012), e desde então é amplamente considerado na implementação de redes neurais. No entanto, no caso de redes neurais recorrentes, a utilização do *dropout* não é trivial e depende de alguns conceitos adicionais. No trabalho de Gal (2015), foi proposta uma maneira eficiente de se aplicar *dropout* em redes neurais recorrentes, a qual consiste em repetir *dropouts* em cada *timestep* nas entradas, saídas e conexões recorrentes das células. O *dropout* disponibilizado na API do Keras considera esses conceitos, e foi testado nas redes desenvolvidas nesse trabalho.

5.2.1 Redes treinadas com RAVDESS

Abaixo estão os resultados encontrados a partir das redes neurais LSTM treinadas com dados do *dataset* da base RAVDESS. Na Tabela 6 estão presentes a melhores configuração de LSTM encontrada para cada parâmetro testado, considerando a acurácia obtida nos testes.

Tabela 6 – Melhores resultados por parâmetro das redes treinadas no RAVDESS por ordem de acurácia

Parâmetros	Camadas	Unidades	Dropout	Taxa de aprendizado	Épocas	Batch	Acurácia(%)
logMelFreqBand	3	200	0.3	0.00008	60	64	**97,91**
MFCCs	3	100	0.3	0.00005	150	32	**96,527**
lspFreq	3	200	0.0	0.00005	150	64	**89,583**
F0finEnv	2	64	0.0	0.0001	100	32	**84,722**
voicingFinalUnclipped	3	64	0.0	0.0006	100	32	**65,972**

Fonte: o Autor

Dentre os parâmetros testados, os que atingiram maior acurácia na etapa de teste foram as potências logarítmicas das bandas de frequência de Mel, com 97,91%. Esse valor foi atingido a partir de uma taxa de aprendizado 0.00008, 3 camadas LSTM e 200 unidades de processamento nas células LSTM. O treinamento dessa rede apresentou *overfitting* ao decorrer das épocas, dessa forma, a acurácia foi melhorada a partir da interrupção do treinamento antes do desenvolvimento de *overfitting*. Além disso, a utilização de *dropout* foi benéfica nessa rede, e ajudou no desempenho da taxa de *loss* do conjunto de validação.

Os MFCCs também se mostraram uma característica eficiente, pois demonstraram uma eficácia de 96,527% na etapa de testes. Esse modelo atingiu a maior acurácia com

150 épocas de treinamento, e o uso de *dropout* também foi importante para estabilizar a taxa de *loss* de conjunto de validação durante o treinamento.

As características com terceira maior acurácia obtida foram as frequências espectrais, as quais, assim como a rede LSTM anterior, atingiram maior acurácia após 150 épocas. Essa rede não apresentou resultados positivos com o uso de *dropout*, e mostrou maior estabilidade com um aumento de unidades de processamento, assim como a rede das potências logarítmicas.

O envelope do contorno suavizado da frequência fundamental apresentou uma acurácia de 84,722%, com um modelo diferente em alguns aspectos dos anteriores. A rede dessa característica apresentou melhores resultados com menos unidades por célula, e uma taxa de aprendizado maior. A maior acurácia para esse caso foi obtida com 100 épocas de treinamento.

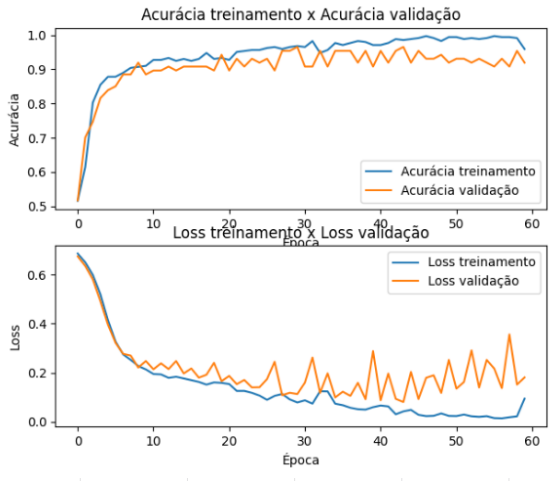
Das características analisadas, a com pior desempenho foi a probabilidade de vocalização do candidato final da frequência fundamental. Durante os testes dessa entrada, houve dificuldade em implementar uma rede que apresentasse convergência. O melhor modelo documentado obteve uma acurácia de 69,298%, no entanto, o seu treinamento não apresentou um bom comportamento, visto que os valores de acurácia e *loss* do conjunto de validação não melhoraram com o andamento das épocas.

Apesar das características testadas terem apresentados acurácias com valores parecidos, é importante a consideração da qualidade das acurácias obtidas, isto é, a capacidade da rede neural conseguir classificar dados externos aos dados de treinamento. Para isso, analisou-se o comportamento da rede em relação ao conjunto de validação durante o treinamento. No caso da rede das potências logarítmicas das bandas de frequência de Mel, o modelo com maior acurácia apresentou o comportamento apresentado na Figura 7.

Pode-se observar que a acurácia do conjunto de validação aumentou de maneira semelhante ao conjunto de teste, porém apresentou uma maior instabilidade. No caso da taxa de *loss*, um comportamento parecido foi obtido, com os valores de treinamento e acurácia decrescendo juntos, mas com o de validação mostrando instabilidade. Uma forma de lidar com essa instabilidade é diminuir a robustez da rede, para que essa possa convergir mais rapidamente, outra possibilidade seria diminuir a taxa de aprendizado, para que o otimizador encontre um ponto ótimo de maneira mais controlada. Ambas essas soluções foram testadas nesse caso, e apesar de terem demonstrado uma melhora no processo de aprendizado de acordo com os gráficos, não apresentaram uma maior acurácia. Na Figura 8 está representado o comportamento desse modelo com menos unidades de processamento e com uma taxa de aprendizado de 0.00001.

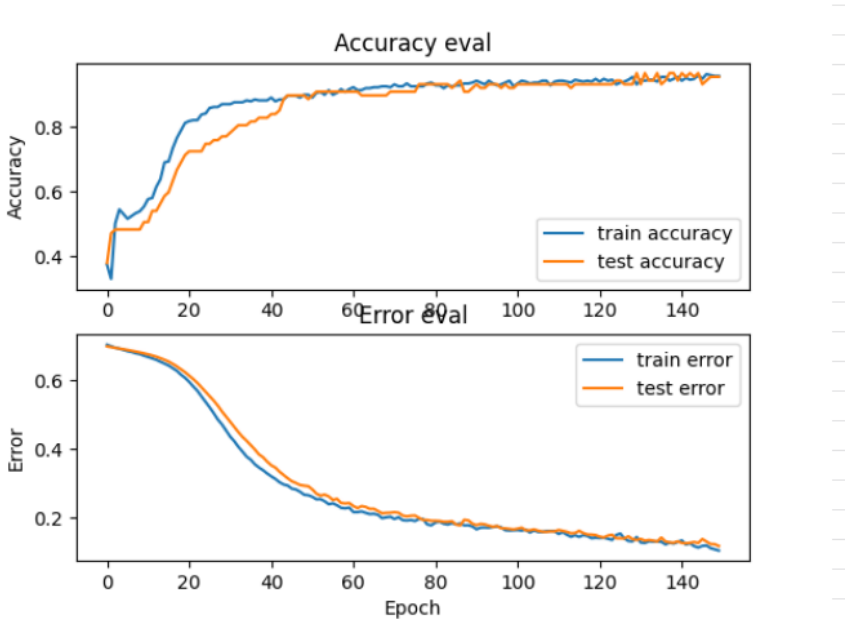
Com os MFCCs como entrada, o comportamento dos modelos desenvolvidos foi diferente. A maior acurácia obtida foi após o treinamento por 150 épocas, com o valor

Figura 7 – Gráficos de treinamento de uma rede treinada com logMelFreqBand da base RAVDESS



Fonte: o Autor

Figura 8 – Gráficos de treinamento de uma rede treinada com logMelFreqBand da base RAVDESS

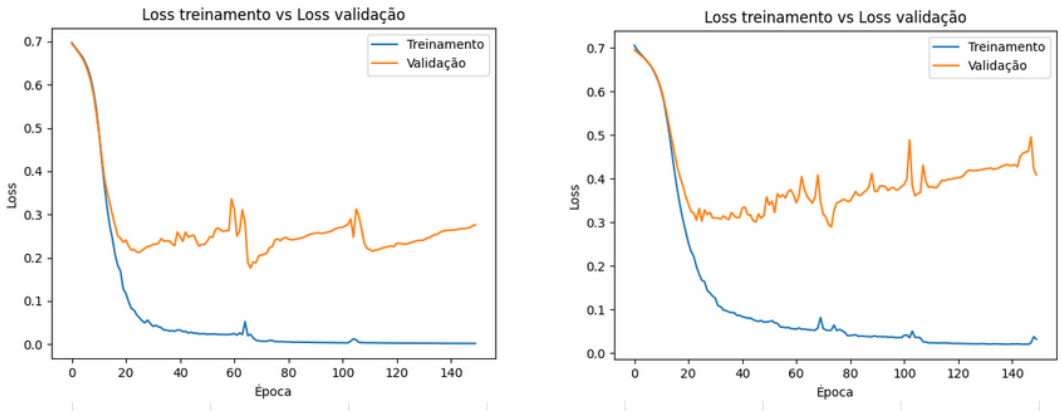


Fonte: o Autor

de acurácia do conjunto de validação eventualmente convergindo. A partir da variação da taxa de *loss*, percebe-se que o modelo sofreu *overfitting*, com essa taxa subindo com o andamento das épocas. A utilização do *dropout* ajudou no controle desse crescimento. A Figura 9 se trata de uma comparação entre um treinamento desse modelo com e sem *dropout*.

No modelo com maior acurácia utilizando as 8 frequências espectrais como entrada, o comportamento obtido durante o treinamento foi bastante instável, e a partir da variação

Figura 9 – Gráficos de loss durante o treinamento da rede treinadas com MFCCs da base RAVDESS com e sem dropout



Fonte: o Autor

dos hiperparâmetros, não foi encontrada uma configuração estável com maior acurácia. Esse comportamento foi o mesmo observado nas redes do envelope da frequência fundamental e da probabilidade de vocalização. Em ambos os casos as redes não convergiram, e apresentaram altas taxas de *loss* no conjunto de validação durante o treinamento.

5.2.2 Redes treinadas com SAVEE

O treinamento de redes neurais LSTM com dados do *dataset* SAVEE estão representados na Tabela 7. Assim como no exemplo anterior, a tabela apresenta a melhores configuração de LSTM encontrada para cada parâmetro testado, considerando a acurácia obtida nos testes.

Tabela 7 – Melhores resultados por parâmetro das redes treinadas no SAVEE por ordem de acurácia

Parâmetros	Camadas	Unidades	Dropout	Taxa de aprendizado	Épocas	Batch	Acurácia(%)
logMelFreqBand	3	100	0.3	0.00005	150	64	**100**
MFCCs	2	100	0.0	0.00005	100	64	**93,333**
F0finEnv	3	100	0.0	0.00001	100	32	**91,666**
lspFreq	1	50	0.0	0.001	150	32	**89,999**
voicingFinalUnclipped	1	64	0.0	0.001	150	32	**78,333**

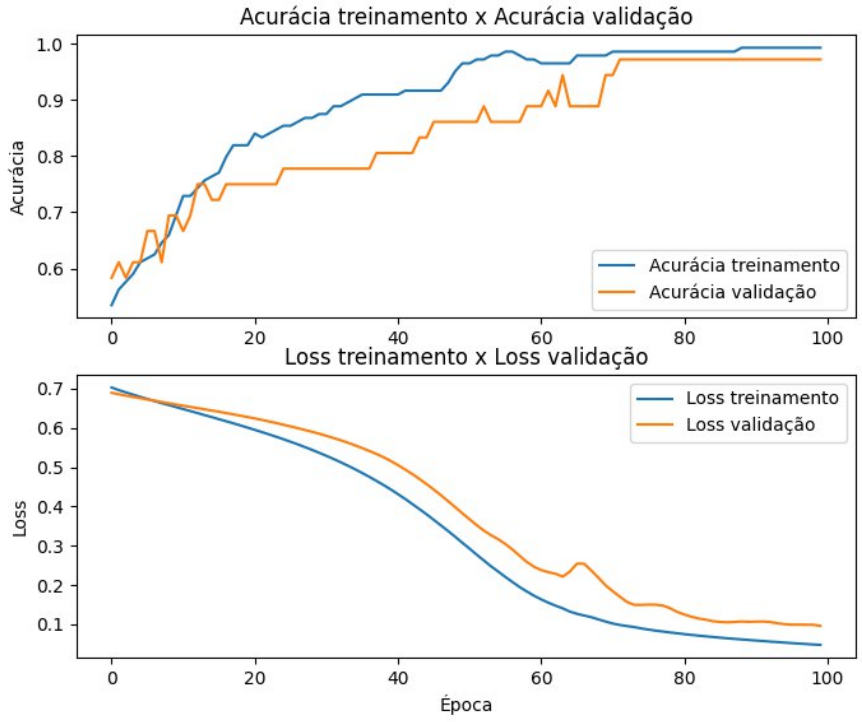
Fonte: o Autor

Nesses experimentos, a rede com maior acurácia foi treinada com as potências logarítmicas, utilizando 3 camadas de células LSTM, com 100 unidades de processamento e uma taxa de aprendizado de 0,00005. Essa rede atingiu acurácia de 100% no conjunto de teste, o que indica um problema no treinamento, visto que é muito incomum um classificador atingir um valor de 100% sem alguma inconsistência no processo de aprendizado. Nesse

caso, uma possível explicação é a falta de diversidade do banco de dados utilizado, visto que o SAVEE contém áudios de apenas 4 atores, todos do sexo masculino. Com isso, o conjunto usado para teste pode ser muito semelhante ao usado no treinamento, de forma que o modelo consiga uma acurácia perfeita até mesmo com dados com os quais não foi treinado. Esse processo de treinamento indica uma rede neural com pouca qualidade em suas saídas.

A segunda maior acurácia foi obtida a partir dos MFCCs. A rede neural obtida contém duas camadas de células LSTM e uma taxa de aprendizado de 0,00005, atingindo 93,333% de acurácia. O gráfico do processo de treinamento dessa rede apresentou um comportamento diferente do observado com o *dataset* RAVDESS, nesse caso a taxa de *loss* do conjunto de validação acompanhou a taxa de *loss* do treinamento. A Figura 10 representa o processo de treinamento para essa rede.

Figura 10 – Gráficos de loss durante o treinamento da rede treinadas com MFCCs da base SAVEE



Fonte: o Autor

O envelope da frequência fundamental foi a característica com a terceira maior acurácia. A rede configurada para esse parâmetro utilizou de uma taxa de aprendizado de valor 0,00001, com 3 camadas de células LSTM com 100 unidades de processamento. O processo de aprendizado dessa rede também apresentou diferenças em relação à rede do

dataset anterior. Assim como no caso dos MFCCs, essa rede apresentou uma taxa de *loss* menor ao decorrer das épocas.

No caso das 8 frequências espectrais, a acurácia obtida foi de 89,999%, a partir de uma rede com 1 camada de células LSTM com 50 unidades de processamento. A taxa de aprendizado utilizada foi a de 0,001. Essa rede apresentou um treinamento instável, com os valores da taxa de *loss* não convergindo para um valor menor. O mesmo comportamento foi obtido com as probabilidades de vocalização, as quais apesar da acurácia obtida, não resultaram em um treinamento efetivo da rede neural.

5.2.3 Comparativo

Os experimentos realizados com os dois *datasets* apresentaram aspectos em comum e diferenças, os quais foram evidenciados nos resultados e análises. Em ambos os *datasets*, a melhor acurácia foi obtida através das potências logarítmicas das bandas de frequência de Mel, com os MFCCs apresentando a segunda melhor precisão. No entanto, observou-se diferenças no comportamento dos treinamentos, as quais estão relacionadas com a natureza e quantidade dos dados utilizados. No caso do *dataset* com dados da base SAVEE, percebeu-se que as redes foram influenciadas pela baixa variação dos dados, gerando saídas de menor qualidade. Com isso, faz-se relevante testes com dados de *datasets* diferentes dos utilizados para a formação dos conjuntos de treino, validação e teste das redes desenvolvidas.

5.3 Testes entre datasets

Com o objetivo de validar a qualidade das redes neurais treinadas com as características analisadas nesse trabalho, foram realizados testes entre as redes LSTM apresentadas anteriormente e os dois *datasets* montados. Com isso, as redes treinadas com o *dataset* da base RAVDESS foram sumbmetidas a um conjunto de teste com dados do *dataset* da base SAVEE, e as redes treinadas com dados da base SAVEE foram testados com áudios da base RAVDESS.

Esse processo se deu a partir do armazenamento das redes com melhor acurácia e da adaptação de suas camadas de entrada, de forma a permitir o processamento das entradas geradas pelo *dataset* usado para o teste, as quais, após o processo de normalização, contêm um tamanho diferente das entradas com qual a rede foi treinada.

5.3.1 Redes treinadas em RAVDESS e testadas com o dataset SAVEE

A partir dos resultados apresentados na Tabela 8, percebe-se que a rede treinada com as potências logarítmicas apresentou a melhor generalização. No geral, os valores de *loss* foram altos, portanto os modelos não convergiram o suficiente para serem totalmente

Tabela 8 – Resultados das redes treinadas em RAVDESS e testadas com o dataset SAVEE por ordem de acurácia

Rede	**Acurácia(%)**	**Loss**
logMelFreqBand	85%	0,6149
MFCCs	63,33%	1,5685
F0finEnv	61,66%	0,8812
voicingFinalUnclipped	53,33%	0,9413
lspFreq	43,33%	1,0243

Fonte: o Autor

confiáveis com dados externos. Isso se dá devido à forma como a rede se comportou durante o treinamento, ou seja, a maneira como seus pesos internos foram ajustados de acordo com as entradas. Como observado anteriormente, a rede treinada com potências logarítmicas obteve o maior nível de convergência, e isso se fez evidente no fato de essa rede ter apresentado a maior acurácia nesse teste.

5.3.2 Redes treinadas em SAVEE e testadas com o dataset RAVDESS

Na Tabela 9 estão os resultados dos testes das redes treinadas com o *dataset* SAVEE com dados do *dataset* RAVDESS. Pode-se observar uma baixa acurácia e grandes valores de *loss* no geral, resultantes da falta de generalização das redes no processo de treinamento. Como já visto, as redes treinadas pelo *dataset* SAVEE apresentaram saídas de menor qualidade, portanto, suas acurácias apresentaram uma queda significativa quando submetidas a um *dataset* diferente.

Tabela 9 – Resultados das redes treinadas em SAVEE e testadas com o dataset RAVDESS por ordem de acurácia

Rede	**Acurácia(%)**	**Loss**
F0finEnv	64,583%	0,9962
MFCCs	60,714%	1,4253
logMelFreqBand	52,083%	2,3486
lspFreq	51,388%	2,3209
voicingFinalUnclipped	45,833%	0,8378

Fonte: o Autor

5.4 Testes com dataset de mentiras

Além dos testes com os *datasets* montados nesse trabalho, foram realizados validações usando o conjunto de dados construído por Marcolla, Santiago e Dazzi (2020). Esse

conjunto se trata de áudios de expressões divididas entre verdades e mentiras, enunciadas por homens na língua portuguesa. Através desse teste, pode-se medir a eficiência das redes treinadas nesse trabalho na classificação do estresse contido no ato de contar uma mentira.

5.4.1 Redes treinadas com RAVDESS testadas com dados do dataset de mentiras

A Tabela 10 apresenta os valores de acurácia e *loss* obtidos pelas redes treinadas com o *dataset* RAVDESS quando submetidas ao dados de Marcolla, Santiago e Dazzi (2020).

Tabela 10 – Resultados das redes treinadas com RAVDESS testadas com os dados de Marcolla, Santiago e Dazzi (2020) por ordem de acurácia

Rede	**Acurácia(%)**	**Loss**
logMelFreqBand	60%	1,37
voicingFinalUnclipped	46%	0,8770
lspFreq	46%	1,026
MFCCs	34%	3,1970
F0finEnv	30%	1,7863

Fonte: o Autor

Os resultados apresentados na Tabela 10 evidenciam que as redes treinadas não foram capazes de classificar com precisão os áudios da base testada. Assim como no teste anterior, a rede treinada com as potências logarítmicas apresentou a melhor acurácia, no entanto, o valor do *loss* registrado nesse teste foi significativamente maior do que o anterior, ou seja, a classificação cometeu erros mais graves. Com isso, evidencia-se que o problema de classificar falas como mentira ou verdade exige mais robustez das redes, de forma que essas sejam capazes de detectar padrões numa fala mentirosa a partir do estresse nessa contida.

5.4.2 Redes treinadas com SAVEE testadas com dados do dataset de mentiras

A Tabela 11 apresenta os resultados dos testes envolvendo as redes treinadas com o *dataset* SAVEE e os dados de Marcolla, Santiago e Dazzi (2020).

Como pode-se observar, as acurácias obtidas foram muito baixas, enquanto os valores de *loss* foram altos. Assim como no teste anterior, as redes foram incapazes de generalizar o conhecimento adquirido.

Tabela 11 – Resultados das redes treinadas com SAVEE testadas com os dados de Marcolla, Santiago e Dazzi (2020) por ordem de acurácia

Rede	**Acurácia(%)**	**Loss**
lspFreq	43,999%	1,4584
voicingFinalUnclipped	41,999%	0,8565
logMelFreqBand	41,999%	1,7695
MFCCs	36,000%	1,7606
F0finEnv	34,000%	1,3167

Fonte: o Autor

5.5 Discussões

A realização dos experimentos documentados permitiu a visualização de fenômenos importantes no aprendizado das redes neurais. A partir da análise dos processos de treinamento, evidenciou-se que dependendo do *dataset* utilizado, os mesmos dados de entrada podem gerar comportamentos diferentes. Além disso, a natureza e quantidade dos dados se fez um fator relevante na interpretação das saídas geradas pelas redes neurais desenvolvidas.

Em ambos os *datasets* testados, as redes apresentaram boas acurácias no geral independente do processo de treinamento, mas quando submetidas a dados de outro *dataset*, resultaram em acurácias mais baixas. No caso dos testes entre os *datasets* desse trabalho, as acurácias caíram significativamente, e no teste com o *dataset* de mentiras, caíram ainda mais. Ou seja, as redes que já eram incapazes de reconhecer dados com emoções específicas falharam ainda mais no caso de detectar o estresse numa mentira, o qual representa uma tarefa mais desafiadora.

Dessa forma, as análises feitas nesse capítulo permitiram um entendimento de como as redes neurais desenvolvidas se comportaram com os dados extraídos. A observação dos processos de treinamento, por exemplo, fez possível a melhor interpretação de resultados encontrados. Da mesma maneira, os valores obtidos evidenciaram as consequências da falta de variação nos dados de treinamento de uma rede neural. O *dataset* SAVEE, por exemplo, por conter dados menos variados e em menor quantidade, resultou em valores menos confiáveis.

6 Conclusões

De forma geral, a implementação de redes neurais LSTM para detecção de estresse apresenta diversos desafios. A extração de LLDs, por exemplo, exige conhecimentos sólidos sobre análise de sinais e áudio digital, de forma que seja possível a manipulação e o processamento dos dados resultantes de cada característica. Outro fator importante é o conjunto de dados a ser utilizado, visto que resultados confiáveis dependem de treinamentos efetivos. Nesse trabalho, foram analisados os principais fatores que compõem esse tipo de problema.

A partir dos experimentos e resultados obtidos, observou-se que os LLDs são características importantes para a classificação de estresse na voz, uma vez que apresentaram resultados interessantes nas redes neurais desenvolvidas e são amplamente citados na literatura. A extração dessas características é um importante avanço na área de detecção de fala, e junto com as redes neurais LSTM, compõem o estado da arte para a resolução de problemas desse tipo.

As características da voz com melhores resultados nesse trabalho foram as potências logarítmicas das bandas de frequência de Mel e os MFCCS. A partir dos testes realizados, ambas mostraram carregar informações importantes sobre o estresse na voz, atingindo acurácias altas em redes treinadas e validadas com o mesmo *dataset*. No entanto, foram levantadas questões sobre a qualidade das saídas obtidas, considerando os dados utilizados e o processo de treinamento.

Com isso, fez-se evidente a relação entre a qualidade da classificação e a natureza dos dados utilizados. Também foi possível observar a importância da interpretação do processo de aprendizado de uma rede neural, uma vez que são revelados fatores relevantes à confiabilidade da rede, como generalização e convergência.

6.1 Trabalhos futuros

A execução desse trabalho exigiu uma grande quantidade de testes, portanto, a utilização de métodos mais eficientes para refinamento de redes neurais seria de grande relevância. Também seria interessante o uso de métricas como matrizes de confusão para melhor visualização dos resultados.

No contexto dos dados utilizados, uma abordagem relevante seria a consideração da unificação dos *datasets*, de forma a comparar o desempenho obtido em relação aos treinamentos com um único *dataset*. Outra possibilidade é a utilização de técnicas de *data augmentation* nas bases de dados, com o objetivo de melhorar os processos de treinamento

e validação, trazendo robustez às redes desenvolvidas.

Em relação à extração de características, o uso de espectrogramas como entradas é uma solução com potencial, pois apesar de exigirem maior pré-processamento, podem conferir mais confiabilidade às redes neurais treinadas.

Referências

AVILA, A. R. et al. Speech-based stress classification based on modulation spectral features and convolutional neural networks. In: **2019 27th European Signal Processing Conference (EUSIPCO)**. [S.l.: s.n.], 2019. p. 1–5. Nenhuma citação no texto.

Bhatti, M. W.; Yongjin Wang; Ling Guan. A neural network approach for human emotion recognition in speech. In: **2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)**. [S.l.: s.n.], 2004. v. 2, p. II–181. Nenhuma citação no texto.

CARROLL, L. The Voice Lab: Is it just numbers? In: . [S.l.: s.n.], 2011. Nenhuma citação no texto.

COSETL, R. C.; LOPEZ, J. M. D. B. Voice stress detection: A method for stress analysis detecting fluctuations on lippold microtremor spectrum using fft. In: **CONIELECOMP 2011, 21st International Conference on Electrical Communications and Computers**. [S.l.: s.n.], 2011. p. 184–189. Nenhuma citação no texto.

DWARAMPUDI, M.; REDDY, N. Effects of padding on lstms and cnns. In: . [S.l.: s.n.], 2019. Nenhuma citação no texto.

El Ayadi, M.; KAMEL, M. S.; KARRAY, F. Survey on speech emotion recognition: Features, classification schemes, and databases. **Pattern Recognition**, v. 44, n. 3, p. 572 – 587, 2011. ISSN 0031-3203. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0031320310004619>. Nenhuma citação no texto.

EVGENIOU, T.; PONTIL, M. Support vector machines: Theory and applications. In: . [S.l.: s.n.], 2001. v. 2049, p. 249–257. Nenhuma citação no texto.

GAL, Y. A theoretically grounded application of dropout in recurrent neural networks. 12 2015. Nenhuma citação no texto.

GULHANE, Y.; RODE, S.; LADHAKE, S. Application of fuzzy logic in stress analysis. In: . [S.l.: s.n.], 2011. p. 679–685. Nenhuma citação no texto.

HAN, H.; BYUN, K.; KANG, H.-G. A deep learning-based stress detection algorithm with speech signal. In: . [S.l.: s.n.], 2018. p. 11–15. Nenhuma citação no texto.

HARNSBERGER, J. et al. Stress and deception in speech: Evaluating layered voice analysis. **Journal of forensic sciences**, v. 54, p. 642–50, 06 2009. Nenhuma citação no texto.

HAYKIN, S. **Redes Neurais Princípios e Práticas**. [S.l.]: Bookman, 2008. Nenhuma citação no texto.

HINTON, G. et al. Improving neural networks by preventing co-adaptation of feature detectors. **arXiv preprint**, arXiv, 07 2012. Nenhuma citação no texto.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, v. 9, p. 1735–80, 12 1997. Nenhuma citação no texto.

JULIAO, M. et al. Speech features for discriminating stress using branch and bound wrapper search. In: . [S.l.: s.n.], 2015. p. 3–14. ISBN 978-3-319-27652-6. Nenhuma citação no texto.

KINGMA, D.; BA, J. Adam: A method for stochastic optimization. **International Conference on Learning Representations**, 12 2014. Nenhuma citação no texto.

KURNIAWAN, H.; MASLOV, A. V.; PECHENIZKIY, M. Stress detection from speech and galvanic skin response signals. In: **Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems**. [S.l.: s.n.], 2013. p. 209–214. Nenhuma citação no texto.

MARCOLLA, F.; SANTIAGO, R.; DAZZI, R. Novel lie speech classification by using voice stress. In: . [S.l.: s.n.], 2020. p. 742–749. Nenhuma citação no texto.

MCEWEN, B. S. Protective and damaging effects of stress mediators: central role of the brain. **Dialogues Clin Neurosci**, v. 8, n. 4, p. 367–381, 2006. Nenhuma citação no texto.

MENDOZA, E.; CARBALLO, G. Acoustic analysis of induced vocal stress by means of cognitive workload tasks. **J Voice**, v. 12, n. 3, p. 263–273, Sep 1998. Nenhuma citação no texto.

PARISI, G. et al. Continual lifelong learning with neural networks: A review. **Neural Networks**, 02 2018. Nenhuma citação no texto.

POSNER, J.; RUSSELL, J. A.; PETERSON, B. S. The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology. **Dev Psychopathol**, v. 17, n. 3, p. 715–734, 2005. Nenhuma citação no texto.

REIMERS, N.; GUREVYCH, I. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. 07 2017. Nenhuma citação no texto.

ROTHKRANTZ, L.; WEES, J.-W.; VARK, R. Voice stress analysis. In: . [S.l.: s.n.], 2004. v. 3206, p. 449–456. Nenhuma citação no texto.

RUSSELL, J. A circumplex model of affect. **Journal of Personality and Social Psychology**, v. 39, p. 1161–1178, 12 1980. Nenhuma citação no texto.

SALEHINEJAD, H. et al. Recent advances in recurrent neural networks. 12 2017. Nenhuma citação no texto.

SCHULLER, B.; STEIDL, S.; BATLINER, A. The interspeech 2009 emotion challenge. In: . [S.l.: s.n.], 2009. p. 312–315. Nenhuma citação no texto.

SELYE, H. Stress and the general adaptation syndrome annual. In: **Review of Medicine**. [S.l.: s.n.], 1951. v. 2, p. 327–342. Nenhuma citação no texto.

Shanmugasundaram, G. et al. A comprehensive review on stress detection techniques. In: **2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)**. [S.l.: s.n.], 2019. p. 1–6. Nenhuma citação no texto.

SONDHI, S. et al. Voice analysis for detection of deception. In: . [S.l.: s.n.], 2016. p. 1–6. Nenhuma citação no texto.

STREETER, L. A. et al. Pitch changes during attempted deception. **J Pers Soc Psychol**, v. 35, n. 5, p. 345–350, May 1977. Nenhuma citação no texto.

SWAIN, M.; ROUTRAY, A.; KABISATPATHY, P. Databases, features and classifiers for speech emotion recognition: a review. **International Journal of Speech Technology**, v. 21, 01 2018. Nenhuma citação no texto.

TEIXEIRA, J.; FERREIRA, D.; CARNEIRO, S. Análise acústica vocal - determinação do jitter e shimmer para diagnóstico de patalogias da fala. In: . [S.l.: s.n.], 2011. ISBN 978-972-8826-24-6. Nenhuma citação no texto.

TOMBA, K. et al. Stress detection through speech analysis. In: . [S.l.: s.n.], 2018. p. 394–398. Nenhuma citação no texto.

VERVERIDIS, D.; KOTROPOULOS, C. Emotional speech recognition: Resources, features, and methods. **Speech Communication**, v. 48, n. 9, p. 1162 – 1181, 2006. ISSN 0167-6393. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167639306000422>. Nenhuma citação no texto.

VERVERIDIS, D.; KOTROPOULOS, C. A state of the art review on emotional speech databases. 05 2012. Nenhuma citação no texto.

ZHANG, J. et al. Analysis of stress in speech using adaptive empirical mode decomposition. In: . [S.l.: s.n.], 2009. p. 361 – 365. Nenhuma citação no texto.

Zhou, G.; Hansen, J. H. L.; Kaiser, J. F. Nonlinear feature based classification of speech under stress. **IEEE Transactions on Speech and Audio Processing**, v. 9, n. 3, p. 201–216, 2001. Nenhuma citação no texto.

ZUO, X.; FUNG, P. A cross gender and cross lingual study on acoustic features for stress recognition in speech. In: **ICPhS**. [S.l.: s.n.], 2011. Nenhuma citação no texto.

Apêndices

APÊNDICE A – Artigo

Rede Neural para Identificar Nível de Estresse na Voz: uma abordagem testando parâmetros

Eduardo Kohler[1], **Rafael de Santiago**[2]

[1] Departamento de Informática e Estatística
Universidade Federal de Santa Catarina – Florianópolis, SC – Brazil

Abstract. *The development of stress analysis is relevant in areas of research such as psicology, computer science and others. In recent years, the non intrusive analysis of voice features and the use of automatic classification are well discussed topics. In order to get better results, as well as to eliminate human intervention, the use of neural networks has been promising, as they are capable of detecting patterns and making decisions after training. This project has the goal of avaliating the use of LSTM networks and voice features to detect stress, developing a better understanding of the problems involving this area of research. In order to achieve that, different configurations of LSTM neural networks were used, as well as features extracted using OpenSMILE and public databases. With these procedures, the analysis of important concepts such as the efficiency of certain features, the training processes with different datasets and the consequences involving the nature of the data. From the results obtained, it's clear that features including the logarithmic power of Mel-frequency bands and MFCCs are relevant to stress detection, besides that, it's evident that aspects such as variability and quality of data have big impacts on the training process in general.*

Resumo. *O desenvolvimento de métodos para análise de estresse na voz é relevante para diversas áreas, e envolve conceitos da psicologia, computação e análise comportamental. Nos últimos anos, fez-se relevante a análise não intrusiva de características extraídas da voz, assim como o uso de classificadores para resultados automáticos. A fim de se obter melhores apurações, como eliminar a necessidade de intervenção humana no processo, considera-se o uso de rede neurais, as quais a partir de treinamento são capazes de detectar padrões e tomar decisões. Esse projeto tem o objetivo de avaliar o uso de LSTMs na tarefa de detecção de estresse na voz, através da análise de diferentes bases de dados e características extraídas da voz. Para isso, foram utilizadas variações de configurações de redes neurais LSTM e o software OpenSMILE para extração. Com base nesses procedimentos, foi possível a análise de fatores importantes, como a eficácia de diferentes características, os processos de treinamento para diferentes datasets e as consequências da natureza e quantidade de dados utilizados. A partir dos resultados, percebeu-se que características como as potências logarítmicas das bandas de frequência de Mel e os MFCCs são relevantes para a detecção de estresse, e ficou evidenciada a importância de aspectos como quantidade e variabilidade nos dados do processo de treinamento.*

1. Introdução

A detecção de estresse na voz se tornou uma importante ferramenta em áreas relacionadas à psicologia para identificar emoções na fala como raiva e felicidade, assim como na computação, onde pode ser utilizada em sistemas de reconhecimento de voz e aplicações com controle por fala [Cabrera Cosetl and Baez Lopez 2011].

Diversos estudos e equipamentos foram desenvolvidos com o objetivo de detectar micro tremores na voz, ou outros fenômenos fisiológicos resultantes de situações de estresse. Dos dispositivos desta área, o mais conhecido é o polígrafo, o qual registra atividade cardiovascular, padrão respiratório e condutividade da pele, a partir da presunção de que respostas verdadeiras e falsas apresentarão diferenças nessas características. Apesar de ser frequentemente reconhecido como uma ferramenta de detecção de mentira, há muitas controvérsias em relação a sua efetividade [Sondhi et al. 2016]. Além disso, o polígrafo é incapaz de tomar decisões, sendo necessária a análise humana dos dados registrados. Desta forma, expõe-se a pessoa avaliada ao julgamento de outro ser humano, sujeito a falhas e quebra de sigilo.

Desta maneira, faz-se relevante o desenvolvimento de métodos alternativos para detecção de estresse na voz. Um desses é chamado de VSA (do inglês *Voice Stress Analysis*), o qual consiste em procedimentos para medições, sem contato corporal, de respostas psicológicas involuntárias da voz de uma pessoa que está sob estresse [Cabrera Cosetl and Baez Lopez 2011]. A partir disso, é possível treinar modelos para que detectem certos padrões emocionais e consigam inferir previsões sobre os dados analisados, como explicitado no artigo “A neural network approach for human emotion recognition in speech”, de [Bhatti et al. 2004].

Uma das aplicações recentes de detecção de estresse na voz foi desenvolvida por [Marcolla et al. 2020]. Trata-se de um trabalho no qual a partir da extração de características de sinais de voz, foi possível treinar modelos de redes neurais para decidir se uma fala era verdadeira ou não. Os resultados apresentados foram promissores, e evidenciaram a relação entre o estresse na voz e o ato de contar uma mentira, assim como a relevância dos MFCCs como uma característica da voz. No entanto, apesar de ter apresentado metodologias concisas, o trabalho fez uso de poucas amostras, que não eram muito diversas entre si, e utilizou apenas uma característica extraída da voz.

Esse trabalho objetiva avaliar a utilização de LSTMs e diferentes parâmetros extraídos da voz na tarefa de detecção de estresse na voz, a partir de base de dados públicas e os dados coletados no trabalho de [Marcolla et al. 2020].

1.1. Objetivos

O presente trabalho tem como objetivo geral avaliar o uso de LSTMs na tarefa de detecção de estresse na voz, através da análise de diferentes bases de dados e características extraídas da voz.

Os objetivos específicos são:

- Especificar as características da voz e base de dados utilizadas;
- Obter os dados para treinamento, validação e testes e realizar pré-processamento;
- Desenvolver as redes neurais LSTM;

- Realizar experimentos com as características extraídas e as bases de dados especificadas;
- Analisar os resultados obtidos nos experimentos;
- Divulgar os resultados.

2. Método de Pesquisa

Este trabalho seguiu as seguintes etapas:

1. Realização de pesquisas a partir de artigos e livros da área de redes neurais e aprendizado de máquina, de maneira a formar uma base de referências e exemplos, para então ser possível abordar assuntos mais específicos.

2. Estudo com o objetivo de definir as características da voz a serem utilizadas como entradas nas redes, e também como obtê-las a partir das gravações de áudio.

3. Análise de ferramentas e bibliotecas para a implementação das redes neurais e seus distintos modelos, assim como para o tratamento, processamento e normalização dos dados de entrada do sistema.

4. Obtenção dos dados de entrada.

5. Implementação dos modelos e o treinamento das redes neurais.

6. Realização de comparações e considerações relevantes, inclusive em relação ao trabalho de [Marcolla et al. 2020].

7. Documentação do trabalho desenvolvido.

3. Fundamentação Teórica

Nesse capítulo são apresentados conceitos da detecção de estresse pela voz pertinentes ao presente trabalho, mais especificamente, fundamenta-se quais as características da voz pode-se extrair para eventual análise, e quais os métodos de classificação presentes na literatura. Além disso, são conceituados métodos de inteligência artificial, e discutidas as configurações das bases de dados públicas de vozes anotadas.

3.1. Estresse

"Estresse" é uma palavra que geralmente se refere a experiências que causam sentimentos de ansiedade e frustração. Os causadores de estresse mais comuns são aqueles que operam cronicamente, geralmente em níveis baixos, como acontecimentos do dia a dia. No entanto, há também situações extremas que evocam a resposta de "luta ou fuga" no corpo humano, as quais, ao contrário dos aborrecimentos diários, são estressores agudos, portanto suas consequências podem ser mais facilmente detectadas [McEwen 2006].

O responsável pelas mudanças comportamentais e fisiológicas geradas pelo estresse é o sistema nervoso autônomo, o qual pode ser dividido em duas seções, os sistemas simpático e parassimpático. É através do sistema simpático que o corpo humano ativa glândulas e órgãos para defender o corpo de ameaças, ocasionando também reações como aumento da frequência cardíaca, fluxo sanguíneo rápido para os músculos, ativação das glândulas sudoríparas e aumento da frequência respiratória [Kurniawan et al. 2013].

3.2. Estresse na voz

A voz humana pode efetivamente sinalizar o estado psicológico de um indivíduo, seja emocional, físico, intencional ou inconsciente [Sondhi et al. 2016]. Embora a fala seja uma atividade vocal da qual muito seja verbal, há várias vocalizações humanas que são essencialmente não linguísticas, como entonação, qualidade da voz, prosódia, ritmo e pausa. Esses fenômenos representam um sistema de sinalização não verbal, que se entrelaça com o sistema verbal ou linguístico, trazendo entre outras coisas, informações sobre os aspectos fisiológicos e estado psicológico do falante [Rothkrantz et al. 2004]. São esses fenômenos que sofrem as alterações mais relevantes em situações de estresse. Como já citado, umas das respostas do corpo humano a estressores é a tensão muscular, a qual influencia também as cordas vocais, podendo então alterar direta ou indiretamente a produção de fala [Zhang et al. 2009].

3.3. Métodos para Detecção de Estresse

O estresse pode ser detectado através de vários métodos e técnicas, uma vez que sua manifestação pode resultar em diferentes fatores. Alguns dos parâmetros considerados nesse quesito são: expressão facial, mudanças na voz, manifestações comportamentais e emocionais, características físicas e sinais fisiológicos [Shanmugasundaram et al. 2019]. A maioria desses efeitos podem ser medidos e classificados através de sensores e tecnologias modernas, de formas não intrusivas, as quais já foram investigadas extensivamente nas décadas passadas [Kurniawan et al. 2013].

No que se refere à detecção de estresse na voz, há mais de um parâmetro relevante, e diversas técnicas de análise surgiram com o passar do tempo e avanço tecnológico. No passado as principais características analisadas eram as relacionadas à prosódia, como entonação, duração e intensidade. Nos últimos 20 anos, a maioria dos estudos utilizam de LLDs (do inglês *Low Level Descriptors*) para extrair informações da fala, como a frequência fundamental, jitter e shimmer, energia, e outros [Schuller et al. 2009].

3.4. Detecção de estresse na voz

A detecção de estresse pela voz é um caso específico da detecção de emoções, e para isso, há mais de um método adequado, e diversos parâmetros a serem analisados. Dentre esses parâmetros, o com maior consenso e mais utilizado é a frequência fundamental, ou F0 [Juliao et al. 2015]. Entretanto, métricas para medição de energia e frequências como os MFCC (do inglês *Mel-frequency cepstrum coefficients*) também foram propostas e analisadas, como no artigo de Marcolla et al. 2020. Outro parâmetro relevante é o TEO (do inglês *Teager Energy Operator*), o qual, de acordo com Zhou et al. 2001, é o parâmetro que melhor consegue refletir a estrutura da fala sob condições de estresse. Apesar de serem usados para o mesmo propósito, os diferentes métodos de análise de características da voz podem derivar de diferentes modelos. Métricas como MFCC, por exemplo, derivam de um modelo de produção de fala linear, enquanto de acordo com a teoria por trás dos TEO, a origem dos sons no contexto da fala são interações não lineares [Zuo and Fung 2011].

Um dos conceitos adequados para a análise da voz reside na utilização de métodos para transformar o sinal da fala em um domínio de frequências, que então podem ser comparadas de acordo com o estado do indivíduo analisado. No artigo “Voice Stress

Detection: A method for stress analysis detecting fluctuations on Lippold microtremor spectrum using FFT", de Cabrera Cosetl and Baez Lopez 2011, usou-se FFT sobre o sinal da voz demodulado e um algoritmo para detecção de frequências dominantes, e através desses foi possível observar que os componentes de frequência entre 8 e 12 Hz apresentaram uma diminuição de magnitude quando uma pessoa estava sob estresse.

Zuo and Fung 2011 no estudo "A Cross Gender And Cross Lingual Study On Acoustic Features For Stress Recognition In Speech", observaram parâmetros como MFCCs, TEOs e F0 (frequência fundamental) e compararam seus comportamentos na voz humana sob estresse. Após análise e classificações, os autores chegaram à conclusão que os parâmetros mais precisos para medição de estresse foram os MFCCs e TEOs. Além disso, concluiu-se que os índices de acurácia na classificação aumentaram em sistemas dependentes do gênero do entrevistado.

3.5. Métodos automatizados

Após a extração de sinais e parâmetros da voz, o próximo passo para a detecção de estresse é a classificação e eventual decisão sobre a presença ou não de estresse na voz. Para isso, existem diferentes métodos e algoritmos, a seguir serão citados alguns dos mais abordados em estudos relacionados.

Artigos como os de Juliao et al. 2015, Kurniawan et al. 2013 e Zuo and Fung 2011 utilizaram de SVMs (do inglês *Support Vector Machine*) para classificar e obter decisões sobre as características extraídas da voz. SVM é um conceito desenvolvido no âmbito da teoria de aprendizado estatístico, e é utilizado em áreas como reconhecimento facial e processamento de dados biológicos para diagnóstico médico, por exemplo. O funcionamento de uma SVM é o seguinte: a partir de um conjunto de dados de treinamento, e da medição da disparidade entre os valores esperados e os previstos pela máquina, busca-se por uma função que minimize essa diferença [Evgeniou and Pontil 2001]. No caso da detecção de estresse na voz, utiliza-se dos parâmetros extraídos de falas para o treinamento da SVM, a qual é responsável por classificar e inferir sobre os áudios analisados.

Outro método de classificação é exposto nos artigos de Marcolla et al. 2020 e Han et al. 2018, os quais fazem uso de redes neurais artificiais com arquitetura LSTM (do inglês *Long Short-Term Memory*) para tirar conclusões sobre os dados extraídos. As redes neurais artificiais funcionam a partir de treinamento com dados, assim como SVMs, mas diferem em alguns fatores, como estrutura e quantidade de dados necessária, por exemplo.

3.6. Long Short-Term Memory (LSTM)

As redes neurais LSTM foram propostas por Hochreiter and Schmidhuber 1997 com o objetivo de evitar que sinais importantes desapareçam da memória do sistema com o passar do tempo, o que pode comprometer a rede neural caso essa necessite de uma informação de um passado distante. Esse é um problema comum nas redes neurais recorrentes, denominado "problema do desaparecimento do gradiente". Para mitigar esse comportamento, as LSTM utilizam de unidades de processamento chamadas de "células de memória", as quais através de "portões" que manipulam a memória, lembram ou esquecem informações ao longo do tempo. Cada célula contém três tipos de portões em sua estrutura, um *forget gate*, um *input gate* e um *output gate*. O *forget gate* é responsável por remover as informações que não são mais úteis ao estado da célula, o *input gate* realiza

a adição de informação ao estado, e o *output gate* extrai informações úteis do estado da célula para formar um sinal de saída [Hochreiter and Schmidhuber 1997].

3.7. Bases de Dados sobre Voz Anotadas

Atualmente há diversos laboratórios de pesquisa e empresas com o objetivo de desenvolver soluções na área detecção de emoções na fala. Para que esse desenvolvimento aconteça, a existência de bases de dados de vozes é de grande importância [Ververidis and Kotropoulos 2012]. Além disso, é relevante a análise sobre a modelagem e qualidade de tais bases de dados, uma vez que conclusões incorretas podem aparecer caso uma base de má qualidade seja usada. Alguns dos critérios para o julgamento sobre uma base de dados são a natureza das emoções expressas, as características das pessoas entrevistadas, como idade, gênero e nacionalidade e quais os tipos de frases ou palavras utilizadas [El Ayadi et al. 2011].

Há três tipos de bases de dados para sistemas de reconhecimento de emoções na fala: as com emoções naturais, as com emoções atuadas e as com emoções induzidas. As bases naturais são aquelas que contém falas reais e espontâneas, como gravações de *callcenters*, interações entre paciente e médico e gravações durante situações anormais. As bases com emoções atuadas são aquelas que possuem gravações coletadas a partir de atores profissionais, os quais reproduzem emoções nas falas de maneira simulada. Por último, as bases com emoções induzidas utilizam de métodos para influenciar os entrevistados a expressarem certas emoções. As emoções básicas geralmente usadas na literatura são: raiva, medo, tristeza, prazer sensorial, diversão, satisfação, contentamento, excitação, nojo, desprezo, orgulho, vergonha, culpa, constrangimento e alívio [Swain et al. 2018]. Além disso, sinais complementares como pressão sanguínea, batimentos cardíacos e respiração também podem ser documentados [Ververidis and Kotropoulos 2006].

Alguns exemplos de base de dados de falas são: Belfast Natural Database, Kids' Audio Speech Corpus NSF/ITR Reading Project, Magdeburger Prosodie Korpus, SUSAS (Speech Under Simulated and Actual Stress), SAVEE (Surrey Audio-Visual Expressed Emotion) e RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song).

4. Trabalhos Relacionados

Os trabalhos similares referenciados nesse projeto foram reunidos através da ferramenta SCOPUS, utilizando os termos *voice/speech stress analysis*, *voice/speech stress detection* e *voice/speech stress* como palavras-chave para busca. Os critérios para escolha foram: a utilização de redes neurais para classificação e a detecção de estresse exclusivamente através da voz.

4.1. Stress Detection Through Speech Analysis

No artigo de Tomba et al. 2018 foram analisados parâmetros para a detecção de estresse na voz, assim como métodos para a classificação. Os parâmetros escolhidos foram características do sinal de áudio gerado pela voz, uma vez que é um processo não intrusivo e extensamente abordado na literatura relacionada. As características analisadas foram energia média, intensidade média e MFCCs, e as estratégias de classificação testadas foram SVMs e RNAs.

Concluiu-se que os MFCCs são bons parâmetros para medição de estresse por voz, principalmente quando considerados a média e desvio padrão. Além disso, os autores propuseram como melhorias para o trabalho exposto o aumento do número de características da voz, assim como a utilização de bases de dados mais amplas, incluindo casos reais de estresse.

4.2. A Deep Learning-based Stress Detection Algorithm with Speech Signal

Nesse artigo, Han et al. 2018 propuseram um algoritmo para detecção de estresse pela voz utilizando *deep learning*. Para isso, combinou-se uma rede neural LSTM e dois tipos de classificadores, utilizando de coeficientes *mel-filterbank* extraídos da voz como os parâmetros a serem analisados. Os classificadores usados foram uma camada SVM e uma camada *softmax*, as quais tiveram os resultados comparados posteriormente.

Dentre esses, o que obteve maior acurácia foi o LSTM com SVM e média dos frames, com 66,4% de precisão. Observou-se também que a utilização da média dos frames, em vez do último frame, é a alternativa mais eficiente. Para aumentar a acurácia em trabalhos futuros, os autores propuseram uma abordagem multimodal.

4.3. Speech-Based Stress Classification based on Modulation Spectral Features and Convolutional Neural Networks

O trabalho exposto nesse artigo, de Avila et al. 2019, tem como principal objetivo validar a utilização de uma rede neural convolucional (CNN, do inglês *convolutional neural network*) para detecção de estresse na voz. Para isso, usou-se características espectrais de modulação e a base de dados Speech Under Simulated and Actual Stress (SUSAS). Os resultados obtidos pelo modelo proposto foram comparados com outros dois sistemas, um desses com classificador SVM, e outro com uma rede neural profunda (DNN, do inglês *deep neural network*). Ambos utilizaram características extraídas com a ferramenta OpenSMILE.

Os autores concluíram que os coeficientes espectrais de modulação podem ser úteis para a detecção de estresse, principalmente com a utilização de redes neurais convolucionais e redes neurais profundas.

4.4. Speech-Based Stress Classification based on Modulation Spectral Features and Convolutional Neural Networks

O trabalho exposto no artigo de Marcolla et al. 2020 analisou a utilização de redes neurais para a detecção de mentiras pela voz. A arquitetura considerada foi a rede neural recorrente LSTM, e as características extraídas da voz foram os MFCCs.

Concluiu-se que há a possibilidade de detecção de mentiras pela voz a partir de redes neurais, e que para a obtenção de resultados mais relevantes nessa área, pode-se considerar a utilização de bases de dados mais volumosos, assim como a consideração de diferentes idiomas e sotaques.

5. Desenvolvimento

5.1. Extração de características da voz

Para a extração de características da voz, usou-se o software de código aberto OpenSMILE (Open-Source Media Interpretation by Large feature-space Extraction), o qual

é utilizado amplamente na área de análise de sinais e machine learning. Através de métodos de processamento de sinais, o OpenSMILE é capaz de realizar diversos cálculos e transformações sobre arquivos de áudio, de forma a extrair uma grande variedade de LLDs.

Os LLDs extraídos foram:

- MFCCs: Coeficientes cepstrais de frequência de Mel (do inglês *Mel-frequency cepstral coefficients*);
- logMelFreqBand: Potência logarítmica das bandas de frequência de Mel (do inglês *Logarithmic power of Mel-frequency bands*);
- lspFreq: 8 Frequências espectrais computadas a partir de 8 coeficientes de predição (do inglês *8 line spectral pair frequencies computed from 8 LPC coefficients*);
- F0finEnv: Envelope do contorno suavizado da frequência fundamental (do inglês *Envelope of the smoothed fundamental frequency contour*);
- voicingFinalUnclipped: A probabilidade de vocalização do candidato final da frequência fundamental (do inglês *The voicing probability of the final fundamental frequency candidate*);

5.2. Pré-processamento das características extraídas

Com o objetivo de organizar os dados extraídos em um formato compatível com redes neurais recorrentes LSTM, foram aplicados processos de ordenamento e normalização. Primeiramente, os dados contidos nos arquivos .csv são lidos de acordo com as características escolhidas para teste, em seguida, essas características são carregadas em um uma matriz tridimensional com dimensões nº de áudios x nº de frames x nº de características. Os métodos utilizados foram: *padding*, *masking* e *one-hot encoding*.

5.2.1. Datasets

As etapas de treinamento, validação e teste das redes neurais desenvolvidas foram realizadas a partir de dados adquiridos de base de áudios públicas especializadas em emoções. As bases utilizadas foram a SAVEE (Surrey Audio-Visual Expressed Emotion) e a RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song), as quais apresentam dados sobre diferentes emoções. Durante a busca por dados, não foram encontradas bases públicas que lidassem com estresse especificamente. Dessa forma, fez-se necessário o aprofundamento em conceitos de psicologia e neurociência, com o objetivo de correlacionar as emoções encontradas nas bases de dados com o estresse.

Na base de dados SAVEE, há um total 480 arquivos de áudio, divididos igualmente entre os 4 atores. Para cada falante, há 15 gravações diferentes para cada emoção e 30 áudios da categoria neutra, totalizando 120. Durante o processo de formação do *dataset*, utilizou-se os 15 áudios disponíveis da emoção raiva, os 15 da emoção ódio e os 30 das falas neutras de cada ator. Como resultado, obteu-se um *dataset* com 240 arquivos de áudio, igualmente dividido entre as categorias sem e com presença de estresse.

Através da base de dados RAVDESS, outro *dataset* foi construído de maneira semelhante. Nessa base há um total de 1440 arquivos, separados igualmente entre 24

atores, 12 do sexo masculino e 12 do sexo feminino. Para cada falante, a base disponibiliza 4 áudios neutros e 8 de cada emoção abordada. Dessa forma, foi possível a obtenção de 12 áudios da categoria sem estresse para cada ator, totalizando 288 áudios. Para a categoria com a presença de estresse, selecionou-se 12 áudios das emoções raiva e medo, divididos igualmente entre as duas emoções e os níveis de intensidade, gerando um total de 288 áudios para essa categoria, e 576 para todo o *dataset*. As características de cada *dataset* estão expostas na Tabela 1.

Table 1. Comparação entre *datasets*

Dataset	**Idioma**	**Nº de amostras**	**Nº de atores**	**Emoções**	**Tipo de amostra**	**Base de dados**
Dataset 1	Inglês	240	4	Neutro, raiva e medo	Expressões	SAVEE
Dataset 2	Inglês	576	24	Neutro, calma, raiva e medo	Expressões	RAVDESS

Durante a etapa de treinamento e testes, os *datasets* foram divididos em subconjuntos de treinamento, validação e teste. A separação desses subconjuntos se deu da seguinte forma: do total de áudios, 25% foram utilizados para testes, e dos 75% restantes, 80% formaram o conjunto de treinamento e 20% o conjunto de validação.

5.3. Redes Neurais LSTM

Durante o processo de refinamento das redes desenvolvidas, os hiperparâmetros manipulados foram: número de unidades de processamento das células LSTM, taxa de aprendizado, *dropout*, tamanho de *batch* e épocas de treinamento. Além disso, foram testadas diversos tipos de entrada a partir das características (LLDs) extraídas dos áudios. O processo de treinamento e refinamento das redes neurais testadas foi documentado e discutido no capítulo seguinte.

6. Experimentos

Os experimentos realizados nesse trabalho consistiram nas seguintes etapas:

1. Testes de entradas e configurações de LSTM com o *dataset* da base RAVDESS;
2. Testes de entradas e configurações de LSTM com o *dataset* da base SAVEE;
3. Teste da LSTM com maior acurácia da primeira etapa com dados da base da segunda etapa;
4. Teste da LSTM com maior acurácia da segunda etapa com dados da base da primeira etapa;
5. Teste de ambas as LSTMs com maior acurácia com os dados coletados no trabalho de [Marcolla et al. 2020];

6.1. Entradas e configurações de LSTM

As duas primeiras etapas foram executadas a partir de uma série de testes envolvendo configurações de redes neurais LSTM e os dados contidos nos *datasets*. Através do conjunto de características extraídas de cada áudio e da customização dos hiperparâmetros das redes, foi possível a análise de combinações de diferentes entradas com configurações de LSTMs. As características da voz analisadas em cada *dataset* foram as seguintes:

- MFCCs: Coeficientes cepstrais de frequência de Mel (do inglês *Mel-frequency cepstral coefficients*);

- logMelFreqBand: Potência logarítmica das bandas de frequência de Mel (do inglês *Logarithmic power of Mel-frequency bands*);
- lspFreq: 8 Frequências espectrais computadas a partir de 8 coeficientes de predição (do inglês *8 line spectral pair frequencies computed from 8 LPC coefficients*);
- F0finEnv: Envelope do contorno suavizado da frequência fundamental (do inglês *Envelope of the smoothed fundamental frequency contour*);
- voicingFinalUnclipped: A probabilidade de vocalização do candidato final da frequência fundamental (do inglês *The voicing probability of the final fundamental frequency candidate*);

Os hiperparâmetros das LSTMs considerados foram:

- Número de camadas LSTM;
- Número de unidades de processamento das células LSTM;
- Taxa de aprendizado;
- *Dropout*;
- Tamanho dos *batches*;
- Número de épocas de treinamento;

6.2. Redes treinadas com RAVDESS

Abaixo estão os resultados encontrados a partir das redes neurais LSTM treinadas com dados do *dataset* da base RAVDESS. Na tabela abaixo estão presentes a melhores configuração de LSTM encontrada para cada parâmetro testado, considerando a acurácia obtida nos testes.

Table 2. Melhores resultados por parâmetro das redes treinadas no RAVDESS por ordem de acurácia

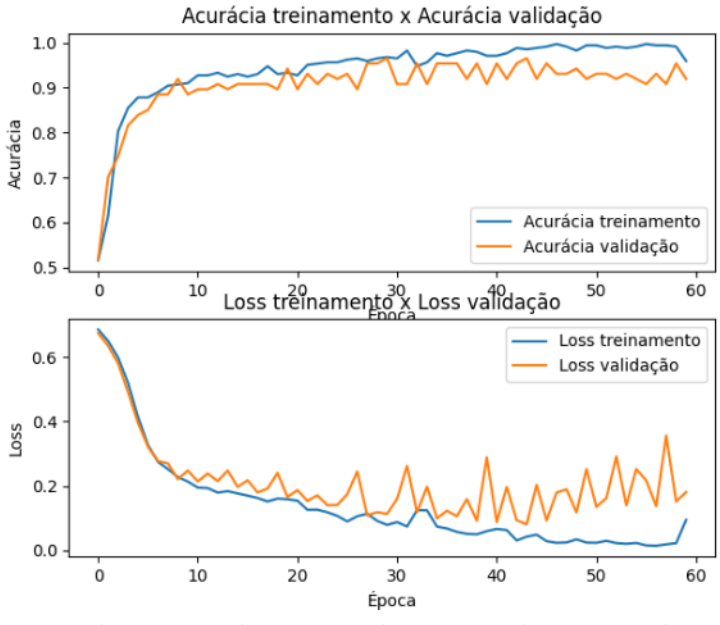
Parâmetros	Camadas	Unidades	Dropout	Taxa de aprendizado	Épocas	Batch	Acurácia(%)
logMelFreqBand	3	200	0.3	0.00008	60	64	**97,91**
MFCCs	3	100	0.3	0.00005	150	32	**96,527**
lspFreq	3	200	0.0	0.00005	150	64	**89,583**
F0finEnv	2	64	0.0	0.0001	100	32	**84,722**
voicingFinalUnclipped	3	64	0.0	0.0006	100	32	**65,972**

Dentre os parâmetros testados, os que atingiram maior acurácia na etapa de teste foram as potências logarítmicas das bandas de frequência de Mel, com 97,91%. Esse valor foi atingido a partir de uma taxa de aprendizado 0.00008, 3 camadas LSTM e 200 unidades de processamento nas células LSTM. O treinamento dessa rede apresentou *overfitting* ao decorrer das épocas, dessa forma, a acurácia foi melhorada a partir da interrupção do treinamento antes do desenvolvimento de *overfitting*. Além disso, a utilização de *dropout* foi benéfica nessa rede, e ajudou no desempenho da taxa de *loss* do conjunto de validação.

Os MFCCs também se mostraram uma característica eficiente, pois demonstraram uma eficácia de 96,527% na etapa de testes. Esse modelo atingiu a maior acurácia com 150 épocas de treinamento, e o uso de *dropout* também foi importante para estabilizar a taxa de *loss* de conjunto de validação durante o treinamento.

Apesar das características testadas terem apresentados acurácias com valores parecidos, é importante a consideração da qualidade das acurácias obtidas, isto é, a capacidade da rede neural conseguir classificar dados externos aos dados de treinamento. Para isso, analisou-se o comportamento da rede em relação ao conjunto de validação durante o treinamento. No caso da rede das potências logarítmicas das bandas de frequência de Mel, o modelo com maior acurácia apresentou o comportamento apresentado na Figura 1.

Figure 1. Gráficos de treinamento de uma rede treinada com logMelFreqBand da base RAVDESS



Pode-se observar que a acurácia do conjunto de validação aumentou de maneira semelhante ao conjunto de teste, porém apresentou uma maior instabilidade. No caso da taxa de *loss*, um comportamento parecido foi obtido, com os valores de treinamento e acurácia decrescendo juntos, mas com o de validação mostrando instabilidade. Uma forma de lidar com essa instabilidade é diminuir a robustez da rede, para que essa possa convergir mais rapidamente, outra possibilidade seria diminuir a taxa de aprendizado, para que o otimizador encontre um ponto ótimo de maneira mais controlada. Ambas essas soluções foram testadas nesse caso, e apesar de terem demonstrado uma melhora no processo de aprendizado de acordo com os gráficos, não apresentaram uma maior acurácia. Na Figura 2 está representado o comportamento desse modelo com menos unidades de processamento e com uma taxa de aprendizado de 0.00001.

Com os MFCCs como entrada, o comportamento dos modelos desenvolvidos foi diferente. A maior acurácia obtida foi após o treinamento por 150 épocas, com o valor de acurácia do conjunto de validação eventualmente convergindo. A partir da variação da taxa de *loss*, percebe-se que o modelo sofreu *overfitting*, com essa taxa subindo com o andamento das épocas. A utilização do *dropout* ajudou no controle desse crescimento. A Figura 3 se trata de uma comparação entre um treinamento desse modelo com e sem *dropout*.

Figure 2. Gráficos de treinamento de uma rede treinada com logMelFreqBand da base RAVDESS

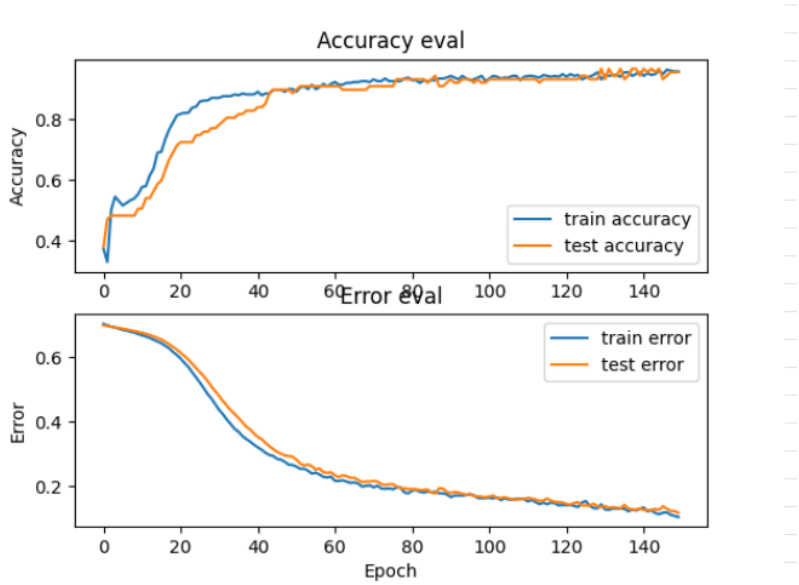
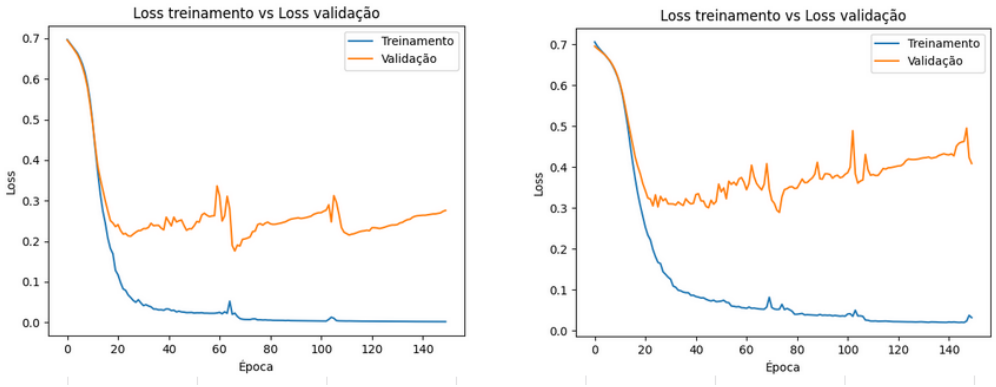


Figure 3. Gráficos de loss durante o treinamento da rede treinadas com MFCCs da base RAVDESS com e sem dropout



No modelo com maior acurácia utilizando as 8 frequências espectrais como entrada, o comportamento obtido durante o treinamento foi bastante instável, e a partir da variação dos hiperparâmetros, não foi encontrada uma configuração estável com maior acurácia. Esse comportamento foi o mesmo observado nas redes do envelope da frequência fundamental e da probabilidade de vocalização. Em ambos os casos as redes não convergiram, e apresentaram altas taxas de *loss* no conjunto de validação durante o treinamento.

6.3. Redes treinadas com SAVEE

O treinamento de redes neurais LSTM com dados do *dataset* SAVEE estão representados na tabela abaixo. Assim como no exemplo anterior, a Tabela 3 apresenta a melhores configuração de LSTM encontrada para cada parâmetro testado, considerando a acurácia obtida nos testes.

Nesses experimentos, a rede com maior acurácia foi treinada com as potências

Table 3. Melhores resultados por parâmetro das redes treinadas no SAVEE por ordem de acurácia

Parâmetros	Camadas	Unidades	Dropout	Taxa de aprendizado	Épocas	Batch	Acurácia(%)
logMelFreqBand	3	100	0.3	0.00005	150	64	**100**
MFCCs	2	100	0.0	0.00005	100	64	**93,333**
F0finEnv	3	100	0.0	0.00001	100	32	**91,666**
lspFreq	1	50	0.0	0.001	150	32	**89,999**
voicingFinalUnclipped	1	64	0.0	0.001	150	32	**78,333**

logarítmicas, utilizando 3 camadas de células LSTM, com 100 unidades de processamento e uma taxa de aprendizado de 0,00005. Essa rede atingiu acurácia de 100% no conjunto de teste, o que indica um problema no treinamento, visto que é muito incomum um classificador atingir um valor de 100% sem alguma inconsistência no processo de aprendizado. Nesse caso, uma possível explicação é a falta de diversidade do banco de dados utilizado, visto que o SAVEE contém áudios de apenas 4 atores, todos do sexo masculino. Com isso, o conjunto usado para teste pode ser muito semelhante ao usado no treinamento, de forma que o modelo consiga uma acurácia perfeita até mesmo com dados com os quais não foi treinado. Esse processo de treinamento indica uma rede neural com pouca qualidade em suas saídas.

A segunda maior acurácia foi obtida a partir dos MFCCs. A rede neural obtida contém duas camadas de células LSTM e uma taxa de aprendizado de 0,00005, atingindo 93,333% de acurácia. O gráfico do processo de treinamento dessa rede apresentou um comportamento diferente do observado com o *dataset* RAVDESS, nesse caso a taxa de *loss* do conjunto de validação acompanhou a taxa de *loss* do treinamento. A Figura 4 representa o processo de treinamento para essa rede.

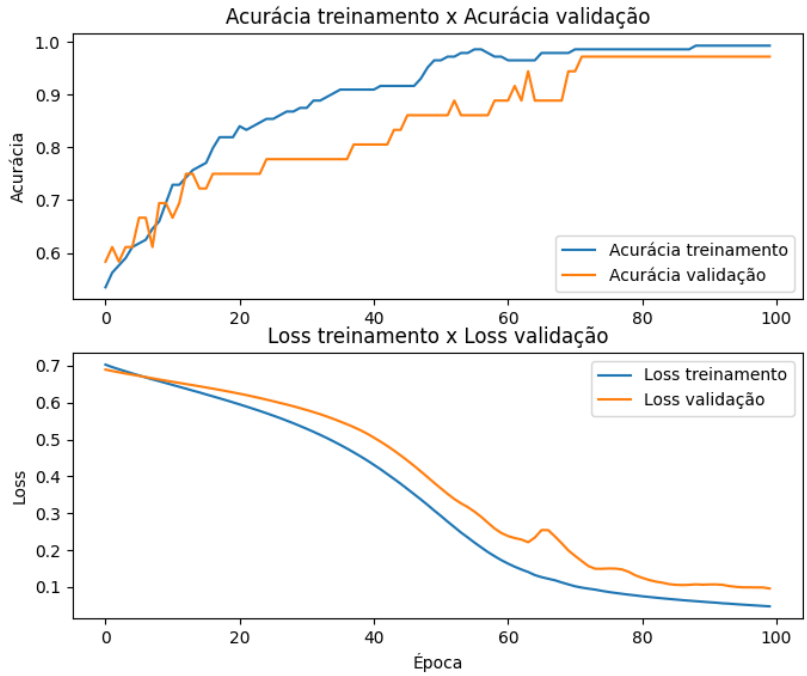
O envelope da frequência fundamental foi a característica com a terceira maior acurácia. A rede configurada para esse parâmetro utilizou de uma taxa de aprendizado de valor 0,00001, com 3 camadas de células LSTM com 100 unidades de processamento. O processo de aprendizado dessa rede também apresentou diferenças em relação à rede do *dataset* anterior. Assim como no caso dos MFCCs, essa rede apresentou uma taxa de *loss* menor ao decorrer das épocas.

No caso das 8 frequências espectrais, a acurácia obtida foi de 89,999%, a partir de uma rede com 1 camada de células LSTM com 50 unidades de processamento. A taxa de aprendizado utilizada foi a de 0,001. Essa rede apresentou um treinamento instável, com os valores da taxa de *loss* não convergindo para um valor menor. O mesmo comportamento foi obtido com as probabilidades de vocalização, as quais apesar da acurácia obtida, não resultaram em um treinamento efetivo da rede neural.

6.4. Comparativo

Os experimentos realizados com os dois *datasets* apresentaram aspectos em comum e diferenças, os quais foram evidenciados nos resultados e análises. Em ambos os *datasets*, a melhor acurácia foi obtida através das potências logarítmicas das bandas de frequência de Mel, com os MFCCs apresentando a segunda melhor precisão. No entanto, observou-se diferenças no comportamento dos treinamentos, as quais estão relacionadas com a natureza e quantidade dos dados utilizados. No caso do *dataset* com dados da base SAVEE, percebeu-se que as redes foram influenciadas pela baixa variação dos dados, gerando

Figure 4. Gráficos de loss durante o treinamento da rede treinadas com MFCCs da base SAVEE



saídas de menor qualidade. Com isso, faz-se relevante testes com dados de *datasets* diferentes dos utilizados para a formação dos conjuntos de treino, validação e teste das redes desenvolvidas.

7. Testes entre *datasets*

Com o objetivo de validar a qualidade das redes neurais treinadas com as características analisadas nesse trabalho, foram realizados testes entre as redes LSTM apresentadas anteriormente e os dois *datasets* montados. Com isso, as redes treinadas com o *dataset* da base RAVDESS foram sumbmetidas a um conjunto de teste com dados do *dataset* da base SAVEE, e as redes treinadas com dados da base SAVEE foram testados com áudios da base RAVDESS.

Esse processo se deu a partir do armazenamento das redes com melhor acurácia e da adaptação de suas camadas de entrada, de forma a permitir o processamento das entradas geradas pelo *dataset* usado para o teste, as quais, após o processo de normalização, contêm um tamanho diferente das entradas com qual a rede foi treinada.

7.1. Redes treinadas em RAVDESS e testadas com o dataset SAVEE

A partir dos resultados apresentados na Tabela 4, percebe-se que a rede treinada com as potências logarítmicas apresentou a melhor generalização. No geral, os valores de *loss* foram altos, portanto os modelos não convergiram o suficiente para serem totalmente confiáveis com dados externos. Isso se dá devido à forma como a rede se comportou durante o treinamento, ou seja, a maneira como seus pesos internos foram ajustados de acordo com as entradas. Como observado anteriormente, a rede treinada com potências

logarítmicas obteve o maior nível de convergência, e isso se fez evidente no fato de essa rede ter apresentado a maior acurácia nesse teste.

Table 4. Resultados das redes treinadas em RAVDESS e testadas com o dataset SAVEE por ordem de acurácia

Rede	Acurácia(%)	Loss
logMelFreqBand	85%	0,6149
MFCCs	63,33%	1,5685
F0finEnv	61,66%	0,8812
voicingFinalUnclipped	53,33%	0,9413
lspFreq	43,33%	1,0243

7.2. Redes treinadas em SAVEE e testadas com o dataset RAVDESS

Na Tabela 5 estão os resultados dos testes das redes treinadas com o *dataset* SAVEE com dados do *dataset* RAVDESS. Pode-se observar uma baixa acurácia e grandes valores de *loss* no geral, resultantes da falta de generalização das redes no processo de treinamento. Como já visto, as redes treinadas pelo *dataset* SAVEE apresentaram saídas de menor qualidade, portanto, suas acurácias apresentaram uma queda significativa quando submetidas a um *dataset* diferente.

Table 5. Resultados das redes treinadas em SAVEE e testadas com o dataset RAVDESS por ordem de acurácia

Rede	Acurácia(%)	Loss
F0finEnv	64,583%	0,9962
MFCCs	60,714%	1,4253
logMelFreqBand	52,083%	2,3486
lspFreq	51,388%	2,3209
voicingFinalUnclipped	45,833%	0,8378

8. Testes com *dataset* de mentiras

Além dos testes com os *datasets* montados nesse trabalho, foram realizados validações usando o conjunto de dados construído por [Marcolla et al. 2020]. Esse conjunto se trata de áudios de expressões divididas entre verdades e mentiras, enunciadas por homens na língua portuguesa. Através desse teste, pode-se medir a eficiência das redes treinadas nesse trabalho na classificação do estresse contido no ato de contar uma mentira.

8.1. Redes treinadas com RAVDESS testadas com dados do dataset de mentiras

A Tabela 8.1 abaixo apresenta os valores de acurácia e *loss* obtidos pelas redes treinadas com o *dataset* RAVDESS quando submetidas ao dados de [Marcolla et al. 2020].

Os resultados apresentados na Tabela 8.1 evidenciam que as redes treinadas não foram capazes de classificar com precisão os áudios da base testada. Assim como no teste anterior, a rede treinada com as potências logarítmicas apresentou a melhor acurácia,

Table 6. Resultados das redes treinadas em RAVDESS e testadas com o dataset SAVEE por ordem de acurácia

Rede	Acurácia(%)	Loss
logMelFreqBand	60%	1,37
voicingFinalUnclipped	46%	0,8770
lspFreq	46%	1,026
MFCCs	34%	3,1970
F0finEnv	30%	1,7863

no entanto, o valor do *loss* registrado nesse teste foi significativamente maior do que o anterior, ou seja, a classificação cometeu erros mais graves. Com isso, evidencia-se que o problema de classificar falas como mentira ou verdade exige mais robustez das redes, de forma que essas sejam capazes de detectar padrões numa fala mentirosa a partir do estresse nessa contida.

8.2. Redes treinadas com SAVEE testadas com dados do dataset de mentiras

A Tabela 7 apresenta os resultados dos testes envolvendo as redes treinadas com o *dataset* SAVEE e os dados de Marcolla et al. 2020.

Table 7. Resultados das redes treinadas em SAVEE e testadas com o dataset RAVDESS por ordem de acurácia

Rede	Acurácia(%)	Loss
lspFreq	43,999%	1,4584
voicingFinalUnclipped	41,999%	0,8565
logMelFreqBand	41,999%	1,7695
MFCCs	36,000%	1,7606
F0finEnv	34,000%	1,3167

Como pode-se observar, as acurácias obtidas foram muito baixas, enquanto os valores de *loss* foram altos. Assim como no teste anterior, as redes foram incapazes de generalizar o conhecimento adquirido.

9. Discussões

A realização dos experimentos documentados permitiu a visualização de fenômenos importantes no aprendizado das redes neurais. A partir da análise dos processos de treinamento, evidenciou-se que dependendo do *dataset* utilizado, os mesmos dados de entrada podem gerar comportamentos diferentes. Além disso, a natureza e quantidade dos dados se fez um fator relevante na interpretação das saídas geradas pelas redes neurais desenvolvidas.

Em ambos os *datasets* testados, as redes apresentaram boas acurácias no geral independente do processo de treinamento, mas quando submetidas a dados de outro *dataset*, resultaram em acurácias mais baixas. No caso dos testes entre os *datasets* desse trabalho, as acurácias caíram significativamente, e no teste com o *dataset* de mentiras, caíram ainda

mais. Ou seja, as redes que já eram incapazes de reconhecer dados com emoções específicas falharam ainda mais no caso de detectar o estresse numa mentira, o qual representa uma tarefa mais desafiadora.

Dessa forma, as análises feitas nesse capítulo permitiram um entendimento de como as redes neurais desenvolvidas se comportaram com os dados extraídos. A observação dos processos de treinamento, por exemplo, fez possível a melhor interpretação de resultados encontrados. Da mesma maneira, os valores obtidos evidenciaram as consequências da falta de variação nos dados de treinamento de uma rede neural. O *dataset* SAVEE, por exemplo, por conter dados menos variados e em menor quantidade, resultou em valores menos confiáveis.

10. Conclusões

De forma geral, a implementação de redes neurais LSTM para detecção de estresse apresenta diversos desafios. A extração de LLDs, por exemplo, exige conhecimentos sólidos sobre análise de sinais e áudio digital, de forma que seja possível a manipulação e o processamento dos dados resultantes de cada característica. Outro fator importante é o conjunto de dados a ser utilizado, visto que resultados confiáveis dependem de treinamentos efetivos. Nesse trabalho, foram analisados os principais fatores que compõem esse tipo de problema.

A partir dos experimentos e resultados obtidos, observou-se que os LLDs são características importantes para a classificação de estresse na voz, uma vez que apresentaram resultados interessantes nas redes neurais desenvolvidas e são amplamente citados na literatura. A extração dessas características é um importante avanço na área de detecção de fala, e junto com as redes neurais LSTM, compõem o estado da arte para a resolução de problemas desse tipo.

As características da voz com melhores resultados nesse trabalho foram as potências logarítmicas das bandas de frequência de Mel e os MFCCS. A partir dos testes realizados, ambas mostraram carregar informações importantes sobre o estresse na voz, atingindo acurácias altas em redes treinadas e validadas com o mesmo *dataset*. No entanto, foram levantadas questões sobre a qualidade das saídas obtidas, considerando os dados utilizados e o processo de treinamento.

Com isso, fez-se evidente a relação entre a qualidade da classificação e a natureza dos dados utilizados. Também foi possível observar a importância da interpretação do processo de aprendizado de uma rede neural, uma vez que são revelados fatores relevantes à confiabilidade da rede, como generalização e convergência.

11. Trabalhos futuros

A execução desse trabalho exigiu uma grande quantidade de testes, portanto, a utilização de métodos mais eficientes para refinamento de redes neurais seria de grande relevância. Também seria interessante o uso de métricas como matrizes de confusão para melhor visualização dos resultados.

No contexto dos dados utilizados, uma abordagem relevante seria a consideração da unificação dos *datasets*, de forma a comparar o desempenho obtido em relação aos treinamentos com um único *dataset*. Outra possibilidade é a utilização de técnicas de *data*

augmentation nas bases de dados, com o objetivo de melhorar os processos de treinamento e validação, trazendo robustez às redes desenvolvidas.

Em relação à extração de características, o uso de espectrogramas como entradas é uma solução com potencial, pois apesar de exigirem maior pré-processamento, podem conferir mais confiabilidade às redes neurais treinadas.

References

Avila, A. R., Kshirsagar, S. R., Tiwari, A., Lafond, D., O'Shaughnessy, D., and Falk, T. H. (2019). Speech-based stress classification based on modulation spectral features and convolutional neural networks. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5.

Bhatti, M. W., Yongjin Wang, and Ling Guan (2004). A neural network approach for human emotion recognition in speech. In *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*, volume 2, pages II–181.

Cabrera Cosetl, R. and Baez Lopez, J. M. D. (2011). Voice stress detection: A method for stress analysis detecting fluctuations on lippold microtremor spectrum using fft. In *CONIELECOMP 2011, 21st International Conference on Electrical Communications and Computers*, pages 184–189.

El Ayadi, M., Kamel, M. S., and Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572 – 587.

Evgeniou, T. and Pontil, M. (2001). Support vector machines: Theory and applications. volume 2049, pages 249–257.

Han, H., Byun, K., and Kang, H.-G. (2018). A deep learning-based stress detection algorithm with speech signal. pages 11–15.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.

Juliao, M., Silva, J., Aguiar, A., Moniz, H., and Batista, F. (2015). Speech features for discriminating stress using branch and bound wrapper search. pages 3–14.

Kurniawan, H., Maslov, A. V., and Pechenizkiy, M. (2013). Stress detection from speech and galvanic skin response signals. In *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, pages 209–214.

Marcolla, F., Santiago, R., and Dazzi, R. (2020). Novel lie speech classification by using voice stress. pages 742–749.

McEwen, B. S. (2006). Protective and damaging effects of stress mediators: central role of the brain. *Dialogues Clin Neurosci*, 8(4):367–381.

Rothkrantz, L., Wees, J.-W., and Vark, R. (2004). Voice stress analysis. volume 3206, pages 449–456.

Schuller, B., Steidl, S., and Batliner, A. (2009). The interspeech 2009 emotion challenge. pages 312–315.

Shanmugasundaram, G., Yazhini, S., Hemapratha, E., and Nithya, S. (2019). A comprehensive review on stress detection techniques. In *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, pages 1–6.

Sondhi, S., Vijay, R., Khan, M., and Salhan, A. (2016). Voice analysis for detection of deception. pages 1–6.

Swain, M., Routray, A., and Kabisatpathy, P. (2018). Databases, features and classifiers for speech emotion recognition: a review. *International Journal of Speech Technology*, 21.

Tomba, K., Dumoulin, J., Mugellini, E., Abou Khaled, O., and Hawila, S. (2018). Stress detection through speech analysis. pages 394–398.

Ververidis, D. and Kotropoulos, C. (2006). Emotional speech recognition: Resources, features, and methods. *Speech Communication*, 48(9):1162 – 1181.

Ververidis, D. and Kotropoulos, C. (2012). A state of the art review on emotional speech databases.

Zhang, J., Mbitiru, N., Tay, P., and Adams, R. (2009). Analysis of stress in speech using adaptive empirical mode decomposition. pages 361 – 365.

Zhou, G., Hansen, J. H. L., and Kaiser, J. F. (2001). Nonlinear feature based classification of speech under stress. *IEEE Transactions on Speech and Audio Processing*, 9(3):201–216.

Zuo, X. and Fung, P. (2011). A cross gender and cross lingual study on acoustic features for stress recognition in speech. In *ICPhS*.

APÊNDICE B – Código-fonte

B.1 Keras

```
import csv
import os
import sys
import numpy as np
import tensorflow as tf
from numpy.random import seed
import tensorflow.keras as keras
from keras.utils import to_categorical
from keras.models import load_model
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

seed(1)
tf.random.set_seed(2)

model_path = sys.argv[1] if len(sys.argv) > 1 else False

DATAPATH = ''

LSTM_CELLS = 3
UNITS_PER_CELL = 100
DROPOUT_RATE = 0.0
LEARNING_RATE = 0.00001
BATCH_SIZE = 32
EPOCHS_N = 150
ACTIVATION_FUNCTION = 'softmax'
LOSS_FUNCTION = 'binary_crossentropy'

PRE_PADDING = True

SAVE = False

features_to_extract_is09 = []
```

```
features_to_extract_is10 = []

def get_statistical_data():
    with open('', newline='') as file:
        reader = csv.reader(file, delimiter = ',')
        next(reader)
        data = []
        labels = []
        for row in reader:
            if (len(row) > 0):
                label = int(row[-1])
                features = row[1:-1]
                data.append(features)
                labels.append(label)
    data = np.array(data)
    data = data.astype(float)
    return data, labels

def get_frame_wise_data(features):

    def get_selected_features_indexes(features, headers):
        indexes = []
        # print(headers)
        for feature in features:
            index = headers.index(feature)
            indexes.append(index)
        return indexes

    def get_features_from_row(row, selected_features_indexes):
        row_features = []
        for index in selected_features_indexes:
            row_features.append(row[index])
        return row_features

    def normalize_data(data, n_features):

        def get_biggest_size(data):
            biggest_size = 0
```

```
            for feature_set in data:
                feature_set_size = len(feature_set)
                if (feature_set_size > biggest_size):
                    biggest_size = feature_set_size
            return biggest_size

        biggest_size = get_biggest_size(data)

        for index, feature_set in enumerate(data):
            feature_set_size = len(feature_set)
            padding_size = biggest_size - feature_set_size
            padding_array = [np.array(np.zeros(n_features))]
            for _ in range(0, padding_size):
                if (PRE_PADDING):
                    feature_set =
                        np.insert(feature_set, 0, padding_array, axis=0)
                else:
                    feature_set =
                        np.append(feature_set, padding_array, axis=0)
            data[index] = feature_set

    data = []
    labels = []

    for root, dirs, files in os.walk(f""):
        for file in files:
            file_data = []
            label = 1 if '_str' in file else 0
            with open(f"", newline='') as file:
                reader = csv.reader(file, delimiter = ';')

                headers = next(reader)
                selected_features_indexes =
                    get_selected_features_indexes(features, headers)
                for row in reader:
                    if (len(row) > 0):
                        row_features =
                            get_features_from_row(
                                row, selected_features_indexes
```

```
                        )
                    row_features =
                        np.array(row_features, dtype=float)
                    file_data.append(row_features)
            data.append(file_data)
            labels.append(label)
    normalize_data(data, len(features))
    data = np.array(data)
    labels = np.array(labels, dtype=int)
    return data, labels

def one_hot_encode(labels):
    n_labels = len(labels)
    n_unique_labels = len(np.unique(labels))
    one_hot_encode = np.zeros((n_labels,n_unique_labels+1))
    one_hot_encode[np.arange(n_labels), labels] = 1
    one_hot_encode=np.delete(one_hot_encode, 0, axis=1)
    return one_hot_encode

def one_hot_encode_2(labels):
    encoded = to_categorical(labels)
    return encoded

def adapt_model(model, input_shape):
    adapted_model = keras.Sequential()

    for i, layer in enumerate(model.layers):
        layer_name = layer.name
        is_first_layer = (i == 0)

        if 'masking' in layer_name and is_first_layer:
            adapted_model.add(keras.layers.Masking(
                mask_value=layer.mask_value, input_shape=(input_shape)))

        elif 'lstm' in layer_name:
            adapted_model.add(keras.layers.LSTM(layer.units,
                return_sequences=layer.return_sequences,
                recurrent_dropout=layer.recurrent_dropout,
                activation=layer.activation.__name__))
```

```
            if 'dense' in layer_name:
                adapted_model.add(
                    keras.layers.Dense(
                        layer.units, activation=layer.activation.__name__)
                    )

        for i, layer in enumerate(adapted_model.layers):
            layer.set_weights(model.layers[i].get_weights())

        return adapted_model


def create_model(input_shape):
    model = keras.Sequential()

    model.add(tf.keras.layers.Masking(
        mask_value=0, input_shape=(input_shape)))

    for _ in range(0, LSTM_CELLS - 1):
        model.add(keras.layers.LSTM(
            UNITS_PER_CELL, return_sequences=True,
            recurrent_dropout=DROPOUT_RATE))

    model.add(keras.layers.LSTM(UNITS_PER_CELL,
        recurrent_dropout=DROPOUT_RATE))

    model.add(keras.layers.Dense(2, activation=ACTIVATION_FUNCTION))

    return model

def create_model_2(input_shape):
    model = keras.Sequential()

    model.add(tf.keras.layers.Masking(
        mask_value=0, input_shape=(input_shape)))

    for _ in range(0, LSTM_CELLS):
        model.add(keras.layers.LSTM(
```

```
            UNITS_PER_CELL, return_sequences=True,
            recurrent_dropout=DROPOUT_RATE))

    model.add(keras.layers.LSTM(2, activation=ACTIVATION_FUNCTION))

    return model

def prepare_datasets(data, labels, test_size, validation_size):
    X_train, X_test, y_train, y_test =
        train_test_split(data, labels, test_size=test_size)
    X_train, X_validation, y_train, y_validation =
        train_test_split(X_train, y_train, test_size=validation_size)

    return X_train, X_validation, X_test, y_train, y_validation, y_test

def plot_accuracy(history):

    fig, axs = plt.subplots(2)

    axs[0].plot(history.history["accuracy"], label=
        "Acuracia treinamento")
    axs[0].plot(history.history["val_accuracy"], label=
        "Acuracia validacao")
    axs[0].set_ylabel("Acuracia")
    axs[0].legend(loc="lower right")
    axs[0].set_title("Acuracia treinamento x Acuracia validacao")

    axs[1].plot(history.history["loss"], label="Loss treinamento")
    axs[1].plot(history.history["val_loss"], label="Loss validacao")
    axs[1].set_ylabel("Loss")
    axs[1].set_xlabel("Epoca")
    axs[1].legend(loc="upper right")
    axs[1].set_title("Loss treinamento x Loss validacao")

    plt.show()

def plot_loss(history):
    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
```

```
    plt.title('Loss treinamento vs Loss validacao')
    plt.ylabel('Loss')
    plt.xlabel('Epoca')
    plt.legend(['Treinamento', 'Validacao'], loc='upper right')
    plt.show()

class LossHistory(keras.callbacks.Callback):
    def on_train_begin(self, logs={}):
        self.losses = []

    def on_batch_end(self, batch, logs={}):
        self.losses.append(logs.get('loss'))

data, labels = get_frame_wise_data(
    features_to_extract_is09 + features_to_extract_is10)

labels = one_hot_encode_2(labels)

X_train, X_validation, X_test, y_train, y_validation, y_test =
    prepare_datasets(data, labels, 0.25, 0.2)

print(X_train.shape[0], X_train.shape[1], X_train.shape[2])

input_shape = (X_train.shape[1], X_train.shape[2])

if (model_path):
    model = load_model(model_path)
    model = adapt_model(model, input_shape)
else:
    model = create_model(input_shape)

optimizer = keras.optimizers.Adam(learning_rate=LEARNING_RATE)

model.compile(optimizer=optimizer,
    loss=LOSS_FUNCTION, metrics=['accuracy'])

loss_history = LossHistory()

model.summary()
```

```
es_callback = keras.callbacks.EarlyStopping(
    monitor='val_loss', patience=3)

history = model.fit(X_train, y_train, validation_data=
    (X_validation, y_validation),
    batch_size=BATCH_SIZE, epochs=EPOCHS_N)

plot_accuracy(history)
plot_loss(history)

if (SAVE):
    model.save('')

model.summary()

test_loss, test_acc = model.evaluate(X_test, y_test, verbose=2)
print('\nTest accuracy:', test_acc)
print('\nTest loss:', test_loss)
```

B.2 Extração

```
import os
import subprocess

for root, dirs, files in os.walk("./audios"):
    for dir in dirs:
        if (dir in ['stressed', 'neutral']):
            for root, dirs, files in os.walk(f"./audios/{dir}"):
                for file in files:
                    filename = file[0:file.index('.')]
                    if (dir == 'stressed'):
                        label = 2
                        filename += '_str'
                    else:
                        label = 1
                        filename += '_neu'
                    subprocess.run([
                        "SMILExtract", "-C", ".", "-I", f"", "-O",
                        "", "-instname", f"{file}", "-class",
```

```
        str(label), "-D", f""])
```