Universidade Federal do Espírito Santo

Centro Tecnológico

Programa de Pós-Graduação em Engenharia Elétrica

Guilherme Butzke Schreiber Gering

Identificação de sentimento em voz por meio da combinação de classificações intermediárias dos sinais em excitação, valência e quadrante

Vitória

2019

Guilherme Butzke Schreiber Gering

Identificação de sentimento em voz por meio da combinação de classificações intermediárias dos sinais em excitação, valência e quadrante

Universidade Federal do Espírito Santo

Centro Tecnológico

Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Evandro Ottoni Teatini Salles

Vitória

2019

Ficha catalográfica disponibilizada pelo Sistema Integrado de Bibliotecas - SIBI/UFES e elaborada pelo autor

---

Gering, Guilherme, 1994-
G369i Identificação de Sentimento em Voz por meio da Combinação de Classificações Intermediárias dos Sinais em Excitação, Valência e Quadrante / Guilherme Gering. - 2019.
101 f. : il.

Orientador: Evandro Salles.
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Espírito Santo, Centro Tecnológico.

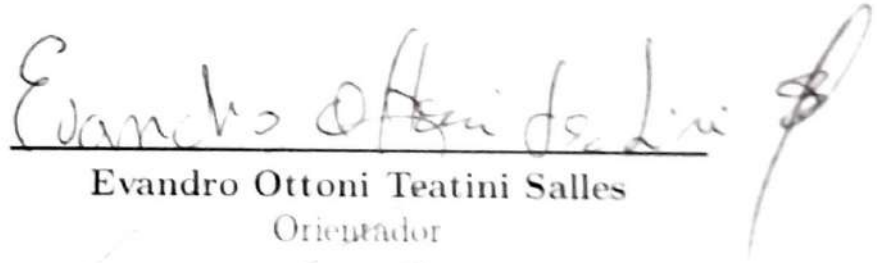
1. Aprendizado de máquinas. 2. Inteligência artificial - Aplicações médicas. 3. Emoções. 4. Reconhecimento automático da voz. I. Salles, Evandro. II. Universidade Federal do Espírito Santo. Centro Tecnológico. III. Título.

CDU: 621.3

Guilherme Butzke Schreiber Gering

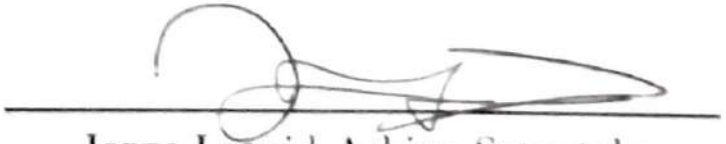
Identificação de sentimento em voz por meio da combinação de classificações intermediárias dos sinais em excitação, valência e quadrante

Trabalho aprovado. Vitória, 20 de dezembro de 2019:



Evandro Ottoni Teatini Salles
Orientador

Rodrigo Varejão Andreão
Membro Interno



Jorge Leonid Aching Samatelo
Membro Externo

Vitória
2019

Dedico este trabalho a todos os que fazem parte de mim

Agradecimentos

Agradeço muito à Deus, por ter me ouvido quando pedi para ter a oportunidade de estar no Mestrado, e que também me atendeu por tantas vezes quando eu precisei de muita força. Não há gratidão que corresponda a todos os seus feitos por mim.

Agradeço à minha família, que com toda a paciência e suporte diário, me incentivam e acreditam muito em mim, mesmo sem entender nada do que eu faço no meu trabalho.

Agradeço aos meus amigos da vida, eternos companheiros, que muito me ouviram reclamar e lamentar, e que sempre ficam felizes comigo a cada pequena conquista.

Agradeço aos meus colegas de laboratório por tantos ensinamentos, ajuda sempre solicita e apoio emocional. Foram o melhor presente que a Ufes me deu.

Agradeço ao Patrick, que esteve tão próximo a mim todo esse tempo, me tornando melhor como aluno e como pesquisador. Sei que não fui um orientando fácil, mas ele conseguiu me ensinar muito sobre o que eu mais gosto de aprender. E agradeço também ao Evandro, pelo apoio na fase final da pesquisa, e por todo o suporte de sempre como um bom coordenador.

Agradeço à Ufes, ao PPGEE, e também aos professores que me guiaram nas disciplinas do Mestrado. No momento em que termino este trabalho, a situação da Educação Nacional nos parece caótica. Devo agradecer por todos aqueles que acreditam no poder da Educação e que lutam pela manutenção e melhoria dos nossos direitos como alunos e professores.

Por fim, agradeço a todos os que me ouviram mais de perto, que sentiram comigo as minhas dificuldades particulares e, mesmo com um silêncio, um sorriso ou um abraço, me ajudaram nessa jornada tão desafiadora, mas tão incrível que foi o Mestrado. Vou lembrar sempre de todos os que estivaram comigo neste momento. Muito Obrigado!

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 1769586.

Tudo Passa.

Resumo

Reconhecimento de sentimento em voz é importante em áreas como saúde, psicologia e telemedicina para obter informações sobre os estados de emoções de um indivíduo. A identificação de sentimento em voz é comumente realizada em classes categóricas como "tristeza" ou "alegria". De acordo com o mapa de afeto de Russell, sentimentos também podem ser classificados por excitação, valência e quadrantes. Neste trabalho é proposto um método para incrementar o desempenho de identificação de sentimentos em classes categóricas utilizando classificadores que realizam classificação intermediária nas classes de excitação, valência e quadrantes usando uma abordagem Multi-tarefa. Além disso, três tipos de classificadores realizam a mesma tarefa de classificação, utilizando diferentes características extraídas do sinal da voz, que combinadas em um *Ensemble* tendem a incrementar os resultados individuais. Para combinar esses resultados e obter a classificação final é proposta uma árvore de decisão que aumentou o desempenho F1 de 0,61 do *Ensemble* de três tipos de classificadores para 0,63 sobre uma base de dados pública.

Palavras-chave: Identificação de Sentimento em Voz, Multi-Tarefa, Excitação-Valência

Abstract

Speech emotion recogntion is important in areas such as health, psychology, and telemedicine for information about an individual's states of emotions. Speech emotion recogntion is commonly performed in categorical classes, such as "sadness" or "joy". According to Russell's map of affection, emotions can also be classified by arousal (excitation), valence, and quadrants. In this work is proposed a method to increase the performance of speech emotion recogntion in categorical classes using classifiers that perform intermediate classification in the classes of valence, excitation and quadrants using a multiview approach. Moreover, three types of classifiers perform the same task, using different features extracted from the voice signal, which combine in one Ensemble they tend to increase individual results. To combine these results and obtain the final classification, a decision tree is proposed and that increases F1 metrics from 0.61 by Ensemble of three kinds of classifiers to 0.63 in a public database.

Keywords: *Speech Emotion Recognition, Multi-Task, Arousal-valence*

Lista de Ilustrações

Lista de Tabelas

Lista de Abreviaturas e Siglas

BN	*Batch Normalization*
CNN	*Convolutional Neural Network*
ELM	*Extreme Learning Machine*
ELU	*Exponential Linear Unit*
FN	*False Negatives*
FP	*False Positives*
GD	*Gradient Descendent*
GRU	*Gated Recurrent Unit*
KKT	Karush-Kuhn-Tucker
LPCC	*Linear Prediction Cepstral Coefficients*
LPMCC	*Linear Predictive Mel cepstrum coefficient*
LSTM	*Long-short Term Memory*
MFCC	*Mel Frequency Cepstral Coefficients*
MLP	*Multi-layer Perceptron*
MLT	*Multi-task Learning*
MSE	*Mean Squared Error*
ReLU	*Rectified Linear Unit*
RNN	*Recurrents Neural Networks*
SER	*Speech Emotion Recognition*
STL	*Single Task Learning*
SV	*Support Vectors*
SVM	*Support Vectors Machines*
TN	*True Negatives*
TP	*True Positives*
UFES	Universidade Federal do Espírito Santo

Sumário

1 Introdução

O complexo sinal de voz pode trazer várias informações a respeito da mensagem, do locutor, da linguagem e da emoção transmitida (LIVINGSTONE; RUSSO, 2018; GADHE et al., 2015; PATHAK; KOLHE, 2016). Humanos têm uma habilidade natural de reconhecer emoções através da fala. No campo da saúde, a identificação de sentimento em voz pode monitorar as condições de paciente em reabilitação ao, aconselhamento psicológico, identificação de autismo e também de pacientes com stress ou depressão (REDDY; VIJAYARAJAN, 2017). O estudo e o entendimento de emoções se aplica também quando se deseja conhecer o bem-estar de uma pessoa (seja um paciente, usuário, ou cliente) em determinado espaço.

O estudo sobre emoções ainda não possui uma universalidade na literatura. Um dos fatores, talvez, seja por este ser tema de interesse em campos multidisciplinares, o que envolve estudos neurológicos, fisiológicos, psicológicos e sociológicos (RODELLAR-BIARGE et al., 2015). Não existe um número certo de tipos de emoções que caracterizam os sentimentos humanos e, por vezes, também não há uma separação precisa entre dois sentimentos similares (como tristeza e depressão), conforme explicado em Wang, Nie e Lu (2014). A literatura indica que uma emoção expressa pode ser muito dependente do falante, da sua cultura e também do ambiente (AYADI; KAMEL; KARRAY, 2011).

Se entende por Identificação de Sentimentos em Voz (*Speech Emotion Recognition* - SER) o reconhecimento de sentimentos por máquinas. Máquinas, inclusive, podem identificar "quem disse" e o "que foi dito" na fala, além de poderem também identificar sentimentos expressos nas frases (GADHE et al., 2015).

SER são sistemas desenvolvidos para identificar sentimento em voz através de algoritmos computacionais. Na literatura é encontrada uma infinidade de métodos e técnicas que processam o sinal da voz de maneira a extrair informações para atribuir às classes de sentimentos. Pode-se entender, para este contexto, que um SER é um algoritmo de máquina que extrai características da voz de um falante e associa esta a um sentimento.

1.1 Categorização de sentimentos

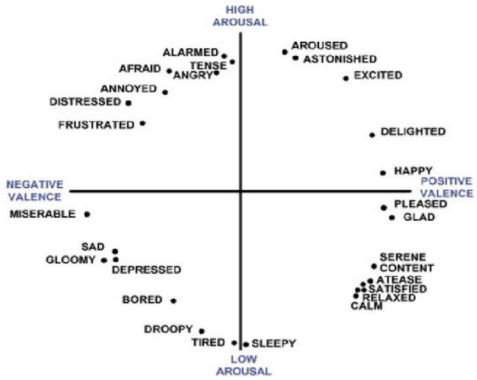
Duas teorias são amplamente utilizadas para classificação de sentimentos. A primeira associa cada sentimento a uma entidade discreta, separável, categorizada em tipos e quantidades (como raiva, medo, tristeza, alegria, etc.). Estas são chamadas de classes categóricas de sentimento (BESTELMEYER; KOTZ; BELIN, 2017).

A segunda teoria avalia cada emoção com um grau de excitação ou de valência

(*arousal and valence*), portanto, em um plano bidimensional. Essas classes são ditas contínuas, e as emoções são decompostas em excitação (ou ativação) ou valência em uma escala de valores (XIA; LIU, 2017). Em Parthasarathy e Busso (2017) também é utilizado dominância (ativa ou passiva) como uma classe contínua de sentimento.

A valência qualifica o sentimento quanto à simpatia, numa escala de sentimentos negativos (desagradáveis) até positivos (agradáveis). Em excitação (ou intensidade), quantificam-se os sentimentos quanto ao nível de ativação provocado pelo mesmo, em uma faixa que vai de baixo (calmo) até alto (excitado) (RUSSELL, 1980). Essa teoria é muito utilizada nos estudos referentes a emoção. A Figura 1 apresenta o modelo bidimensional de emoção descrito que é conhecido na literatura como modelo circumplexo de afeto de Russell (*Russell's Circumplex Model of Affect*) (RUSSELL, 1980), ou mapa de afeto.

Figura 1 – Modelo Bidimensional de emoções baseado circumplexo de afeto de Russell



Fonte: Russell (1980 apud WANG; NIE; LU, 2014)

1.2 Algoritmos para Identificação de Sentimentos em Fala

Cada algoritmo para identificação de sentimento em voz apresenta suas particularidades quanto ao tipo de informação que extrai do sinal de voz. A escolha da arquitetura do SER está relacionada com as características que se deseja extrair da voz, isso porque o desempenho dos algoritmos de máquina é sensível ao tipo e a natureza dos dados de entrada.

A relação entre a formação da fala e o estado emocional de um falante é identificada através de características que podem ser extraídas do sinal da fala. A voz é a principal forma de comunicação humana e de uma fala é possível extrair informações da pessoa como idade, gênero, dialeto, linguagem, stress e outros fatores. Todas essas características agregam informação ao ouvinte, mas a emoção aplicada a fala é capaz de alterar o comportamento desses parâmetros (GHARAVIAN et al., 2011). Alguns atributos que caracterizam a voz humana e os padrões de fala são pitch, pressão sonora (ou sonoridade), timbre e tom

(DASGUPTA, 2017). A análise do sinal de voz em curtos segmentos pode ser modelada como a saída de um filtro linear invariante no tempo excitado por um trem de pulsos quase periódico ou um sinal de ruído aleatório (RABINER; SCHAFER, 2011, pág. 417-418).

É comum em processamento de sinal dividir o sinal em segmentos menores chamados janelas (frames). Conforme citado por Ayadi, Kamel e Karray (2011) considera-se que o sinal de voz de características não estacionárias pode ser representado como estacionário em um período curto de tempo. Características prosódicas (que lidam com qualidades do som) da fala como duração, energia, pitch, frequência e taxa de ruído harmônico e coeficientes cepstrais de predição linear (*Linear Prediction Cepstral Coefficient - LPCC*) (REDDY; VIJAYARAJAN, 2017) são extraídas de cada frame e costumam ser chamadas de características locais, ou de descritores baixo nível. Em contrapartida, características do sinal extraídas por estatísticas ou descritos aplicados ao longo do conjunto de várias janelas seguintes de uma fala são denominadas de características globais ou descritores de alto nível. Ayadi, Kamel e Karray (2011) avaliam que o desempenho de classificação de algoritmos com características globais tem obtido melhores resultados na literatura.

Conforme Zhang, Liu e Weninger (2017), descritores de informação local são pitch, probabilidade de voz, energia, taxa de cruzamento de zero, entre outras; descritores de informação global são média, variância, mínimo, máximo, faixa, mediana, quartis e outros momentos (como suavidade e curtose) de elementos do sinal extraídos da fala. Além dessas, características relacionadas ao espectro do sinal da voz também são utilizadas como descritores da fala, como MFCC (*Mel Frequency Cepstral Coefficients*) e LPMCC (*Linear Prediction Mel Frequency Cepstral Coefficients*) são frequentemente utilizadas (SHEN; CHANGJUN; CHEN, 2011). O trabalho de Sailunaz et al. (2018) apresenta uma detalhada lista de características de fala e seus respectivos usos em modelos de classificadores. Os próximos parágrafos definem os conceitos de pitch e de coeficiente cepstrais.

O pitch da voz pode ser compreendido como a taxa de repetição de um tom ou de um fonema. A frequência associada a um pitch em determinado intervalo de tempo correspondente ao inverso do período de duração do fonema presente formado nesse intervalo (LYON; SHAMMA, 1996, pág.223). O pitch é independente da amplitude ou da intensidade do som (DASGUPTA, 2017). O sinal de pitch também é conhecido como forma de onda glotal pois contém informações a respeito da tensão contida nas cordas vocais e também da pressão de ar subglotal (VERVERIDIS; KOTROPOULOS, 2006). O sinal de pitch é produzido pela vibração da corda vocal e o período de duração do pitch corresponde ao tempo de duas aberturas sucessivas das cordas vocais. Os termos frequência fundamental de fonação ou frequência de pitch estão associados a taxa de vibração das cordas vocais (VERVERIDIS; KOTROPOULOS, 2006).

Rabiner e Schafer (2011, pág.399) mostraram que o logaritmo do espectro de um sinal de voz com eco consiste do logaritmos do sinal do espectro mais o logaritmo da

componente periódica devido ao eco. Em suas análises eles sugeriram que o logaritmo do espectro de um sinal poderia evidenciar o componente periódico e também poderia indicar a ocorrência de um eco. Dessa abordagem é que surge o termo cepstrum que define o espectro de potência do logaritmo do espectro de potência de um sinal. O termo quefrência indica a variável independente do cepstrum do sinal e pode ser entendida como frequência em termos de cepstrum (RABINER; SCHAFER, 2011, pág.400).

O conjunto de coeficientes cepstrais possuem importantes informações sobre o espectro do sinal de fala e tem sido amplamente utilizado em muitos sistemas de processamento de fala (RABINER; SCHAFER, 2011, pág.465). A característica do sinal de voz é ter transições relativamente suaves, e isso se espelha nas mudanças dos valores dos coeficientes ao longo do tempo. Um exemplo que Rabiner e Schafer (2011, pág.400) apresentam para aplicabilidade do cepstrum é que uma componente cepstral de baixa quefrência, ou seja, uma frequência no cepstrum que apresentam alta energia representa uma mudança mais suave no espectro do sinal enquanto que uma componente cepstral de alta quefrência indica uma variação mais abrupta da potência do espectro. Essa característica foi posteriormente utilizada para identificação do período do pitch e da fonação da fala. De tal maneira, as mesmas características podem ser utilizadas como descritores da fala para algoritmos de identificação de sentimento.

Um sistema reconhecedor de sentimento em fala pode ser compreendido como aquele capaz de extrair informações de voz e destas características pressupor a emoção do falante. Os objetivos principais de um SER são identificar os sentimentos presentes em uma fala e sintetizar a mensagem desejada de acordo com uma mensagem pretendida (PATHAK; KOLHE, 2016). Algoritmos de aprendizado de máquina em SER podem realizar tanto tarefas supervisionadas de classificação e regressão como também tarefas não-supervisionadas (BESTELMEYER; KOTZ; BELIN, 2017). O presente trabalho tem a proposta de identificar sentimentos categóricos em voz por meio de bases de dados que são rotuladas. Portanto, aqui se destaca as técnicas baseadas em classificadores para identificação de sentimento em voz.

Em Reddy e Vijayarajan (2017) é afirmado que diferentes classificadores podem ser utilizados em aplicações com sinal de voz, como o Modelo de Mistura de Gaussianas, Cadeias de Markov, Redes Neurais Artificiais, Máquinas de Vetores de Suporte (*Support Vector Machine* – SVM) e Redes Neurais Profundas. Máquinas de Vetores de Suporte têm encontrado resultados muito interessantes na identificação de sentimento em voz. Em Shen, Changjun e Chen (2011) é descrita uma abordagem em que o sinal da voz é representado pelas características prosódicas de energia, pitch, LPCC (*Linear Prediction Cepstral Coefficient*), e também pelas características espectrais MFCC (*Mel Frequency Cepstral Coefficients*) e LPMCC (*Linear Prediction Mel Frequency Cepstral Coefficients*). Os treinos são realizados com as características individuais e também combinadas, e nos

experimentos eles concluem que a combinação das características de Energia, Pitch e LPMCC alcançam melhores resultados. Os testes foram realizados sobre cinco sentimentos da base de dados Berlin e a acurácia foi em torno de 82.5%.

Em SER, redes neurais convolucionais (*Convolutional Neural Network* – CNN) são aplicadas para extrair características do sinal de voz temporal (sinal unidimensional), ou por vezes também de uma representação do espectrograma do sinal da fala, como pode ser visto em Zhao, Mao e Chen (2019) e (BADSHAH et al., 2017). As CNN têm sido utilizadas para a identificação de sentimentos com resultados bem promissores, inclusive melhorando os bons resultados de SVM Zhang et al. (2018). Redes convolucionais podem operar sob sinais que são unidimensionais ou mesmo multidimensionais. Neste trabalho são construído arquiteturas que são uni e bidimensionais, de maneira que são referidas como CNN-1D e CNN-2D, respectivamente.

Arquiteturas LSTM (*Long-Short Term Memory*) são utilizadas para classificação de sinais cujo estado atual tem alta dependência de estados passados por meio de funções com capacidade de armazenar informações relevantes a longo prazo bem como também esquecer informações mais irrelevantes. Em SER, eles tem sido muito utilizadas conectadas as camadas de CNN, como pode ser visto em Zhao, Mao e Chen (2019) e Fayek, Lech e Cavedon (2017). A característica das LSTM de armazenar informações a longo prazo se torna poderosa para identificação de sentimentos, uma vez que a característica do sentimento na fala é predominante a longo prazo. O trabalho de Fayek, Lech e Cavedon (2017) apresenta o resultado para várias arquiteturas de CNN construídas, incluindo aquelas que apresentam camadas de LSTM. O trabalho apresenta arquiteturas CNN que são combinadas com camadas LSTM, de maneira que estas são referenciadas como CNN-LSTM-1D e CNN-LSTM-2D, para o caso uni e bidimensional, respectivamente. A seção apresenta com mais detalhes a definição das topologias de CNN associadas a LSTM.

1.3 Análise Multi-Características para identificação de sentimentos em voz

Diferentes arquiteturas de SER podem extrair diferentes tipos de características do sinal de voz para a mesma finalidade de predição de sentimento. Trabalhos recentes, como os de Zhang et al. (2018) e Shih, Chen e Wu (2017) apontam que a combinação de classificadores em paralelo ajuda no desempenho da identificação de sentimentos. Combinar arquiteturas é interessante pois classificadores distintos podem utilizar (ou encontrar) diferentes informações do sentimento na voz, de maneira que a classificação combinada dos algoritmos utilize de maior informação do sinal de voz, podendo, então, aumentar o desempenho global da classificação se comparada ao desempenho individual de cada classificador. Nesta combinação de resultados, espera-se que a classificação tenha melhor

desempenho pois o sinal de voz é visto sob o "ponto de vista" de diferentes arquiteturas.

Neste trabalho, a inspiração para desenvolver algoritmos combinados de arquiteturas se encontra no propósito de utilizar diferentes características do sinal de voz, como suas características estatísticas (máximo, mínimo, maior incremento e decremento, entre outras) de espectro e cepstro, bem como das características extraídas do espectrograma do sinal de voz, ou mesmo de suas características temporais. Ou seja, múltiplas características do sinal são encontradas, por diferentes classificadores, para que, quando combinadas, apresentem uma classificação de sentimentos com base em informação obtida. Esta abordagem é definida como abordagem Multi-Características.

Tuarob et al. (2014) utiliza as expressões "diferentes visões" e "multi-aspectos" para explicar uma metodologia que combina cinco classificadores heterogêneos para classificação de emoção em texto. Os autores afirmam que cada um dos classificadores reflete uma visão da base de dados utilizada: enquanto um classificador encontra os padrões das palavras, outros capturam diferentes níveis de semântica no áudio. Os resultados mostram que a combinação dos diferentes classificadores aumenta o desempenho global de classificação. Os autores justificam que classificadores treinados sob diferentes visões dos dados podem recompensar erros de outros. Em Zhang et al. (2018), por exemplo, uma arquitetura é proposta para reconhecimento de emoção em voz que combina redes convolucionais unidimensional para classificar o sinal de voz no tempo e redes convolucionais bidimensionais para extrair informações do espectro.

Na pesquisa bibliográfica realizada não foi encontrada universalidade para o assunto. Termos como "análise multi-aspecto", "analise multi-visão" podem ser utilizadas para o mesma finalidade. Os trabalhos de Soleymani et al. (2012) e Vryzas et al. (2018) analisam bases de dados afetiva, e destas eles utilizam algoritmos para extrair informações do vídeo, do texto e da voz. Soleymani et al. (2012) utilizam, além das anteriores, informações de eletrocardiograma, amplitude de respiração e temperatura da pele, entre outras, para classificação de sentimentos. Os autores descrevem essa técnica como "análise multi-modal" (*multi-modal*) de sentimento, pois vários meios são utilizados para obter informações sobre o mesmo sentimento. O presente texto não trabalha com essa abordagem uma vez que seu corpo de trabalho se restringe a identificação de sentimento apenas pelo canal de voz, diferente dos autores que utilizam sinais de vídeo, biológico, entre outros. Contudo, o termo classificação "multi-modal" muitas vezes é encontrada quando se busca palavras chaves associadas ao termo multi-características acima citado e por isso cabe aqui ser diferenciado.

A escolha dos algoritmos neste trabalho foi pensada na perspectiva da análise Multi-Características do sinal de voz. Isso porque cada algoritmo categoriza os dados sob um diferente "ponto de vista": enquanto SVM utiliza estatísticas extraídas do sinal temporal para classificar os sentimentos na fala, o modelo CNN-LSTM-1D extrai infor-

mações de curto e longo-tempo do sinal por meio das camadas recorrentes e por meio de operações convolucionais e o modelo CNN-LSTM-2D obtém informações a respeito de características temporais-espectrais. Combinar os resultados desses diferentes classificadores foi interessante pois a identificação da emoção aconteceu por vários pontos de vista: por características estatísticas devido ao SVM, por características temporais através da CNN-1D e por características temporais-espectrais devido a CNN-2D.

Várias são as técnicas na literatura utilizadas para combinar arquiteturas de classificadores, e então assim realizar a análise Multi-Características. Técnicas de *Ensemble* (em português comitê) utilizam o treino paralelo de mais de um classificador nos quais as saídas dos modelos são combinadas de forma a garantir uma predição final baseada nos resultados individuais de cada arquitetura (SHIH; CHEN; WU, 2017).

O trabalho de Zhang et al. (2018), por exemplo, possui um esquema de *Ensemble* constituído de quatro classificadores e a predição de cada um desses modelos é combinada por um esquema de votação para estabelecer-se uma predição final de classificação.

1.4 Análise Multi-Tarefa para identificação de sentimentos em voz

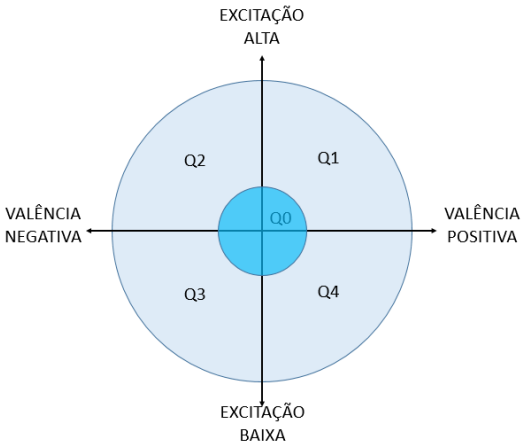
O sentimento classificado por um SER pode ser tanto uma entidade discreta e separável, como raiva, tristeza, alegria, como também um sentimento classificado em excitação ou valência, conforme teoria de Russel (RUSSELL, 1980).

Olhando para o mapa de afeto de Russel (Figura 1), verifica-se que todo sentimento categórico pode ser expresso por um nível de excitação (alta, baixa ou neutra), valência (positiva, negativa, ou neutra) e também pela posição do quadrante no mapa. Um exemplo disso é o sentimento de raiva, que na maioria das línguas é considerado um sentimento de alta excitação e de valência negativa. A Figura 2 ilustra o mapa de afeto, indicando os quadrantes de sentimentos, associado com suas dimensões de excitação e valência. Em suma pode-se dizer que entidades categóricas podem ser representadas em outro espaço dimensional representado pelas grandezas de excitação, valência e quadrante.

Um sistema identificador de sentimentos pode ser treinado para classificar os sentimentos quanto a rótulos discretos, mas também em rótulos de excitação, de valência, e até mesmo de quadrantes. Os trabalhos de Zhang (2018) e Albadawy e Kim (2018) abordam esse procedimento em suas técnicas.

Baseado nestas observações, a hipótese desse trabalho é que se é possível alcançar melhores resultados de identificação de sentimentos ao combinar os resultados de diferentes arquiteturas que realizam a identificação de sentimentos nas subclassificações de excitação, valência e quadrantes. Porém, uma questão que surge é como combinar esses resultados para expressar uma entidade de sentimento.

Figura 2 – Mapa multi-dimensional de sentimentos: excitação, valência e quadrantes



Fonte: Adaptado de Russell (1980 apud WANG; NIE; LU, 2014)

A técnica encontrada para auxiliar nessa questão é encontrada na literatura como aprendizado Multitarefa (*Multi-task Learning* - MLT). Um dos primeiros trabalhos abordando o tema de Multi-Tarefa para classificação é apresentado por Thrun (1997). A ideia principal da metodologia é resolver grandes problemas de aprendizado de máquina quebrando o processo em tarefas menores, geralmente independentes, mas que podem ser combinadas. O artigo defende que muitos problemas do mundo real podem ser resolvidos com auxílio de informações contidas no treino de sinais de outras tarefas derivadas do mesmo domínio.

> "Uma rede neural (ou árvore de decisão, ou ...) treinada em uma tabula rasa em uma única, isolada tarefa complicada é improvável de ser aprendida bem. Por exemplo, uma rede com 1000x1000 pixels de entrada de uma [imagem] de retina é improvável de aprender cenas complexas do mundo real dado o número de padrões e tempo de treino e tempo para ser avaliada. [...] Se simultaneamente essa rede fosse treinada com para reconhecer contornos de objetos, formas, regiões, sub-regiões, distâncias... isso faria aprender melhor a reconhecer objetos no mundo real. Essa abordagem é aprendizado Multi-Tarefa". (THRUN, 1997, tradução pelo autor).

Thrun (1997) ainda explica que na abordagem de aprendizado de tarefa Única (*Single Task Learning - STL*) cada tarefa é uma função das mesmas entrada mas cada uma com saída distinta e com cada treino é realizado isoladamente. Em sentimento em voz, exemplos de STL são redes de mesmo estrutura, com uma treinando o sentimento para rótulos de valência e outra para de excitação. Num aprendizado Multi-Tarefa, uma mesma arquitetura é treinada para todas as tarefas disponíveis, de maneira que o resultado individual de cada uma pode ser cominado. Em Redes Neurais, inclusive, uma arquitetura pode ser treinada tendo duas camadas de saídas distintas e compartilhadas, como nos trabalhos de Zhang, Liu e Weninger (2017) e Xia e Liu (2017).

Para SER em voz, Xia e Liu (2017) apresentam uma abordagem Multi-Tarefa para identificar quatro tipos de sentimento de uma base de voz. Na metodologia aplicada, a tarefa principal consiste em classificar a voz em sentimentos categóricos (raiva, felicidade, tristeza e neutro). Duas tarefas foram utilizadas para auxiliar a tarefa principal: a primeira foi classificar a voz em excitação (alto, neutro, baixo) e a segunda em valência (positiva, neutra, negativa). O autor ainda propõe uma segunda metodologia que ao invés de utilizar classificadores para identificar as classes secundárias, utilizou de tarefas de regressão linear para estimar a excitação e a valência do sinal numa escala que vai de -1 a 1. Os autores utilizaram de redes de crença profunda (um tipo especial de redes neurais profunda), onde a última camada escondida compartilha informações das tarefas principais e das tarefas secundáras. Os experimentos mostraram que, considerando a informação de excitação e de valência, o desempenho de emoções categóricas é maior se comparada com sistemas tradicionais usando classes contínuas.

De acordo com Xia e Liu (2017), os objetivos das tarefas secundárias é auxiliar no aprendizado das tarefas principais. Na abordagem Multi-Tarefa deste trabalho, a tarefa principal é processo de classificar um sentimento em rótulos categóricos enquanto tarefas secundárias são classificações que rotulam o sentimento de entrada em excitação, valência e quadrantes. Vale observar que cada arquitetura pode ser utilizada para realizar tarefas principais, ou secundárias. Como exemplo, na pesquisa foi construído um SVM que rotulou os sentimentos categóricos, bem como se construiu outras três arquiteturas similares que rotularam os sentimentos em excitação, valência e quadrantes. O mesmo foi feito para os demais classificadores, CNN-LSTM-1D e CNN-LSTM-2D, e o *Ensemble* entre as três.

A finalidade das tarefas secundárias é realizar classificações intermediárias de sentimentos em excitação, valência e quadrantes para auxiliar na tarefa principal que é a classificação de sentimentos categóricos. Para combinar os resultados das tarefas secundárias, foi utilizada uma árvore de sequência de decisão binária. O principal objetivo da árvore é criar uma sequência de classificadores, baseada nos resultados dos modelos intermediários construídos, para que, por fim, as falas sejam classificadas em sentimentos categóricos.

Escolheu-se Árvores de Decisão Hierárquicas como algoritmo para combinar os resultados das tarefas secundárias pois estas têm sido utilizadas em SER por voz para classificar sentimentos através de múltiplos classificadores. Em Lee et al. (2011) é utilizada uma classificação de sentimentos baseadas em árvores de decisão binárias hierárquicas onde cada folha de decisão da árvore identifica individualmente uma classe de sentimentos por meio de classificadores SVM ou por Regressão Logística Bayesiana. Liu et al. (2017) e Mao (2018) utilizam classificadores SVM e ELM (*Extreme Learning Machine*) em seu modelo hierárquico. Ambos trabalhos treinam os modelos por meio de classificação um-contra-todos, de maneira que cada nível da árvore separa um sentimento dos demais. Mais

detalhes sobre o uso de árvores hierárquicas de decisão para identificação de sentimento em voz são abordados na Seção 2.4, enquanto que na Seção 3.3 é explicada a metodologia construída para combinar os resultados de classificações intermediárias. Por hora, interessa-se saber que a finalidade da árvore decisão neste trabalho é encontrar a sequência de tarefas principais e secundárias que melhor classifique os sentimentos em rótulos discretos, realizando assim o método que chamamos de Multi-Tarefa.

1.5 Método Proposto

A inspiração para o método proposto neste trabalho foi baseada em análises experimentais realizadas pelo autor na tarefa de classificação de voz em sentimentos principais. Foi observado experimentalmente que, se um problema complexo fosse dividido em problemas mais simples, e se as soluções destes problemas simples fossem utilizadas, o problema mais complexo poderia ser resolvido com resultados mais adequados. Além disso, se fossem extraídas diferentes informações dos mesmos dados, haveria uma possibilidade maior de construir modelos melhores. Foi aí que chegou-se na combinação de duas abordagens: a análise Multi-Tarefa, que divide uma tarefa principal em vários problemas secundários, e na abordagem Multi-Características, que extrai vários tipos de características de um mesmo conjunto de dados.

A ideia por trás da proposta é que se forem construídos e combinados vários classificadores, cada qual destinado a resolver um problema simples (problema secundário) usando conjuntos diferentes de características (visão) extraídos dos dados, o problema principal poderia ser resolvido mais facilmente. Isto é esperado, pois o problema teria sido dividido em problemas de menor complexidade e os classificadores teriam diferentes visões do mesmo conjunto de dados, facilitando a resolução dos problemas mais simples.

Dito isso, a principal contribuição teórica deste trabalho é a proposta de uma técnica de classificação de sentimentos que rotulam o sinal de voz sobre diferentes aspectos de emoção, além de utilizar vários tipos de características extraídas da fala, em duas abordagens principais que o autor define como abordagem Multi-Características e Multi-Tarefa, respectivamente.

A abordagem proposta é Multi-Tarefa pois ela realiza a classificação em rótulos secundários (excitação, valência e quadrantes) para usar como informação a priori na classificação de rótulos principais (raiva, medo, alegria, etc.). Por sua vez, a abordagem também é Multi-Características, pois ela utiliza diferentes características do sinal da voz (estatísticas, cepstrais, temporais) para treinar distintos classificadores (SVM, CNN-1D, CNN-2D).

Pode-se resumir a abordagem proposta como segue. Em uma primeira etapa, um conjunto de diferentes tipos de classificadores são treinados, cada um com características

distintas extraídas do sinal de voz, para rotular os sentimentos em rótulos principais e secundários. Importante destacar que as características não são todas combinadas em um único classificador, por exemplo, uma SVM é treinada com características estatísticas e uma CNN com características Cepstrais, de maneira que cada classificador trabalha com as características que são mais pertinentes de serem utilizadas em sua arquitetura. Em uma segunda etapa, o uso de uma árvore de decisão hierárquica combina os resultados destes diferentes classificadores. A árvore é tem a finalidade de combinar sentimentos de rótulos diferentes, como excitação e valência, para predizer o sentimento categórico referente a voz. Após essas duas etapas, pode-se dizer que a classificação geral dos sentimentos utilizou-se de distintas informações do sinal de voz (através da abrodagem Multi-Características) além de diferentes informações do sentimento pelo mapa de afeto (através da abordagem Multi-Tarefas), para predizer sentimentos categóricos presentes na fala.

Ambas abordagens têm em comum o fato de combinarem entidades para aperfeiçoar o algoritmo de classificação: enquanto a abordagem Multi-Características seleciona diferentes descritores do sinal de entrada para serem combinadas pelo *Ensemble* de arquiteturas, a abordagem Multi-Tarefa seleciona distintas representações do espaço dimensional de sentimentos (excitação, valência e quadrantes) para então auxiliar na predição de um sentimento discreto. Entende-se, em certo nível de abstração, que a abordagem Multi-Características olha para as múltiplas características de entrada dos modelos e as combina, enquanto a abordagem Multi-Tarefa olha para os tipos de maneiras possíveis de representar os sentimentos no mapa de afeto e procura combiná-las para melhor predizer os sentimentos discretos.

1.6 Estrutura do trabalho

Este trabalho está organizado como segue. O Capítulo 2 detalha sobre os algoritmos e classificadores utilizados neste trabalho, explicando um pouco da teoria e das técnicas dos algoritmos de aprendizado de máquina utilizadas para classificação de sentimentos deste trabalho. No Capítulo 3 é apresentada a metodologia proposta para a construção do SER utilizando as abordagens Multi-Características e Multi-Tarefa. Os experimentos, análises e resultados do método aplicado são realizadas no Capítulo 4. Por fim, as conclusões obtidas a partir dos resultados e uma análise de como a metodologia proposta contribuiu no aperfeiçoamento do desempenho da identificação de sentimentos em voz é encontrada no Capítulo 5.

2 Algoritmos de Classificação para Identificação de Sentimento em voz

O presente capítulo tem como finalidade discorrer a respeito dos principais algoritmos de aprendizado de máquina utilizados na pesquisa para predição de sentimentos através de característica extraídas do sinal da voz. Os três principais algoritmos foram Máquinas de Vetores Suporte, Redes Convolucionais uni e bidimensionais, técnicas de agrupamento de resultados por meio de *Ensemble*, além de árvores binárias de decisão hierárquica. Cada seção apresenta uma breve introdução matemática à formulação desses algoritmos bem como a justificativa de de sua utilização neste trabalho. A última seção do capítulo apresenta e explica as métricas utilizadas para medida de desempenho dos resultados de classificadores.

2.1 Máquinas de Vetores Suporte - SVM

Máquina de Vetores Suporte, ou do inglês, *Support Vectors Machines - SVM*, têm encontrado resultados interessantes em processos de classificação de sentimentos em voz. Conforme o *survey* de (REDDY; VIJAYARAJAN, 2017), algoritmos SVM conseguem taxas de assertividade entre 75 a 80% e seu uso é muito difundido devido a sua eficiência e simplicidade, o que garante bons resultados de classificação.

O trabalho de (SHEN; CHANGJUN; CHEN, 2011), que inspirou um dos métodos de classificação aplicados nesta pesquisa, utiliza características extraídas de trechos de fala para classificar sentimentos por meio de uma arquitetura SVM e conseguem uma assertividade de 82.5% dos sentimentos. Os autores utilizam como entrada um vetor de características estatísticas (máximo, mínimo, média, taxa de subida e descida, entre outras) extraídas sobre a energia, *pitch*, coeficientes LPCC (Coeficientes de Predição Linear Cepstral - *Linear Prediction Cepstral Coefficients*)) além dos coeficientes espectrais de MFCC (Coeficientes Mel-Cepstrais de Frequência - *Mel Frequency Cepstral Coefficients*) e LPMCC (Coeficientes Preditivos Lineares Mel-Cepstrais - *Linear Predictive Mel cepstrum coefficient*) do sinal de voz. A concatenação dessas características totaliza um vetor de 52 dimensões, que é posteriormente classificado pelo SVM. A descrição das características, o pré-processamento do sinal e o detalhamento da arquitetura do SVM são apresentados em (SHEN; CHANGJUN; CHEN, 2011).

2.1.1 SVM com margens rígidas

Em um problema de classificação, máquina de Vetores Suporte é uma abordagem que trabalha com problema de maximização da distância entre as fronteiras de separação de duas classes. Fronteiras de separação são hiperplanos que separam os dados em classes distintas. Para dados linearmente separáveis, existem infinitas soluções para o problema. O objetivo do algoritmo SVM é encontrar o hiperplano que minimize o erro de generalização entre as amostras. Isso corresponde a solucionar um problema de maximização das margens entre as amostras, de maneira que a distância entre os hiperplanos de separação entre as classes seja a maior possível.

Considere a Figura 3 como base de explicação. No esquema, x_1 e x_2 representam os Vetores Suporte, H_1 e H_1 a equação dos hiperplanos e d a distância entre as margens. Seja um conjunto de treino $\mathbf{x}$, possuindo N amostras, cada uma associada a uma função objetivo definida pelo espaço de duas classes $Y = \{-1, 1\}$. Supondo os dados de $\mathbf{x}$ linearmente separáveis, pode-se dizer que existe uma infinidade de hiperplanos que separam os dados das duas classes. Neste conjunto de dados, as fronteiras de separação são $H1$ e $H2$ e são apresentadas pelas Equações 2.1 e 2.2, sendo $\mathbf{w}$ normal às fronteiras e b é um coeficiente de ordem zero. Todos os pontos podem ser escritos na forma que seguem a Equação 2.4, ou de uma maneira explícita pelas inequações apresentadas em 2.3. (LORENA et al., 2007)

$$H_1 : \mathbf{w} \cdot \mathbf{x} + b = 1 \tag{2.1}$$

$$H_2 : \mathbf{w} \cdot \mathbf{x} + b = -1 \tag{2.2}$$

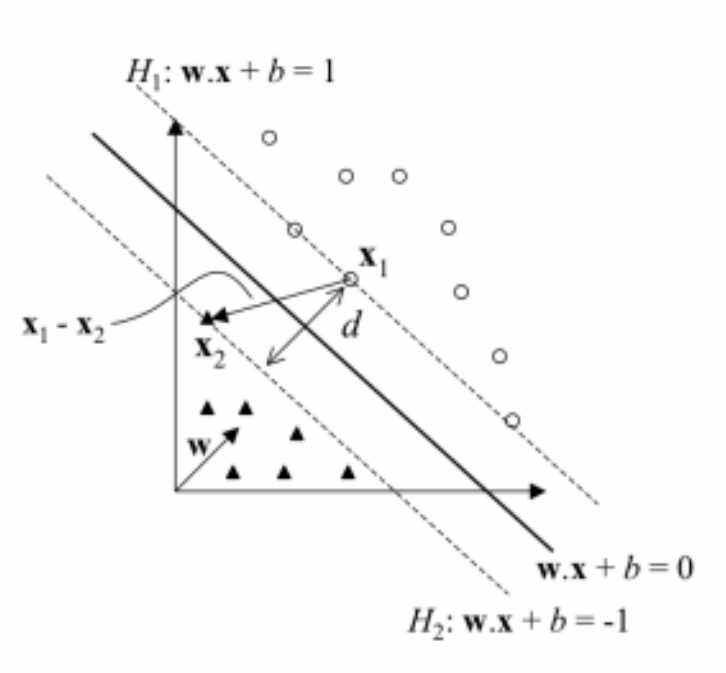
$$\begin{cases} \mathbf{w} \cdot \mathbf{x}_i + b \geqslant +1 & \text{se } y_i = +1 \\ \mathbf{w} \cdot \mathbf{x}_i + b \leqslant -1 & \text{se } y_i = -1 \end{cases} \tag{2.3}$$

$$y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1 \geqslant 0, \quad \forall \left(\mathbf{x}_i, y_i\right) \in T \tag{2.4}$$

Considera-se $\|d\| = 2/\|\mathbf{w}\|$ o módulo da distância entre os hiperplanos H_1 e H_2 calculado a partir do vetor normal(LORENA et al., 2007). O problema do SVM consiste na maximização dessa distância de margem, que pode ser visto também como um problema de minimização quadrática do inverso da distância. Dessa forma, tem-se o seguinte problema de minimização Primal com as seguintes restrições:

$$\begin{gathered} \text{Minimizar } \tfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{Com restricões: } y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1 \geqslant 0, \quad \forall i = 1, \ldots, n \end{gathered} \tag{2.5}$$

Figura 3 – Exemplo esquema de classificação SVM



Fonte: Lorena et al. (2007)

As restrições são impostas para que não haja dados de treino na região entre fronteiras. Por isso, essa nomenclatura é chamada de SVM com margens rígidas (BISHOP, 2006; LORENA et al., 2007). Problemas de otimização que apresentam restrições podem ser resolvidos com multiplicadores de Lagrange, que são associados a parâmetros $\boldsymbol{\alpha}$ e podem ser expressos por:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1\right) \tag{2.6}$$

Basicamente, a função Lagrange encontra os valores de $\boldsymbol{\alpha}$ que minimizam a função custo L, determinando o conjunto de pontos que maximizam a distância entre as margens. Minimizar L implica maximizar α_i e minimizar $\boldsymbol{w}$ e b. Calculando os pontos de sela da função, encontram-se os resultados expressos em 2.7 e 2.8. Substituindo as expressões em 2.5, leva-se a formulação de um problema dual, apresentado em 2.9.

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.7}$$

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{2.8}$$

$$\begin{aligned} &\underset{\alpha}{\text{Maximizar}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) \\ &\text{Com as restrições} \begin{cases} \alpha_i \geqslant 0, \quad \forall i = 1, \ldots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \end{aligned} \tag{2.9}$$

O problema Dual é otimizado encontrando um vetor $\boldsymbol{\alpha}^*$ que maximiza a expressão em 2.9, enquanto que a solução do problema Primal determina um vetor $\mathbf{w}^*$ e b^* que minimizam L. Esses valores podem ser obtidos por meio das uma das três condições de Karush-Kuhn-Tucker (KKT), provenientes da teoria de otimização (BISHOP, 2006, pág.330). As três condições aplicadas estão representadas nas Equações 2.10, 2.11 e 2.12:

$$\alpha_i^* \geqslant 0, \forall i = 1, \ldots, n \tag{2.10}$$

$$y_i \left(\mathbf{w}^* \cdot \mathbf{x}_i + b^*\right) - 1 \geqslant 0, \forall i = 1, \ldots, n \tag{2.11}$$

$$\alpha_i^* \left(y_i \left(\mathbf{w}^* \cdot \mathbf{x}_i + b^*\right) - 1\right) = 0, \forall i = 1, \ldots, n \tag{2.12}$$

Encontrado $\boldsymbol{\alpha}^*$, determina-se $\mathbf{w}^*$ substituindo-o na Equação 2.8. O valor de b^* é obtido aplicando $\mathbf{w}^*$ e $\boldsymbol{\alpha}^*$ na Equação 2.12. A solução do problema apresenta valores de α_i^* diferente de 0 somente para os dados que se encontram sobre os hiperplanos $H1$ e $H2$. Estes são os exemplos que se situam exatamente sobre as margens e são os únicos pontos que participam do cálculo de $\mathbf{w}^*$, sendo portanto os únicos necessários para a definição da região de fronteira e, por isso, são denominados Vetores Suporte (SVs, do Inglês *Support Vectors*)(BISHOP, 2006, pág.330). Para encontrar a equação dos hiperplanos, utiliza-se da média dos pontos b^* associados aos Vetores Suporte, conforme segue na Equação 2.13, onde n_{SV} é a quantidade de vetores suporte, x_j são vetores no conjunto de vetores suporte n_{SV} e y_j são as classes associadas a cada um desses vetores.

$$b^* = \frac{1}{n_{\text{SV}}} \sum_{\mathbf{x}_j \in \text{SV}} \frac{1}{y_j} - \mathbf{w}^* \cdot \mathbf{x}_j \tag{2.13}$$

A solução final é um classificador binário $g(x)$ que pode ser indicada conforme a Equação 2.14. O nome do algoritmo SVM vem justamente dos pontos que estão contidos na margem dos hiperplanos de separação das classes. Esses pontos são importantes pois eles determinam a equação do hiperplano que maximiza a região de fronteira entre as classes.

$$g(\mathbf{x}) = \text{sgn}\left(\sum_{\mathbf{x}_i \in \text{SV}} y_i \alpha_i^* \mathbf{x}_i \cdot \mathbf{x} + b^*\right) \tag{2.14}$$

$$g(\mathbf{x}) = \text{sgn}\left(\sum_{\mathbf{x}_i \in \text{SV}} y_i \alpha_i^* \mathbf{x}_i \cdot \mathbf{x} + b^*\right)$$
$$\text{Para } sgn(x) = \begin{cases} +1 & se \quad x \geq 0 \\ -1 & se \quad x < 0 \end{cases} \tag{2.15}$$

2.1.2 SVM com margens flexíveis

Alguns conjuntos de treinamento podem não ser linearmente separáveis. A presença de ruídos e *outliers* pode formar uma superfície de separação dos dados que não é linear (LORENA et al., 2007). O algoritmo de SVM dito com margens suaves ou flexíveis aumenta as margens de separação na classificação binária de maneira que elementos dentro da faixa de erro são consideradas acerto. Isso previne que dados ruidosos ou discrepantes atrapalhem a generalização do modelo (BISHOP, 2006, pág.332).

Para relaxar as variáveis no processo de classificação, um erro ξ_i é inserido de maneira que a Equação 2.5 do problema Primal se adapte a essa condição, conforme a Equação 2.16 apresenta.

$$y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \geqslant 1 - \xi_i, \quad \xi_i \geqslant 0, \forall i = 1, \ldots, n \tag{2.16}$$

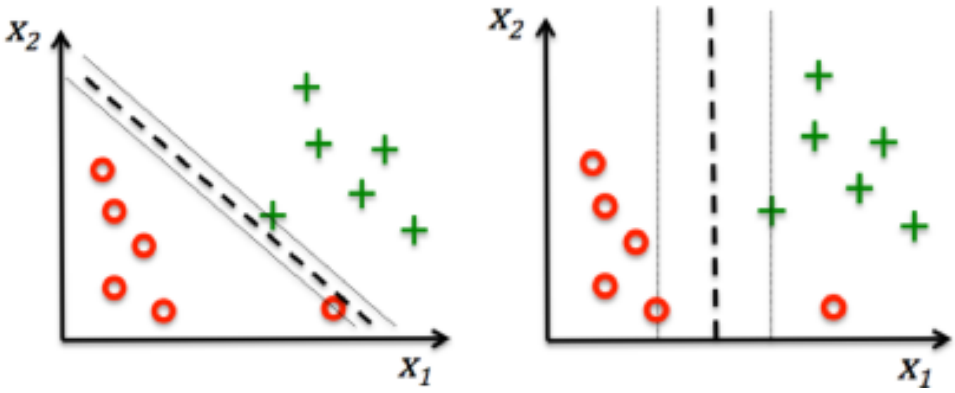
Nessa Equação, ξ_i é o erro restrito a cada variável. Dados com $\xi_i = 0$ são corretamente classificados, o que significa que estão do lado correto da margem ou sob ela. Pontos com $0 \leqslant \xi_i \leqslant 1$ estão dentro da margem de erro, mas do lado correto da fronteira de decisão. Pontos com $\xi_i \geqslant 1$ estão fora da fronteira de decisão e são classificados incorretamente (BISHOP, 2006, pág.332). A soma dos erros ξ_i pode ser associada a um limite de erro máximo de erros de treinamento (LORENA et al., 2007). Com isso, a equação primal de minimização para SVM com margens suaves é dada pela Equação 2.17.

$$\begin{gathered} \underset{\mathbf{w},b,\boldsymbol{\xi}}{\text{Minimizar}} \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{i=1}^{n} \xi_i\right) \\ \text{Com restricões: } y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \geqslant 1 - \xi_i, \quad \xi_i \geqslant 0, \forall i = 1, \ldots, n \end{gathered} \tag{2.17}$$

Na equação, além de $\mathbf{w}$ e b, deseja-se agora minimizar também ξ que é o erro associado a cada amostra de treino. O fator C é de regularização e pondera a soma dos erros marginais em relação a complexidade do modelo (dado pela minimização de $\|\mathbf{w}\|^2$ no processo). Para valores altos de C, o problema possui menos erros no treinamento e tende a ser menos generalista de maneira que para C tendendo ao infinito o problema se torna idêntico ao problema de SVM de margens rígidas. Para valores baixos de C o modelo permite que alguns dados sejam classificados incorretamente (aumenta $\sum_{i=1}^{n} \xi_i$), mas isso pode tornar o modelo mais generalista. A Figura 4 ilustra o caso e nesta observa-se que a separação ilustrada na figura da esquerda possui maior assertividade, porém é menos

generalista (menor distancia entre as margens), ao contrário da figura da direita que possui erros de classificação, mas é generalista a novos dados. O segundo caso compreende situações que são mais robustas quanto a dados ruidosos.

Figura 4 – Comparativo da influência do fator de regularização C no SVM de margens suaves



(a) Valor do Parâmetro C alto (b) Valor do Parâmetro C baixo

Fonte: Lorena et al. (2007)

O problema de otimização para o parâmetro C é mais uma vez quadrático, com as restrições geradas pela Equação 2.17. Resolvendo por uma função Lagrangiana, tem-se o seguinte problema Dual conforme indicado na Equação 2.17.

$$\begin{aligned} &\underset{\alpha}{\text{Maximizar}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) \\ &\text{Com as restrições} \begin{cases} 0 \leqslant \alpha_i \leqslant 1, \quad \forall i = 1, \ldots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \end{aligned} \tag{2.18}$$

O vetor $\boldsymbol{\alpha}^*$representa a solução do problema dual e os vetores $\mathbf{w}^*$, b^* e ξ^* representam a solução do problema primal. O valor de $\mathbf{w}^*$ pode ser encontrado pela Equação 2.8 e ξ^* pode ser determinado conforme Equação 2.19 (LORENA et al., 2007). As condições de KKT para SVM de margens flexíveis e mais considerações a respeito do algoritmo estão explicados em Lorena et al. (2007) e Bishop (2006, pág.331-335). A Equação 2.14 para classificadores de margens rígidas também se aplicada para as SVM de margens flexíveis, com exceção de que as variáveis de cada caso são encontradas por diferentes equações, como demonstrado.

$$\xi_i^* = \max \left\{ 0, 1 - y_i \sum_{j=1}^{n} y_j \alpha_j^* \mathbf{x}_j \cdot \mathbf{x}_i + b^* \right\} \tag{2.19}$$

2.1.3 SVMs não lineares

Muitos problemas reais não apresentam separação linear dos dados, fazendo com que os modelos de SVM lineares não sejam eficiente na classificação dos dados. Contudo, o teorema de Cover (LORENA et al., 2007) demonstra que é possível aplicar uma transformação $\mathbf{\Phi} : \mathbf{X} \rightarrow \Im$, sendo $\mathbf{X}$ o espaço vetorial dos dados de treino e $\Im$ um novo espaço onde os dados podem ser separados por um hiperplano. Duas condições são necessárias para isso ocorrer: a primeira é que a separação seja não linear e a segunda é que a dimensão do novo espaço $\Im$, também chamado de espaço de características, seja suficientemente alta, podendo até mesmo ser infinita (LORENA et al., 2007).

Para o cálculo dos hiperplanos no espaço $\Im$, deve-se conhecer uma função $\mathbf{\Phi}$, ou mesmo encontrá-la, capaz de transformar os dados de tal maneira que, no novo subespaço, os dados sejam linearmente separáveis. O caso modificado do SVM linear com margens flexíveis, aplicando uma transformação não-linear $\mathbf{\Phi}$ sob os dados de treino, consiste de um problema de minimização apresentado na Equação 2.20.

$$\begin{aligned} &\underset{\alpha}{\text{Maximizar}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\mathbf{\Phi}\left(\mathbf{x}_i\right) \cdot \mathbf{\Phi}\left(\mathbf{x}_j\right) \right) \\ &\text{Com as restrições} \begin{cases} 0 \leqslant \alpha_i \leqslant 1, \quad \forall i = 1, \ldots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \end{aligned} \tag{2.20}$$

E, de forma semelhante a Equação 2.14, o classificador geral para um SVM com transformação não-linear dos dados é dado conforme Equação 2.21, em que b^* é obtido pela Equação 2.22 também adaptado da Equação 2.13.

$$g(\mathbf{x}) = \operatorname{sgn}\left(\sum_{\mathbf{x}_i \in \mathrm{SV}} y_i \alpha_i^* \left(\mathbf{\Phi}\left(\mathbf{x}_i\right) \cdot \mathbf{\Phi}\left(\mathbf{x}_j\right) + b^* \right) \right) \tag{2.21}$$

$$b^* = \frac{1}{n_{\mathrm{SV}:\alpha^* < C}} \sum_{\mathbf{x}_j \in \mathrm{SV}:\alpha_j^* < C} \left(\frac{1}{y_j} - \sum_{\mathbf{x}_i \in \mathrm{SV}} \alpha_i^* y_i \mathbf{\Phi}\left(\mathbf{x}_i\right) \cdot \mathbf{\Phi}\left(\mathbf{x}_j\right) \right) \tag{2.22}$$

O fato do espaço de características $\Im$ ter dimensão alta, ou mesmo infinita, implica que a função $\mathbf{\Phi}$ dos dados pode ser custosa ou inviável (LORENA et al., 2007). Contudo, o uso de *kernels* é empregado para calcular o produto interno da função dos vetores no espaço de características, sem determinar explicitamente a função de mapeamento. A vantagem de usar *kernels* é conseguir expressar espaços abstratos de uma maneira simples, sem calcular a função de mapeamento dos pontos. Conforme cita Lorena et al. (2007): "É comum empregar a função *kernel* sem conhecer o mapeamento $\mathbf{\Phi}$, que é gerado implicitamente. A utilidade dos *kernel* está, portanto, na simplicidade de seu cálculo e em sua capacidade de representar espaços abstratos".

Uma função *kernel* K definida pela Equação 2.23 pode ser diretamente substituída nas Equações 2.21 e 2.22 para cálculo direto dos produtos internos da funções $\mathbf{\Phi}$. Os *kernels* mais utilizados estão descritos em (LORENA et al., 2007). A última coluna da tabela apresenta os parâmetros de regulação do *kernel*, que são hiperparâmetros do algoritmo de treinamento.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j) \tag{2.23}$$

2.1.4 SVM Multi-Classes

O algoritmo do SVM foi fundamento como classificador binário. Contudo, existem estratégias para resolver problemas com quantidade maiores de classes.

A abordagem "um contra todos" otimiza k modelos SVM, sendo k o número de classes existentes. Para cada modelo, escolhe-se um distinto rótulo para ser classificado como positivo enquanto as $k-1$ classes restantes são rotuladas como negativas. Cada classificador retorna um valor y_k. Um novo dado é testado por todos modelos, de maneira que aquele que apresenta maior valor y_k é entendido como o que melhor separa os dados, e assim, então, classifica o dado com o rótulo k. Os problemas para essa abordagem são que cada rótulo é classificado por diferentes tarefas, e não há garantia que os valores y_k obtidos estão na mesma escala. Outro problema é o desbalanceamento de dados em cada classificação, uma vez que $k-1$ classes serão agrupadas para serem classificadas contra apenas uma única classe (BISHOP, 2006, pág.338-339)

Outra abordagem mais comum é a "um-contra-um" e nesta são construídos modelos para classificar cada classe k contra cada uma das outras $k-1$ individualmente, totalizando $k(k-1)/2$ modelos treinados. Um novo dado de teste é então classificado por todos os classificadores e, por um esquema de maioria de votos, o rótulo k do dado é então identificado. A desvantagem dessa abordagem é que mais modelos são necessários para serem treinados, se comparado a abordagem "um contra todos". Contudo, melhores resultados são obtidos aplicando esta abordagem (CHANG; LIN, 2013) .

2.2 Redes LSTM

Arquiteturas LSTM (*Long-short Term Memory*), que em tradução livre pode ser entendida como memórias curtas de longo prazo, foram utilizados neste trabalho com a finalidade de extrair características de dependência temporal do sinal e do espectrograma da voz.

A estrutura de uma rede LSTM é derivada de Redes Neurais Recorrentes (*Recurrents Neural Networks* – RNN), as quais surgiram também das tradicionais Redes

Neurais Perceptron Multi-Camada(*Multi-layer Perceptron* – MLP). Os próximos parágrafos formalizam uma breve definição e equacionamento sobre redes MLP e RNN e como a arquitetura LSTM consegue adaptá-las para bons resultados em classificação de série de dados sequenciais.

2.2.1 Redes Neurais Perceptron Multi-Camada

Conforme Graves (2011, pág.13):

> "Uma MLP com um particular conjunto de valores de pesos define uma função de vetores de entrada para vetores de saída. Alterando os pesos, uma simples MLP é capaz de representar muitas funções distintas. [...] é provado que uma MLP com uma única camada contendo um número suficientes de unidades não lineares pode aproximar qualquer função contínua de um domínio de entrada a precisão arbitrária. Por essa razão, MLPs são identificadas como aproximadores universais de função." (GRAVES, 2011, pág.13, tradução do autor).

O objetivo de uma rede MLP é, portanto, aproximar alguma função. Conforme Goodfellow, Bengio e Courville (2016, pág.164, tradução do autor), um classificador "$y = f^*(\boldsymbol{x})$ mapeia um vetor de entrada para uma categoria y. Uma rede define uma mapeamento $\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\theta})$ e aprende os valores de $\boldsymbol{\theta}$ que resultam na melhor aproximação" .

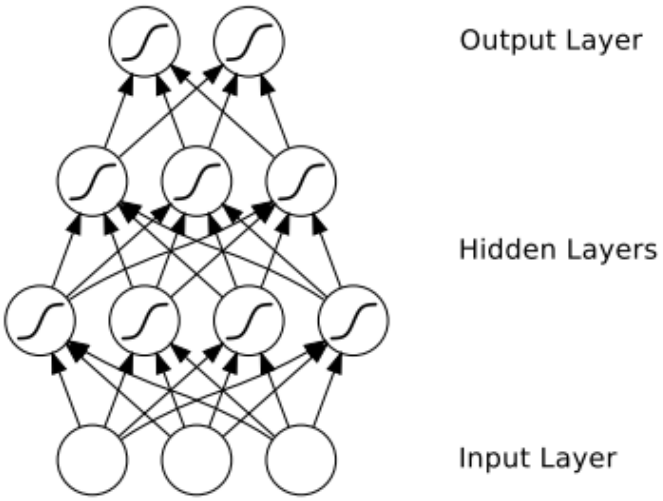
A estrutura de uma MLP pode ser representada conforme a Figura 5. Salienta-se que uma MLP é constituída de uma camada de entrada, de pelo menos uma camada escondida, e uma camada de saída que retorna os valores resultantes das operações entre os pesos da camada anterior. A camada de saída é a última camada da rede e seus valores estabelecem uma aproximação $\boldsymbol{y}^*$ para os valores $\boldsymbol{y}$ que se deseja mapear, de maneira que a quantidade de neurônios nesta camada é também da dimensão de $\boldsymbol{y}$.

Uma MLP consiste de um conjunto de H camadas que são interligadas por uma matriz de pesos $w_{h'h}$ que interligam uma camada imediatamente anterior h' com a camada atual h. A saída a_h da camada h é definida como o produto da entrada de dados da camada pela matriz de pesos, conforme Equação 2.24. Uma função de ativação, responsável por assegurar que a função de mapeamento consiga representar não-linearidades dos dados, é definida por θ_h, sendo aplicada sobre a saída da camada, transformando a saída em b_h, conforme Equação 2.25 (GRAVES, 2011, pág.13).

$$a_h = \sum_{h' \in H_{l-1}} w_{h'h} b_{h'} \tag{2.24}$$

$$b_h = \theta_h \left(a_h \right) \tag{2.25}$$

Figura 5 – Estrutura de uma rede MLP, com uma camada de entrada, intermediária e saída.



Fonte: Adaptado de Graves (2011, pág.13)

Em redes neurais modernas, recomenda-se por padrão funções de ativação ReLU (*Rectified Linear Unit* - Unidade Linear Retificada), expressa pela Equação 2.39. Essa função é aproximadamente linear, o que preserva algumas características que fazem modelos mais generalistas, além de ajudar nos métodos de cálculo de gradientes (GOODFELLOW; BENGIO; COURVILLE, 2016, pág.170).

$$\sigma(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2.26}$$

Para problemas de classificação, a camada de saída da MLP consiste de uma matriz de pesos e de uma função de ativação especial que calcula a possibilidade do dado pertencer a cada uma das classes. O vetor de saída $\boldsymbol{y}$ de uma MLP é encontrado através da função de ativação das unidades da camada de saída, exatamente como feito em outras camadas. Para problemas de classificação com número de classes $k > 2$, é comum utilizar uma função de ativação que normaliza a saída e calcula a probabilidade das classes de maneira que a soma de todas seja unitária (GRAVES, 2011, pág.15). A função *softmax*, uma das mais recentes e utilizadas funções de saída de MLP, é definida conforme Equação 2.2.1. Outras funções de ativação para camada de saída, como função logística sigmoidal, são apresentadas em Goodfellow, Bengio e Courville (2016, pág.178-183). A classe escolhida como resultado da classificação é aquela que apresenta maior probabilidade entre aquelas calculadas na camada de saída. Na equação, k' representa cada um dos K neurônios da camada de saída, sendo k o índice do neurônio em questão, cuja saída é determinada por y_k e $p\left(C_k|x\right)$ é a probabilidade da entrada da rede x ser pertencente a classe C_k. Considera-se que, se y_k é maior que os demais $y_{k'}$, o dado de entrada é rotulado como pertencente a

classe C_k.

$$p\left(C_k|x\right) = y_k = \frac{e^{a_k}}{\sum_{k'=1}^{K} e^{a_{k'}}} \tag{2.27}$$

A etapa de treinamento consiste na atualização dos pesos das camadas da rede. Após um dado de entrada passar pela rede, sua saída é calculada e esse valor é comparado a sua classe objetivo, também chamada de *target*. A assertividade do modelo é calculada por uma função perda. Para tarefas de regressão, a perda mais usual é a perda média quadrática – (*Mean Squared Error* – MSE), descrita na Equação 2.28, enquanto que para tarefas de classificação utiliza-se a perda por Entropia Cruzada, descrita na Equação 2.29 (GRAVES, 2011, pág.16). Na equação, $\mathbf{y}\left(\mathbf{x}_n, \mathbf{w}\right)$ consiste da saída da rede para os pesos $\mathbf{w}$ e a saída $\mathbf{x}_n$, t_n consiste de um vetor de zeros da dimensão do número das N classes, com exceção de um elemento que é unitário representando a classe do dado.

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} \left\|\mathbf{y}\left(\mathbf{x}_n, \mathbf{w}\right) - \mathbf{t}_n\right\|^2 \tag{2.28}$$

$$E(\mathbf{w}, \mathbf{t}) = -\sum_{n=1}^{N} t_n \ln y_n \tag{2.29}$$

O objetivo de atualizar os pesos das camadas é minimizar o erro de predição entre a saída da rede com as *targets* estabelecidas, num processo de otimização. Um dos otimizadores pioneiros em MLP é o Método do Gradiente Descendente (*Gradient Descendent* - GD). A abordagem consiste em atualizar os pesos na direção do gradiente negativo do erro de predição em relação aos pesos. A Equação 2.30 ilustra como os pesos são atualizados. O parâmetro $\eta > 0$ é denominado taxa de aprendizado, e é um hiperparâmetro de treinamento da rede. Na Equação 2.28 observa-se que o erro $E(\mathbf{w})$ é calculado sobre a predição de todo o conjunto de dados N. Em abordagens de redes profundas, N pode ser um número alto e ∇E pode ter cálculo custoso. Dessa necessidade, surge a definição de *batch*, que treina a rede em pacotes menores de dados (a quantidade varia com a arquiteturas), nos quais cada atualização dos pesos da rede acontece quando um *batch* de dados passa pela rede.

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta\nabla E\left(\mathbf{w}^{(\tau)}\right) \tag{2.30}$$

Define-se uma época o uso de todo o conjunto de treinamento (organizado em lotes) no processo de iterativo do gradiente descendente agrupado em mini-lotes (*batches*). *Backpropagation* (retropropação) é o algoritmo que permite calcular de maneira eficiente o gradiente da função de perda avaliada para cada lote. É importante observar o que é dito

em (GOODFELLOW; BENGIO; COURVILLE, 2016, pág.241-242) a respeito das etapas de propagação do erro pela rede e também da atualização dos pesos.

> "Muitos algoritmos de treino envolvem um processo interativo para minimização de uma função de erro que ajusta os pesos numa sequência de etapas. Cada etapa pode ser distinguida entre dois estagio: em um primeiro, as derivadas da função erro são calculadas em relação aos pesos avaliados.[...] Em um segundo estágio, as derivadas são então utilizadas para calcular os ajustes dos pesos. [...] É importante reconhecer que os estágios são distintos. No primeiro, chamado propagação de erro, os erros são propagados através da rede [...] e isso pode ser aplicado a muitos outros tipos de rede, e não somente ao perceptron. Ele também pode ser aplicado a outras funções de erro,[...] como as matrizes Jacobianas e Hessianas. Da mesma forma, o segundo estágio de ajuste de pesos utilizando as derivadas calculadas pode ser realizado utilizando uma variedade de esquemas de otimização, muitos dos quais são substancialmente mais poderosos do que o simples Gradiente Descendente." (GOODFELLOW; BENGIO; COURVILLE, 2016, pág.241-242, tradução do autor)

2.2.2 Redes Neurais Recorrentes

Conforme Graves (2011, pág.18), enquanto redes MLP mapeiam uma entrada para cada saída, arquiteturas de redes neurais recorrentes (*Recorrent Neural network* – RNN) obtém um histórico de entradas anteriores para mapear uma saída. Ainda de acordo com os autores, ao passar o histórico de entradas junto com a entrada atual, os pesos das camadas recorrentes aprendem padrões entre as entradas e o seu histórico, tendo assim uma camada de pesos que funciona como uma "memória" de entradas anterior. Esses pesos da camada recorrente persistem num estado interno da rede, e acabam por influenciar a saída sobre uma nova entrada, diferente do que ocorre com uma rede MLP que só possui pesos associados a uma única entrada. A aplicação de arquiteturas RNN são utilizadas em sinais cujo o instante atual tem alta dependência com instantes passados, como ocorre por exemplo em séries temporais (ZHAO; MAO; CHEN, 2019). Diferente da MLP, uma rede recorrente tem como entrada vários instantes de uma série temporal, e a rede é treinada para reconhecer padrões entre esses instantes (GRAVES, 2011, pág.18).

Considere x_i uma amostra de um conjunto de T observações e seja $\left[x_i^t, x_i^{t+1}, x_i^{t+2}, \cdots\right]$ um conjunto de subsquências de x_i com um tamanho definido l. Esse conjunto de sequência é a entrada da rede, sendo que o tamanho de cada subsequência corresponde a dimensão de entrada da rede recorrente. A título de exemplo, ao treinar a rede com a sequência $[1, 2, 3 \cdots 8, 9, 10]$ em amostras de tamanho de sequência $l = 4$, com passos de 2, tem-se um conjunto de N sequências formadas $[1, 2, 3, 4]$, $[3, 4, 5, 6]$,$[5, 6, 7, 8]$ e $[7, 8, 9, 10]$, que serão passados como dados de treino. O número de elementos na sequência, assim como o tamanho do passo, são hiperparâmetros da rede. Sendo N a quantidade de dados de entrada (ou número de observações), d a dimensão de cada amostra, pode-se dizer que um tensor de tamanho (N, l, d) é formado. Considera-se uma matriz de pesos $w_{h'h^t}$ que

conecta a saída de uma camada escondida de um instante até o próximo. Essa matriz vai ser treinada para obter uma função que determine a dependência que existe entre uma saída anterior e a atual. Como essa matriz é atualizada tendo em vista os valores atuais, com os anteriores, diz-se que ela é recorrente.

O método de *backpropagation* e Gradiente descendente podem ser utilizados em Redes Neurais Recorrentes. O algoritmo de *backpropagation* que leva em consideração os pesos das matrizes recorrentes, bem como as derivadas parciais em relação ao tempo é identificado por *backpropagation throught time* (retropropagação pelo tempo) (GRAVES, 2011, pág.19).

2.2.3 LSTM - *Long Short-Term Memory*

Redes Neurais Recorrentes sofrem de memória de curto prazo, devido ao fenômeno de desaparecimento ou explosão dos gradientes que são calculados no método de *backpropagation*(PASCANU; MIKOLOV; BENGIO, 2013). Soluções encontradas para resolver o problema do gradiente se encontra na utilização de funções de ativações como a ReLU ou mesmo utilizando arquiteturas diferenciadas como a *Gated Recurrent Unit* (GRU) e a LSTM, que são arquiteturas de redes neurais recorrentes adaptadas para diminuir o problema de explosão ou desaparecimento de gradientes.

As arquiteturas LSTM se diferenciam das arquiteturas RNN pois cada célula apresenta funções com capacidade de armazenar informações relevantes a longo prazo bem como também esquecer informações mais irrelevantes. Em SER eles têm sido muito utilizadas conectadas as camadas de CNN, como pode ser visto em Zhao, Mao e Chen (2019) e Fayek, Lech e Cavedon (2017). A característica das LSTM de armazenar informações a longo prazo se torna poderosa para identificação de sentimentos, uma vez que a característica do sentimento na fala é predominante a longo prazo. O trabalho de Fayek, Lech e Cavedon (2017) apresenta o resultado para várias arquiteturas de CNN construídas, incluindo aquelas que apresentam camadas de LSTM.

Nas LSTM, cada célula constitui de uma adaptação de neurônio de uma rede RNN, de maneira que uma camada LSTM possui um número de entradas conforme o número de células escolhidas. O artigo de Varsamopoulos e Bertels (2018) simplifica e ilustra a estrutura de uma célula LSTM, por meio da Figura 6 e das Equações 2.31 a 2.36 listadas abaixo.

$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i\right) \tag{2.31}$$

$$f_t = \sigma\left(x_t U^f + h_{t-1} W^f\right) \tag{2.32}$$

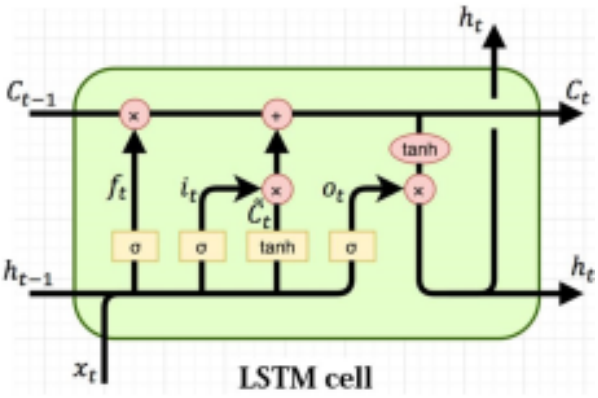
$$o_t = \sigma\left(x_t U^o + h_{t-1} W^o\right) \tag{2.33}$$

$$\tilde{C}_t = \tanh\left(x_t U^g + h_{t-1} W^g\right) \tag{2.34}$$

$$C_t = \sigma\left(f_t \odot C_{t-1} + i_t \odot \tilde{C}_t\right) \tag{2.35}$$

$$h_t = \tanh\left(C_t\right) \odot o_t \tag{2.36}$$

Figura 6 – Estrutura de uma célula LSTM



Fonte: Varsamopoulos e Bertels (2018)

A arquitetura LSTM busca contornar o problema de desaparecimento e explosão de gradientes por meio da ativação ou desativação das células por meio do cálculo de um estado C_t, que é função das saídas de portões (*gates*) e também do estado anterior C_{t-1} da célula. Existem três tipos de portões (*gates*) em cada célula LSTM: *input gate* (i_t), *forget gate* (f_t), *output gate* (o_t). A Figura 6 indica as portas e os estados.

Nas Equações 2.31 a 2.36, x_t é o vetor de entrada no instante atual t e h_t e h_{t-1} são as saídas atual e anterior das camadas escondidas, respectivamente. As matrizes de peso W apresentam a conexão recorrente entre os estados h_t da camada escondida anterior com a atual e as matrizes U conectam as entradas x_t com a camada escondida. Essas matrizes serão atualizadas durante a aprendizagem do modelo, de maneira que U^i e W^i são referentes aos pesos de *input gate*, U^f e W^f são referentes aos pesos de *forget gate* e U^o e W^o são referentes aos pesos de *output gate*. A função $\sigma()$ representa a função de ativação sigmoidal e a função $tanh()$ é a função de ativação tangente hiperbólica.

O *input gate* i_t, representado na Equação 2.31, tem a finalidade de ativar ou desativar o estado da célula em função das informações das amostras atuais. Da mesma

forma, a Equação 2.34 determina $\tilde{C}_t$, que é o valor candidato ao novo estado da célula encontrado pelas amostras atuais. O *forget gate*, representado na Equação 2.32, penaliza a amostra atual por meio de uma função de esquecimento.

A Equação 2.35 mostra que valor de i_t será combinado com o valor $\tilde{C}_t$ por um produto Hadamard (produto ponto-a-ponto representado nas equações por $\odot$) assim como f_t é combinado pelo mesmo produto com o estado C_{t-1}, de maneira que a soma desses produtos indica o estado atual C_t da célula. Pode-se dizer que f_t será aprendido calculando-se a importância do estado anterior C_{t-1} sobre o estado atual. É por isso que entende-se que f_t funciona como um fator de esquecimento, pois ele pode reduzir o peso do estado anterior sobre o atual. Da mesma forma, i_t aprende qual a importância da amostra atual sobre o estado da célula atual C_t, pois ele pondera o quanto do valor $\tilde{C}_t$ influencia no cálculo de C_t.

A saída h_t atual da célula é ponderada pelo portão o_t, e são calculado por meio das Equações 2.33 e 2.36. Observa-se que a saída da célula h_t depende do estado atual C_t ponderado pelo portão o_t, sendo que o valor desse portão é função tanto da entrada atual, como também do estado anterior da célula, conforme Equação 2.33. O cálculo da saída o_t se assemelha muito com o cálculo da saída h_t de uma camada escondida MLP. A diferença do cálculo de h_t na MLP para LSTM é visto na Equação 2.36, que faz com que a saída h_t atual seja função tanto de o_t, como também do estado da célula C_t, que foi calculado por meio de outros portões. É justamente o valor do cálculo de C_t que torna diferente as arquiteturas de MLP e LSTM (GRAVES, 2011, pág.35).

As equações mais detalhadas, incluindo aquelas para *backpropagation* e saída da rede são bem expressas em (GRAVES, 2011, pág.35-38) e também em (VARSAMOPOULOS; BERTELS, 2018). A finalidade de uso em LSTM neste trabalho é justificada pelo fato do sinal de voz ser uma série temporal, de maneira que uma rede pode ser treinada e reconhecer padrões de recorrência para diferentes estados de emoções. Como será explicado melhor na seção seguinte, utilizar LSTM numa arquitetura de rede neural ajuda na identificação de características locais do sinal. Ou seja, uma rede LSTM é capaz de extrair o que aqui se chama de características temporais da voz, identificando padrões que são características do tempo.

2.3 Redes Neurais Convolucionais Associadas a LSTM

Redes Neurais Convolucionais (*Convolutional Neural Networks* – CNN) são tipos especiais de redes neurais que processam dados de topologia espacial ou temporal, como dados de séries temporais ou mesmo imagens, por meio de operações de convolução com finalidade de extração de características. Conforme explicado em Goodfellow, Bengio e Courville (2016, pág.326, tradução do autor) “redes neurais convolucionais são simplesmente

redes neurais que utilizam uma convolução no lugar de uma matriz de multiplicação geral em pelo menos uma das camadas".

Em reconhecimento de emoção por voz, redes convolucionais unidimensional são utilizadas tanto para encontrar informações de alto nível do sinal de voz temporal, bem como redes convolucionais bidimensionais são tipicamente empregadas na extração de características do espectrograma da fala (BADSHAH et al., 2017; ZHAO; MAO; CHEN, 2019). Trabalhos como os de Badshah et al. (2017), Zhao, Mao e Chen (2019) e Satt et al. (2017) apresentam diferentes abordagens de uso de redes neurais convolucionais para classificação de sentimento em voz além do uso de diferentes topologias de arquitetura.

O uso de arquiteturas de redes profundas tem se ampliado em identificação de sentimento em fala. Conforme Zhao, Mao e Chen (2019, tradução do autor), o "processamento de sinal de voz tem sido revolucionado por *deep learning*. Mais e mais pesquisas tem alcançado excelentes resultados em certas aplicações usando redes de crença profunda, redes convolucionais e redes de memória curto prazo".

Em uma operação de convolução unidimensional, um *kernel* $w(n)$ de dimensão $2 \times l - 1$ é aplicado sobre um sinal $x(n)$ e retorna um valor escalar $z(n)$, conforme pode ser visto na Equação 2.37. Para a convolução de um sinal bidimensional, um *kernel* $w(i, j)$ de dimensão $2a - 1 \times 2b - 1$ é aplicado sobre o sinal $x(i, j)$ e retorna um escalar $z(i, j)$, conforme Equação 2.38.

$$z(n) = x(n)^* w(n) = \sum_{m=-l}^{l} x(m) \cdot w(n - m) \tag{2.37}$$

$$z(i, j) = x(i, j)^* w(i, j) = \sum_{s=-a}^{d} \sum_{t=-b}^{b} x(s, t) \cdot w(i - s, j - t) \tag{2.38}$$

A operação de convolução extrai características locais sobre o sinal em cada frame do sinal (ZHAO; MAO; CHEN, 2019). Em uma Rede Neural Convolucional, as matrizes de peso que representam os *kernels* da operação de convolução são otimizadas durante o treino da rede. A inicialização dos *kernels* costuma ser aleatória. Os *kernels* obtidos após o treino da rede não possuem ainda muita interpretação física. Conforme Zhao, Mao e Chen (2019):

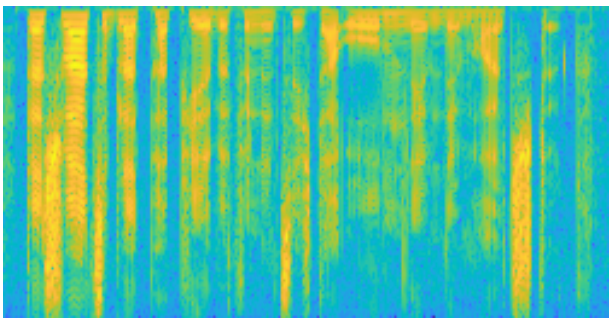
> Redes neurais profundas são tipicamente abordagens de "caixas pretas" pois é extremamente difícil entender como a saída final é alcançada. [...] a interpretação de como as características altamente abstratas são aprendidas por redes neurais profundas ainda é pobre. Mas o desempenho de redes neurais profundas tem sido muito melhor que as abordagens tradicionais. (ZHAO; MAO; CHEN, 2019, tradução do autor)

Dois tipos de arquiteturas de CNN foram utilizados nesta pesquisa. Uma rede convolucional temporal unidimensional que extrai características unidimensionais do

sinal de áudio de entrada e uma rede convolucional espacial bidimensional, que extrai características do sinal por meio de operações sobre espectrogramas gerados a partir dos dados de voz.

Espectrogramas são representações visuais do espectro do sinal da voz ao longo do tempo. Sob um sinal de fala criam-se várias janelas, que podem inclusive serem sobrepostas. A cada janela de N amostras aplica-se a Transformada de Fourier (TF). Após o procedimento, para cada uma das M janelas tem-se um espectro de Fourier de tamanho N. A matriz $M \times N$ criada pode ser vista como uma imagem, onde o valor da TF de cada coluna é expressa em uma escala de cores. A imagem formada denomina-se espectrograma do sinal (BADSHAH et al., 2017). Vale salientar que o espectrograma é uma representação bidimensional do intervalo de tempo de um sinal unidimensional, no qual o tempo é indicado no eixo horizontal, a frequência no eixo vertical e a amplitude da frequência por uma escala de cores aplicada (BADSHAH et al., 2017). A Figura 7 apresenta o espectrograma de uma dos áudios da base de dados Berlin (BURKHARDT et al., 2005).

Figura 7 – Espectrograma de um dos áudios da base de dados Berlin



Fonte: Autor

As saídas de cada camada convolucional é passada por uma função de ativação. Algumas funções são recorrentes, como a ReLU e a ELU (*Exponential Linear Unit* - Unidade Exponencial-Linear). As equações de ambas estão apresentadas em 2.39 e 2.40. Na Equação 2.41, z_i^l e z_i^{l-1} representam a i-ésima característica de saída da l-ésima camada e z_j^{l-1} representa a entrada da característica na $(l-1)$-ésima camada, σ representa a função de ativação escolhida e w_{ij}^l representa a matriz de convolução entre a i-ésima e j-ésima característica. A função de ativação utilizada na arquitetura de CNN deste trabalho foi a ELU. Esta foi escolhida, conforme explicado em (ZHAO; MAO; CHEN, 2019), devido a sua característica de assumir valores negativos de função de ativação, diferente da ReLU, o que faz com que a média das características seja trazida próxima de zero, fazendo com

que acelere o aprendizado da rede, aumentando a acurácia final.

$$\sigma(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2.39}$$

$$\sigma(x) = \begin{cases} x & x \geq 0 \\ \alpha\left(e^{x} - 1\right) & x < 0 \end{cases} \tag{2.40}$$

$$z_i^l = \sigma\left(b_i^l + \sum_j z_j^{l-1} * w_{ij}^l\right) \tag{2.41}$$

Arquiteturas de redes convolucionais possuem camadas adicionais, como as de *pooling* e de normalização de *batch* (*Batch Normalization*-BN) que processam o sinal resultante da operação de convolução. O BN aplica uma normalização de média e desvio padrão nos dados de entrada, sendo que os parâmetros dessa normalização também são aprendidos. Essa técnica tem encontrado resultados promissores em algumas aplicações de redes convolucionais, quando se associa camadas de BN à camadas convolucionais (ZHAO; MAO; CHEN, 2019). A aplicação de BN altera a saída da camada convolucional, conforme pode ser visto na Equação 2.42. Camadas de *pooling* aplica sobre as características uma função não linear de subamostragem que reduz a dimensão das características. A *max-pooling* é a utilizada nesse trabalho e a saída desta função consiste do maior valor da região de *pooling* selecionada. A Equação 2.43 apresenta a saída de uma camada de *pooling*, sendo Ω_k uma região sob a qual a máscara de *pooling* é aplicada, z_p^l e z_k^l representam, respectivamente, a entrada e a saída da l-ésima camada de *pooling*.

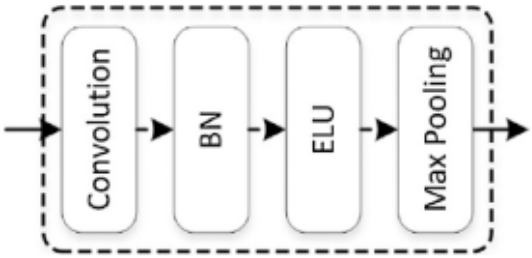
$$z_i^l = \sigma\left(BN\left(b_i^l + \sum_j z_j^{l-1*} w_{ij}^l\right)\right) \tag{2.42}$$

$$z_k^l = \max_{\forall p \in \Omega_k} z_p^l \tag{2.43}$$

A seção 1.2 explica a definições de informações de alto nível (globais) ou de baixo nível (locais). No contexto de redes neurais, uma rede convolucional funciona como um extrator de características locais (por vezes também chamadas de salientes), que representam as informações de baixo nível tanto quando aplicadas no sinal unidimensional quanto no espectrograma do sinal (BADSHAH et al., 2017; ZHAO; MAO; CHEN, 2019). As primeiras camadas de uma CNN extraem características de baixo nível, que após serem treinadas e aprendidas, representam informações aqui chamadas de informações locais. À medida que se aumenta a profundidade da rede, características de maior complexidade são extraídas, que após aprendidas são traduzidas em informações que são chamadas de globais. Zhao, Mao e Chen (2019) denominam o bloco de camadas responsáveis pela

extração de características convolucionais como “bloco de extração de informações locais”. De acordo com o trabalho dos autores, um bloco de informação local consiste de uma camada convolucional associada com camada de ativação, de normalização de *batch* e de *pooling*. O esquema de um bloco de extração local de características está apresentado na Figura 8. Vários blocos individuais de extração de características locais podem ser associados, de maneira a formar uma rede de extração de características mais profunda, que podem ser posteriormente analisadas por outras arquiteturas de classificador, como redes densas, que geralmente são constituídas de redes MLP comum, ou mesmo de redes recorrentes e até mesmo de modelos SVM (HUANG et al., 2014).

Figura 8 – Bloco de extração de características locais convolucionais



Fonte: Zhao, Mao e Chen (2019)

As características extraídas pelo bloco local podem ser analisadas por meio de redes recorrentes LSTM. Uma sequência de características locais pode ser aprendida como uma função de membros de dados sequenciais, e redes LSTM são especialistas no processamento de dados com dependência temporal. Conforme Zhao, Mao e Chen (2019) as características locais extraídas do sinal de voz aplicadas em redes LSTM resulta numa combinação que permite identificar informações globais e de alto nível acerca do sinal processado. Arquiteturas que combinam ambas estruturas de CNN com LSTM são aqui chamadas de redes CNN-LSTM. Uma rede CNN-LSTM-1D indica aplicação da rede sob um sinal de voz unidimensional, enquanto que CNN-LSTM-2D realizam a operação sobre o espectrograma do sinal da voz.

Inspirado nos trabalhos de Badshah et al. (2017), Satt et al. (2017) e Zhao, Mao e Chen (2019), as Figuras 9 e 10 ilustram a estrutura CNN-LSTM-1D e CNN-LSTM-2D utilizadas neste trabalho, respectivamente. Detalhes sobre o tamanho dos dados de entrada estão apresentados na Seção 3.7.2. Vale observar que as estruturas definidas foram validadas em trabalhos experimentais. Desta forma, definições de estrutura das redes como a quantidade de filtros utilizados, número de camadas convolucionais, entre outras podem ser melhor explicadas nos artigos base já citados.

Figura 9 – Estrutura de rede convolucional unidimensional CNN-LSTM-1D

Input: Sinal 1 x 9600
Conv. 1: 64 filtros (1x3, *strides=1)*
Batch-Normalization
ELU
Max-pooling (4,strides=4)
Conv. 2: 128 filtros (1x3, *strides=1)*
Batch-Normalization
ELU
Max-pooling (4,strides=4)
LSTM (256)
Flatten
Softmax
Saída : Número de classes do tipo de rótulo

Fonte: Autor

Figura 10 – Estrutura de rede convolucional bidimensional CNN-LSTM-2D

Input: Espectrograma 129x135
Conv. 1: 120 filtros (13x13, *strides=4)*
ReLu
Max-pooling (3x3,strides=2)
Conv 2: 256 filtros (5x5, *strides=1)*
ReLu
Max-pooling (2x2,strides=2)
Conv 3: 384 filtros (3x3, *strides=1)*
ReLu
Max-pooling (2x2)
LSTM (256)
Densa (2048)
Softmax
Saída: Número de classes do tipo de rótulo

Fonte: Autor

2.4 Árvores de decisão Hierárquica

Árvores de decisão hierárquicas têm sido utilizadas em SER por voz para classificar sentimentos através de múltiplos classificadores. Conforme Zhou (2012, pág.04):

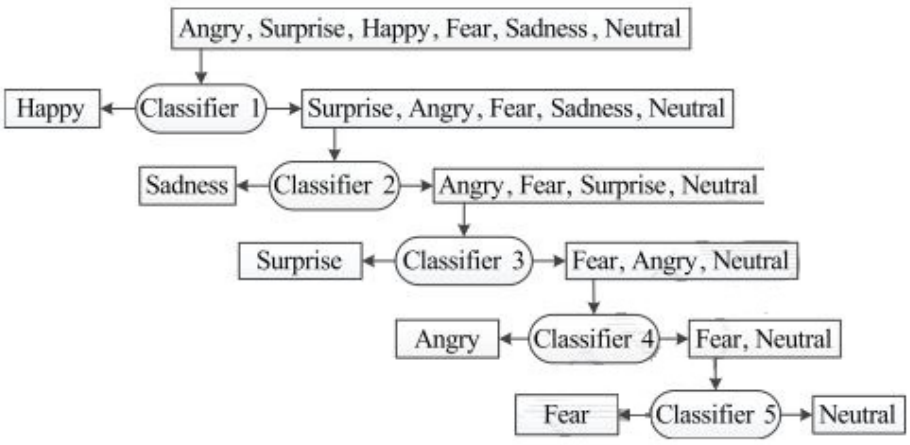
> "Uma árvore de decisão consiste de um conjunto de testes de decisão de árvore trabalhando en um caminho de dividir para conquistar. Cada nó não-folha (folha de decisão) é associado a um teste de característica, também chamado de divisão: dados passando por esse nó serão separados em diferentes subconjuntos de acordo com seus diferentes valores no teste de característica. Cada nó-folha é associado com um rótulo, que será associado com as amostras que caírem nesse nó.[...] Em cada época, um conjunto de dados é fornecido e uma divisão é selecionada, para então essa visão ser usada para dividir os dados em subconjuntos e, cada subconjunto, ser considerado como um conjunto de dados fornecido em outra etapa. A chave do algoritmo de árvore de decisão é como são

> selecionadas as divisões [dos dados]."Zhou (2012, pág.04, tradução do autor)

Lee et al. (2011), utilizam classificação de sentimentos baseadas em árvores de decisão binárias hierárquicas onde cada folha de decisão da árvore identifica individualmente uma classe de sentimentos por meio de classificadores SVM ou por Regressão Logística Bayesiana. Liu et al. (2017) e Mao (2018) utilizam classificadores SVM e ELM (*Extreme Learning Machine*) em seu modelo hierárquico. Ambos trabalhos treinam os modelos por meio de classificação um-contra-todos. No artigo de (MAO; WANG; ZHAN, 2010), por exemplo, sete classes de sentimentos são presentes e 21 modelos (combinação $C_{7,2} = 21$) são originados para separar individualmente cada classe. A classe individual do modelo de melhor desempenho é escolhido como decisão positiva para o primeiro nível hierárquico. O próximo nível da árvore tem como decisão positiva a classe individual do modelo de segundo melhor desempenho, e assim sucessivamente. Liu et al. (2017) iniciam o procedimento de maneira parecida a de (MAO, 2018) mas a cada novo nível da árvore, novos modelos são treinados excluindo-se as classes já separadas em níveis anteriores.

A Figura 11 ilustra a árvore binária hierárquica utilizada por Liu et al. (2017) em seu trabalho.

Figura 11 – Árvore de decisão hierárquica binária de sentimentos



Fonte: Liu et al. (2017)

Vários podem ser os critérios de separação dos conjuntos de dados em cada folha de decisão. A métrica aplicada nos trabalhos de Liu et al. (2017) e Mao (2018) é Grau de Confusão, que associa a probabilidade de acerto de uma classe para com as outras baseadas na Matriz de Confusão dos resultados obtidos pelos classificadores um-contra-todos (MAO, 2018).

Neste trabalho, a árvore de decisão hierárquica tem a finalidade de combinar as informações obtidos por amostras classificadas em classificadores sob diferentes rótulos, como os rótulos principais, de excitação, valência e quadrante, conforme explorado na Seção 1.4. Como cada tipo de classificador apresentarem diferentes tipos de rótulos, cada folha de decisão da árvore construída neste trabalho separa amostras baseado nos resultados de classificação de cada classe F1. A sequência com que os classificadores é definido na árvore e a maneira como as classes são separadas em cada nó será apresentada na metodologia proposta na Seção 3.3.

2.5 Técnica *Ensemble* de classificação

Técnicas de *ensemble* utilizam o treino paralelo de mais de um classificador nos quais as saídas dos modelos são combinadas de forma a garantir uma predição final baseada nos resultados individuais de cada tarefa (SHIH; CHEN; WU, 2017). Em trabalhos recentes, como o de Zhang et al. (2018), um esquema de *ensemble* constituído de quatro classificadores e a predição de cada um desses modelos é combinada por um esquema de votação para estabelecer-se uma predição final de classificação. Em Xia e Liu (2017) uma Rede de Crença Profunda é utilizada para classificar tantos sentimentos em excitação e valência de sinal, compartilhando a mesma arquitetura de rede. Isso faz com que no treino, tanto o erro encontrado pelo valor de excitação quanto pelo o de valência sejam propagados pela rede, e os pesos ajustados consistem, portanto, de um resultado de otimização compartilhado, tratando-se assim de uma caso de *Ensemble*.

Em classificadores que retornam um valor probabilístico para cada classe, o resultado final do *Ensemble* poder ser uma combinação ponderada dos resultados individuais ou mesmo o maior entre os valores individuais encontrados (SHIH; CHEN; WU, 2017). Alguns classificadores, como o SVM, podem não apresentar uma medida direta de probabilidade de classe, sendo comum a utilização de regressão logística para estimar esses resultados. A predição final de um *ensemble* também pode ser feita sobre resultados categóricos dos dados.

Tuarob et al. (2014) apresenta alguns métodos para combinar resultados categóricos de classificadores em um *Ensemble*. Os modelos *Ensemble* adotados neste trabalho apresentam o método de maioria de votos, ou moda, para predição final dos dados, onde cada classificador prediz uma classe e a mais votada é a classificação final do *Ensemble*.

Neste trabalho, a técnica de *ensemble* por maioria de votos é utilizada para combinar modelos que apresentam a predição de mesmo rótulos de sentimentos, porém com diferentes arquiteturas e finalidades. Arquitetura SVM, CNN-LSTM-1D e CNN-LSTM-2D que classificam um sinal de voz sobre um mesmo rótulo são combinadas e, por um esquema de votação, representam uma nova resposta sobre o modelo de predição.

Conforme será visto na Seção de resultados, essa técnica tende torna mais assertiva a classificação dos modelos comparado aos resultados individuais de cada um. A finalidade desse método, como explicado na seção 1.3 é contemplar uma análise Multi-visão do sinal de voz.

2.6 Métricas de avaliação de desempenho de classificação

Em uma classificação, define-se A como elementos rotulados a uma devida classe e $\overline{A}$ os elementos que não são rotulados a essa classe. Define-se como Verdadeiros Positivos (*True Positives* - TP) as amostras de A que foram classificadas corretamente como A e Verdadeiros Negativos (*True Negatives* - TN) os elementos de $\overline{A}$ que são classificados corretamente como $\overline{A}$. Define-se Falsos Positivos (*False Positives* - FP) os elementos de $\overline{A}$ que foram classificados incorretamente como sendo da classe A e Falsos Negativos (*False Negatives* - FN) os elementos da classe A que foram classificados incorretamente como $\overline{A}$. Define-se precisão e *recall* conforme Equações 2.44 e 2.45 (FLACH; KULL, 2015).

$$prec = \frac{TP}{TP + FP} \tag{2.44}$$

$$recall = \frac{TP}{TP + FN} \tag{2.45}$$

A principal métrica utilizada nesta pesquisa para avaliar o desempenho de cada classificador foi a F1-*score*, que consiste da média harmônica entre a precisão e o *recall* de cada classe. A Equação 2.46 apresenta o cálculo da métrica para cada uma das classes. Vale aqui ressaltar que uma classificação ótima é aquela que apresenta valor F1 unitário para cada classe, o que significa que todos todas amostras foram classificadas corretamente (FLACH; KULL, 2015).

$$\text{F1 -}score \triangleq 2\frac{\text{prec} \times \ Recall}{\text{prec} + \ Recall} \tag{2.46}$$

Em uma classificação, cada classe apresenta seus valores de precisão, *recall* e F1, baseados no resultados da classificação. Assim, entende-se que a média dos valores F1 encontrados para cada rótulo pondera a classificação geral como o mesmo peso para cada classe. O valor F1 calculado para essa abordagem é denominado F1-macro (FLACH; KULL, 2015).

Contudo, pode-se avaliar a classificação contando globalmente o número de verdadeiros/falsos positivos/negativos. Nesse caso, o valor F1 é calculado assumindo peso igual entre as amostras, e denomina-se F1-micro (FLACH; KULL, 2015).

A média das métricas F1 encontradas para cada classe em F1-macro podem ser ponderada pela quantidade de elementos existente em cada classe. Essa quantia é chamada de suporte. Com isso, define-se a métrica F1-ponderada como sendo a média dos resultados F1 de cada classe ponderada pelo seu respectivo suporte. Devido ao ponderamento, essa métrica pode resultar em valores de F1 que não estão entre os valores de precisão e *recall* (FLACH; KULL, 2015).

A acurácia apresenta o percentual de acertos do classificador, podendo ter uma abordagem macro, micro ou ponderada como explicado. Embora comumente utilizada na literatura, a métrica acurácia não é muito adequada para medida de classificação de bases de dados desbalanceadas, sendo melhor utilizar a métrica F1 que indica melhor os efeitos de desbalanceamento por meio dos diferentes valores de precisão e *recall* encontrados e utilizados em sua fórmula (FLACH; KULL, 2015). deste fato, definiu-se o F1-ponderado como sendo a principal métrica dos resultados encontrados nesta pesquisa, embora as métricas de acurárica, e também F1-macro ainda sejam utilizadas em alguns momentos para explorar alguns resultados.

3 Metodologia

A abordagem proposta neste trabalho para identificação de sentimentos discretos é realizada em duas etapas. Na primeira etapa, vários classificadores são construídos para identificar o sentimento em voz em diferentes aspectos (excitação, valência, quadrantes, sentimentos categóricos). Ou seja, um mesmo modelo é treinado para classificar o sentimento sobre diferentes tipos de rótulos. Na segunda etapa, as saídas destes classificadores são usadas como características de entrada em uma árvore de decisão hierárquica, que por final classifica o sinal em rótulos discretos de sentimentos. Neste texto, fala-se que a Árvore de Decisão Hierárquica realiza um aprendizado multi-tarefa, pois o algoritmo encontra combina resultados de vários classificadores, com tipos diferentes de rótulos (excitaçao, valência, quadrantes e principais) para classificar a amostra de áudio quanto a um sentimento individual.

A seção que segue detalha a estrutura da base de dados de voz Berlin. É necessário destacar as características dessa Base pois a metodologia deste trabalho foi construída com base nos número e nos tipos de sentimentos presentes nessa base. Como será visto nas seções seguintes, algumas adaptações seriam necessárias para que a metodologia desenvolvida se aplique a outras bases de voz que descrevem sentimentos distintos. As seções posteriores detalham a metodologia construída, desde a sua nomenclatura, passando pela descrição dos classificadores utilizados, até encontrando a justificativa para o uso das árvores de decisão hierárquica.

3.1 Base de dados para emoção em Voz Berlin e terminologia adotada

A base de dados de emoção em voz utilizada nesta pesquisa foi a base de dados Berlin (BURKHARDT et al., 2005). A base é do tipo simulada, composta por cinco falantes de gênero masculino e cinco falantes de gênero feminino que totalizam 535 sinais de áudio gravados a uma taxa de 48kHz e reamostrados a 16kHz, com o tempo de cada sinal duração entre dois a doze segundos. Sete categorias de sentimento são representadas nessa base: Raiva (***A****nger*), Tédio (***B****oredom*), Desgosto (***D****isgust*), Medo (***F****ear*), Felicidade (***H****appy*), Tristeza (***S****ad*) e estado Neutro (***N****eutral*). Ao longo do texto, essas classes de sentimentos são por vezes indicadas pela primeira letra do nome do sentimento em inglês.

A Figura 12 apresenta o mapa de emoções para a base de dados Berlin. Ela se assemelha a Figura 1 apresentado no capítulo de introdução, indicando apenas os sentimentos presentes na base adotada. Observa-se que os sete sentimentos da base estão

distribuídos sobre cinco quadrantes: os quatros indicados na imagem, e um quinto quadrante que é denominado Q0 que apresenta apenas o sentimento neutro. Cada quadrante pode ser expresso também pela combinação de duas entidades que expressam o sentimento: excitação e valência. O eixo horizontal representa a valência de cada um dos sentimentos, de forma que os sentimentos mais a direita representam emoções mais positivas, enquanto que os da esquerda representam emoções mais negativas. O eixo vertical representa a excitação dos sinais, de maneira que quanto mais superior for a posição dos sentimentos, mais "alto" este é considerado enquanto que quanto mais inferior for sua posição mais "baixo" este é. Entende-se por "alto" sentimentos com excitação alta e baixo o oposto.

Figura 12 – Modelo Bidimensional de emoções baseado no circumplexo de afeto de Russell para as emoções da base de dados Berlin

Fonte: Adaptado de Burkhardt et al. (2005)

Denomina-se neste trabalho como "**Tipo de rótulo**" o nome dado a um grupo de classes que representam o sinal de voz. Seis tipos de rótulos foram utilizados para a base de dados Berlin: 1) rótulos principais, que caracterizam os sentimentos categoricamente, como em "raiva" e "tristeza"; 2) rótulos de Excitação; 3) Valência e 4) Quadrante, que caracterizam o sinal de voz quanto a sua posição no modelo circumplexo de afeto (Figura 12); 5) rótulos ADF, que classificam o sinal em *"Anger"*, *"Disgust"* e *"Fear"*, se o sinal for previamente classificado como pertencente ao quadrante 2 (Q2); e 6) rótulos BS, que classificam o sinal em *"Boredom"* e *"Sadness"*, se o sinal for previamente classificado como pertencente ao quadrante 3 (Q3). A justificativa para existência de classificadores com

rótulos do tipo ADF e BS está apresentada na Seção 3.4. A lista abaixo apresenta as classes pertencentes a cada tipo de rótulo.

1. Rótulos principais: ***A**nger, **B**oredom, **D**isgust, **F**ear, **H**appiness, **S**adness, **N**eutral*
2. Rótulos de Excitação: Alta (**H**), Neutra (0), Baixa (**L**)
3. Rótulos de Valência: Positiva (+), Neutra (0), Negativa (−)
4. Rótulos de Quadrante: **Q1**, **Q2**, **Q3**, **Q4** e **Q0** (neutro)
5. Rótulos ADF: **A**, **D**, **F**
6. Rótulos BS: **B**, **S**

A Tabela 1 apresenta como cada sentimento discreto está relacionado com as classes dos demais tipos de rótulos. Por exemplo, o sentimento "*Anger*" está relacionado ao nível de excitação alta (H), valência negativa (−) e ao quadrante 2 (Q2). Destaca-se que o quadrante Q4 não apresenta emoção representada na base de dados Berlin. Q0 refere-se ao sentimento neutro. A tabela ainda apresenta a quantidade de amostras registradas para cada sentimento. Reparar que a base de dados Berlin é desbalanceada, ou seja, a quantidade de elementos de cada classe apresentam quantidades bem discrepantes (observar por exemplo a quantidade de elementos de "Anger" e "Disgust"). Esse fato é importante para justificar alguns resultados encontrados.

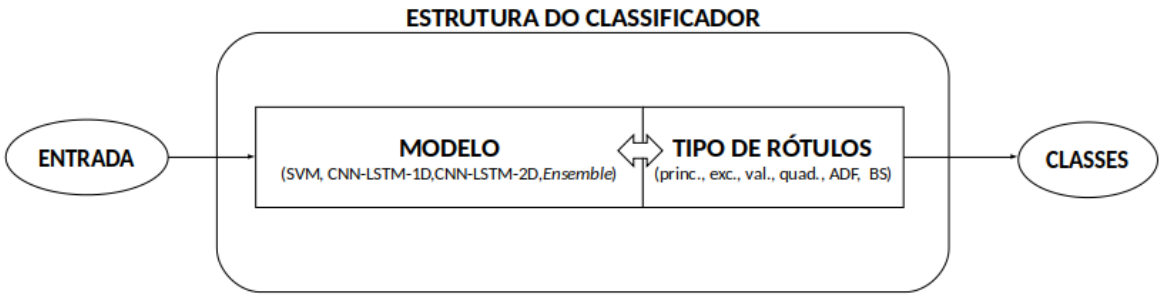
Tabela 1 – Rótulos dos sentimentos categorizados individualmente, por excitação, valência e por quadrantes e quantidade de amostras de cada classe

	Sentimento	Categóricos	Excitação	Valência	Quadrantes	ADF	BS	Qnt.
0	Anger	A	H	(-)	Q2	A	-	127
1	Boredom	B	L	(-)	Q3	-	B	81
2	Disgust	D	H	(-)	Q2	D	-	46
3	Fear	F	H	(-)	Q2	F	-	69
4	Happiness	H	H	(-)	Q1	-	-	71
5	Sadness	S	L	(+)	Q3	-	S	62
6	Neural	N	0	(0)	Q0	-	-	79

3.2 Classificadores

Quatro tipos de classificadores foram utilizados neste trabalho para identificação das emoções dos sinais de voz: SVM (1), CNN-LSTM-1D (2), CNN-LSTM-2D (3) e o *Ensemble* (4) destes classificadores, sendo estes apresentados e descritos em maior detalhamento no Capítulo 2. Cada um desses classificadores pode ser treinado pelos seis tipos de rótulos

Figura 13 – Estrutura genérica de um classificador



Fonte: Autor

citados: rótulos principais, de excitação, valência, quadrantes, rótulos ADF e BS. A Figura 13 apresenta a estrutura genérica para os classificadores realizado nesta pesquisa.

A terminologia empregada para descrever os classificadores associa o nome do modelo seguido do nome do tipo de rótulo sob o qual ele foi treinado. Por exemplo, um classificador CNN-LSTM-1D-valência utiliza a arquitetura CNN-LSTM-1D para categorizar os sentimentos quanto a sua valência (positiva, neutra, negativa). Como outro exemplo, um classificador *Ensemble*-principais combina os resultados de CNN-LSTM-1D-principais, CNN-LSTM-2D-principais e SVM-principais para rotular o sinal em dados categóricos.

Observando ainda a Figura 13, verifica-se que 24 modelos (4 classificadores $\times$ 6 tipos de rótulos) foram treinados para predição de diferentes classes de sentimento. Os quatro modelos treinados sob rótulos principais retornam a classes categóricas de sentimentos, enquanto os modelos treinados sobre rótulos de excitação retornam classes que são “alta”,“neutra” ou “baixa” e assim por diante. Observa-se que os modelos treinados sob rótulos “BS” retornam apenas classes que são “B” ou “S” e enquanto os modelos treinados sob os rótulos “ADF” retornam apenas valores “A”, “D” ou “F”.

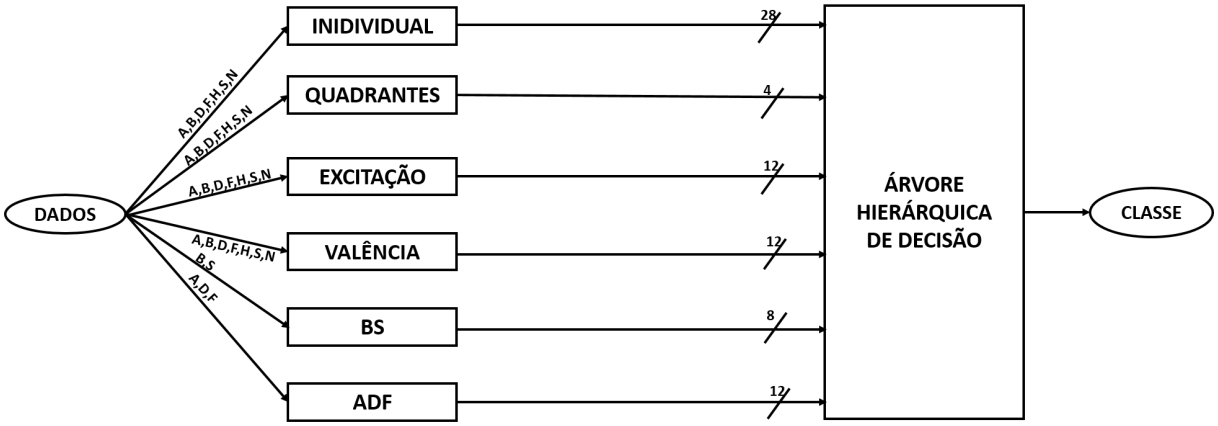
A métrica utilizada nesta pesquisa para avaliar o desempenho de cada classificador foi a F1-*score*, que consiste da média harmônica entre a precisão e o *recall* de cada classe, conforme visto na Seção 2.6. Vale aqui ressaltar que a métrica é calculada para cada uma das classes do modelo e também para o classificador como um todo. Uma classificação ótima é aquela que apresenta valor F1 unitário para cada classe, o que significa que todas as amostras foram classificadas corretamente. A métrica F1 para o classificador geral consiste na média aritmética das métricas individuais de cada classe.

Como será visto na Seção 3.3, a métrica F1 foi a escolhida para encontrar a melhor sequência de construção das folhas das árvores. Com isso, é necessário detalhar melhor a quantidade de valores F1 gerados.

A Figura 14 ilustra uma representação simplificada da geração dos resultados F1

dos classificadores antes de passar pela árvore de decisão, onde o bloco indicado como dados representa os dados de treino dos modelos de classificação. A entrada da estrutura apresentada é um conjunto de dados de sinal de voz que vai ser passado aos algoritmos treinados para classificar os dados em seis tipos de rótulos: principais, excitação, valência, BS, ADF e quadrantes (ver Tabela 1). Cada bloco recebe os dados de entrada e rotula o sinal através de quatro tipos de classificadores: SVM, CNN-LSTM-1D, CNN-LSTM-2D e *Ensemble*. Importante reparar na Figura 14 que há exceções nos blocos BS e ADF, pois esses recebem apenas os dados com sentimentos do terceiro e do quarto quadrante, respectivamente. O traço indicado na saída de cada classificador aponta para o número de valores F1 encontrados para cada tipo de rótulo. Após o treino, cada bloco vai gerar uma quantidade de valores F1 referentes ao número de classes e dos quatro classificadores testados. Por exemplo, os classificadores de rótulos principais geram juntos 28 valores F1 pois cada um dos quatro classificadores (CNN-LSTM-1D, CNN-LSTM-2D, SVM, *Ensemble*) é treinado com sete classes. A Tabela 2 resume o exposto e mostra que há seis tipos de rótulos de sentimentos apresentando 22 tipos de classes sendo avaliadas por meio de 24 classificadores e encontrou-se assim 88 valores de F1, resultantes do treino de vários classificadores para vários tipos de rótulos.

Figura 14 – Estrutura do método proposto. Em uma primeira etapa o sinal é classificado em diferentes tipos de rótulos por quatro tipos de classificadores. Na segunda etapa, as saídas dos classificadores são avaliados por uma árvore de decisão.



fonte: Autor

Ainda com base na Figura 14, aponta-se que o conjunto de valores F1 vai ser passado para o algoritmo de árvore de decisão hierárquica, que define a sequência de classificadores, entre todos treinados, que melhor classifica os dados em rótulos individuais. A sequência de classificadores na árvore de decisão foi escolhida com base nos valores F1 obtidas para cada classe e por cada classificador. Dados de teste são então passados pelos modelos sequenciados na árvore de decisão hierárquica. Observa-se que os dados de entrada passam por uma sequência de modelos intermediários que classificam os dados em

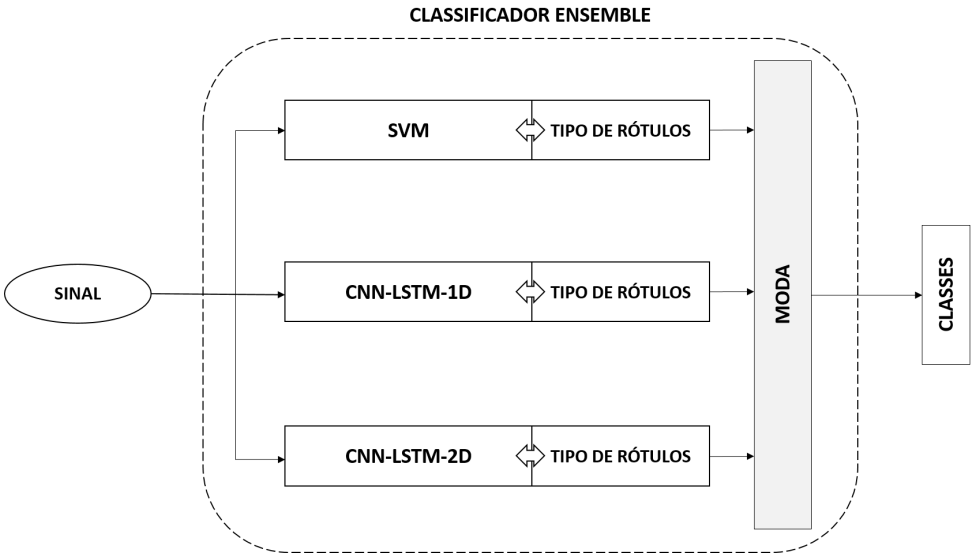
Tabela 2 – Quantidade de classes, número de classificadores utilizados e quantidade de valores F1 encontrados por tipo de rótulo

Tipo de Rótulo	**Qnt. Classes**	**N° Classificadores Utilizados**	**Qnt. Valores F1 Encontrados**
Principais	7	4	28
Excitação	3	4	12
Valência	3	4	12
Quadrantes	4	4	16
BS	2	4	8
ADF	3	4	12
	22	24	88

diferentes tipos de rótulos, até que eles sejam, nas extremidades da árvore, classificados em rótulos principais.

Neste trabalho, para realizar a análise Multi-Características do sinal (ver Seção 1.3), o *Ensemble* das técnicas de SVM, CNN-1D, CNN-2D foi realizado num esquema de classificação de maioria de votos de forma que o resultado do método consiste do sentimento mais encontrado pelos três classificadores (a moda dos valores). A Figura 15 ilustra o esquema do Ensemble. A hipótese do autor é que seu desempenho seria no mínimo igual ao melhor desempenho obtido entre dos três classificadores.

Figura 15 – Modelo Ensemble de arquiteturas propostas para análise Multi-Características



Fonte: Autor

3.3 Metodologia para construção de Árvores de Decisão Hierárquica

A metodologia utilizada para combinar os resultados dos classificadores foi criar uma árvore de decisão hierárquica binária, detalhado na Seção 2.4. Cada nível da árvore utiliza as saídas de algum modelo de classificação para separar uma classe das demais dentro de um tipo de rótulo. O termo binário é devido pois cada nível estabelece se um dado é "sim" ou "não" pertencente a uma determinada classe estabelecida. Como estratégia de divisão em cada folha da árvore, decidiu-se separar a classe de maior valor F1 entre das demais possíveis no tipo de rótulo.

Para exemplificar, supõe-se que um classificador SVM-excitação seja escolhido como nível de separação de uma folha. Supondo ainda que os valores F1 encontrados nesse classificador para cada classe foram 0.9 para excitação "alta", 0.8 para "neutra" e 0.7 para "baixa". A classe "alta" é a de maior F1, então o nível de decisão vai supor em hipótese verdadeira que o sentimento em questão é de excitação "alta", se o classificador SVM-excitação predizer essa mesma classe e o nível de decisão vai supor em hipótese contrária que o sentimento em questão é de excitação "não-alta", se o classificador SVM-excitação predizer "baixo" ou "neutro". Em termos mais abstratos, diz-se que esse nível de decisão "separa" a excitação "alta" das demais classes "baixa" e "neutra".

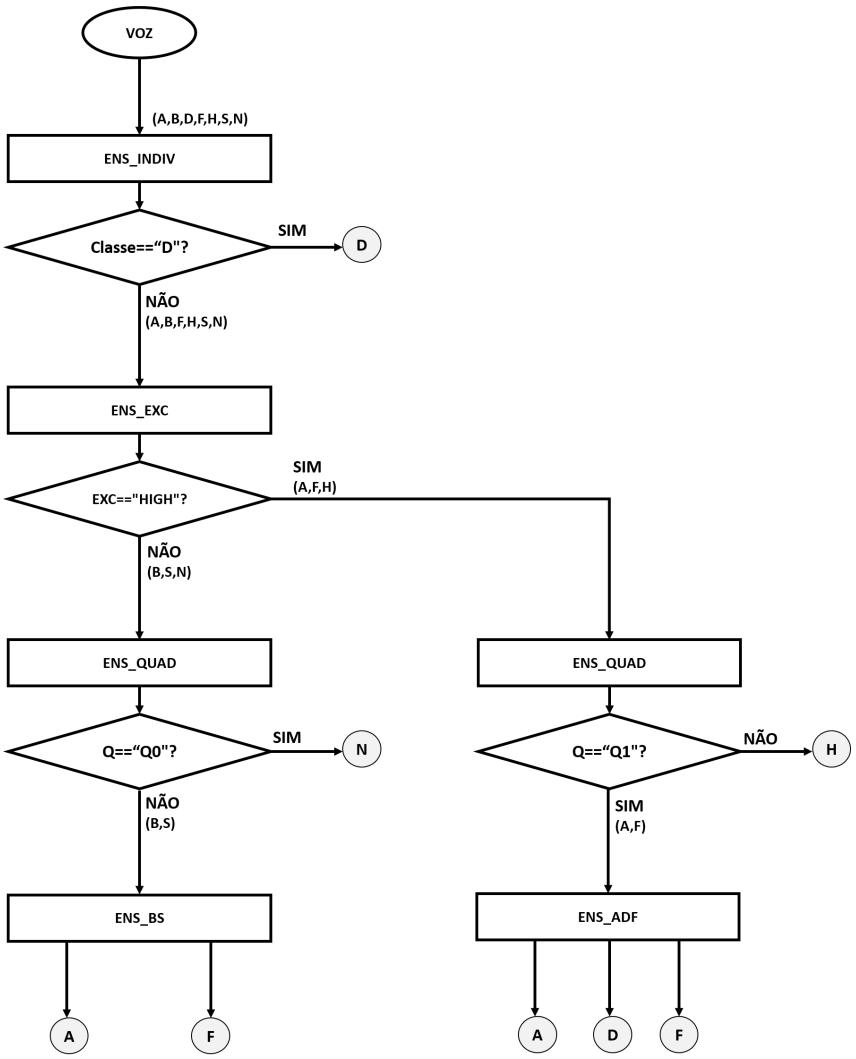
O objetivo do algoritmo da árvore de decisão hierárquica foi escolher a sequência de classificadores que melhor separasse uma classe das demais em um mesmo tipo de rótulo. A cada divisão, o conjunto de sentimentos discretos associados a cada classe vai ser divido também. No exemplo anterior, a divisão do sentimento em "alto" ou não-"alto", fez com que se separasse os sentimentos discretos H,A,D,F (que são "altos") em um ramo e os sentimentos B e S ou N em outro ramo. Cada novo ramo vai procurar um novo classificador que consiga separar os sentimentos discretos separados pelo ramo anterior. A árvore para de crescer quando suas extremidades classificam apenas sentimentos discretos categóricos.

Os 88 valores de F1 encontrados pelos classificadores são passados para a árvore e o algoritmo vai escolher o maior resultado F1 encontrado entre todos. Identifica-se qual classificador e qual tipo de rótulo tiveram maior desempenho. Essa informação é usada para construir o primeiro nó de decisão. O classificador utilizado para fazer a decisão é descartado e o procedimento é repetido para encontrar outro nó de decisão abaixo desse.

A Figura 16 apresenta um esquema de árvore construída com os dados de validação dessa pesquisa. Entres os 88 valores de F1 encontrados, o maior valor foi encontrado para a classe D obtida pelo classificador SVM-excitação. Portanto, o primeiro nível de decisão da árvore vai utilizar esse classificador para dizer se o sentimento de entrada é D (hipótese afirmativa) ou se não é (hipótese contrária). Do segundo nível da árvore em diante, o algoritmo vai procurar o maior resultado F1 entre as classes que ainda não

foram classificadas. No exemplo da Figura 16, após separar D, o resultado de maior F1 encontrado foi “ALTA” previsto pelo classificador *Ensemble*-excitação. Portanto, este classificador foi utilizado para predizer se um sentimento é “ALTA” em hipótese afirmativa ou se não é “ALTA” em hipótese contrária. Analisando a hipótese afirmativa, apenas os sentimentos “A”,“F” e “H” são “ALTOS”. Dentre os classificadores que rotulam essas classes, o classificador *Ensemble*-quadrantes é o que melhor separa as mesmas, e esse esquema de busca e separação acontece até o ponto em que cada nó final da árvore classifique sentimentos categóricos. Isso faz com que, automaticamente, um sentimento classificado como excitação, valência e quadrantes sempre tenham que passar por outros classificadores para “separar” as emoções pertencentes a seus quadrantes. Caso haja empate nos desempenhos de F1, priorizou-se utilizar aqueles classificadores que consegue separar a maior quantidade de sentimentos discretos.

Figura 16 – Exemplo de Árvore Hierárquica de Decisão para classificação de sentimentos



Fonte: Autor

O objetivo final da árvore é classificar os dados em sentimentos principais e o

processo de construção da árvore termina quando todos as folhas das árvores terminarem em algum desses rótulos. Isso faz com que, automaticamente, um sentimento classificado como excitação, valência e quadrantes sempre tenham que passar por outros classificadores para "separar" as emoções pertencentes a seus quadrantes.

Os dados do conjunto de dados foram divididos em conjuntos de treino, validação e teste. Os dados de treino foram utilizados no treinamento dos classificadores. Os dados de validação foram utilizados para calcular os desempenhos F1 para cada classe dos classificadores e criar a estrutura da árvore. Uma vez que a estrutura da árvore foi construída, dados de testes foram testados na estrutura proposta para assim mensurar o seu desempenho.

3.4 Classificadores ADF, BS e mudança nos classificadores de Quadrantes

Os classificadores com os tipos de rótulos ADF e BS foram construídos para separar as classes pertencentes somente aos quadrantes Q2 e Q3, e eles nunca são usados antes da árvore rotular o sentimento a um desses quadrantes. Estes classificadores foram utilizados especificamente para aprimorar os resultados da árvore, isso pois, uma vez que se classifica um sentimento pertencente a Q2 ou Q3, é necessário um modelo sequente para classificar os sentimentos nestes quadrantes.

No caso apresentado pela Figura 16, quando o sentimento é classificado, por exemplo, como quadrante Q2, é necessário que um classificador separe "B" e "S". Daí vem a necessidade de construir classificadores que separam sentimentos pertencentes ao mesmo quadrante de emoção. Os classificadores principais poderiam ser utilizados para a mesma finalidade. Contudo, experimentalmente verificou-se que se torna mais assertivo utilizar modelos treinados apenas com os tipos de sentimentos pertencentes aos quadrantes ao usar modelos que contemplam o treino de todos os tipos de sentimentos.

3.5 Mudança de metodologia para outras bases de dados

A metodologia proposta nesta pesquisa foi construída com os dados existente na base de dados de voz Berlin. Para aplicar a metodologia a alguma outra base, algumas modificações podem ser necessárias. Os próximos parágrafos exemplificam as alterações necessárias para aplicação da metodologia sobre outras bases de sentimento em voz.

A base RAVDESS (LIVINGSTONE; RUSSO, 2018) é uma base simulada, constituída de 7356 sentenças de áudio, gravada por 24 atores expressando oito estados de sentimentos: neutro, calmo, feliz, triste, raiva, medo, desgosto e surpresa. O sentimento

"calmo" é caracterizado como de baixa excitação e de baixa valência, pertencente ao quarto quadrante de sentimentos. Diferente da Base Berlin que possui sentimentos em três quadrantes mais o neutro, para se aplicar a metodologia proposta nesta pesquisa na base de sentimento em voz RAVDESS, deve-se levar em conta, portanto, mais um quadrante representado pelo sentimento "calmo".

A base de Santiago-Omar Caballero-Morales (2013 apud SWAIN; ROUTRAY, 2018) é também simulada, e nas 233 sentenças de áudio quatro sentimentos são representados: felicidade (primeiro quadrante), raiva (segundo quadrante), tristeza (terceiro quadrante) e o estado neutro (quadrante zero). Todos os sentimentos dessa base estão localizados em quadrantes de afeto distintos, de maneira que rotular os sentimentos pelo tipo de rótulo individual tem o mesmo efeito que rotular por quadrantes. Com isso, ao aplicar a metodologia proposta da pesquisa nessa base, se deve levar em conta que a classificação em rótulos de quadrantes é desnecessária.

A base Quiros-ramirez et al. (2014 apud SWAIN; ROUTRAY, 2018) é natural e nela as sentenças de áudio são classificadas apenas como positivas ou negativas. Ou seja, apenas rótulos de sentimento em excitação são presentes. Essa, portanto, não seria uma base apropriada para aplicar a metodologia apresentada em questão, pois não apresenta classificações intermediárias de sentimentos e tampouco os sentimentos são rotulados em classes principais.

3.6 Materiais e recursos utilizados

Todo o trabalhado foi desenvolvido no laboratório Cisne, parte do Programa de Pós-graduação da Universidade Federal do Espírito Santo - UFES. As configurações de máquina utilizada neste trabalho foram:

1. Sistema operacional Linux, distribuicão Ubuntu 18.04 LTS;
2. Placa mãe: Z370M AORUS Gaming-CF;
3. RAM: 32GB DDR4 2133 MHz;
4. CPU: Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz;
5. GPU: NVIDIA Titan V;
6. HD: SSD Corsair Force LE 240GB + HD 2TB;

Todos os códigos foram desenvolvidos pela linguagem Python 3.0. A construção de redes CNN-LSTM em específico foram construídas pela API Keras. Para construção do classificador SVM em Python foi utilizada a biblioteca LIBSVM (CHANG; LIN, 2013).

3.7 Metodologia para construção dos classificadores

Esta seção explora como os dados foram tratados para realização do treinamento dos classificadores. Detalhes de como os dados foram estratificados em treino, validação e teste e como os modelos foram validados são apresentados no Capítulo 4.

3.7.1 Treinamento de classificadores SVM

Conforme metodologia adotada pelo artigo de Shen, Changjun e Chen (2011) as etapas para construção do modelo de SVM foram: de cada uma das 535 amostras de áudio da base de dados, extrai-se um vetor de características do sinal de voz. Parte deste conjunto de vetores é então utilizado para treino do classificador SVM. O kernel de melhor desempenho encontrado foi Kernel Linear. Após o treino, o modelo construído é apto a ser validado e testado pelo outro conjunto de dados.

O método e as características extraídas do sinal de voz são baseados no artigo de Shen, Changjun e Chen (2011). Abaixo encontra-se a descrição dos cálculos realizados sobre o sinal de voz para gerar o conjunto de características. A Tabela 3 resume a descrição e mais detalhes sobre as técnicas estão expostas no trabalho dos autores. Conforme ainda aponta o artigo, o uso de algumas características pode ser suprido para melhor desempenho do classificador SVM. Visto isso, nesta pesquisa utilizou-se as características dos grupos 0, 2 e 4 da Tabela 3, de maneira que os dados de entrada do SVM são formados por vetores de 123 dimensões. Uma análise foi realizada para verificar que o desempenho das classificações realizadas com esses grupos de características é a melhor combinação de entrada, e isso foi verificado. Quanto ao número de classes utilizados no classificador, essa depende do tipo de rótulo da classificação.

1. **Energia**: sobre o sinal de energia do sinal cálcula-se: calcula-se: máximo, média e variância da Energia; máximo, média e mediana da duração da taxa de subida/descida de Energia; Faixa de Interquartil da taxa de subida/descida de Energia; Faixa de Interquartil da duração da subida/descida de Energia. Totaliza-se, portanto, 19 características.

2. ***Pitch***: sobre o *pitch* do sinal, calcula-se: máximo, média e variância da Energia; máximo, média e mediana da duração da taxa de subida/descida do *pitch*; Faixa de Interquartil da taxa de subida/descida do *pitch*; Faixa de Interquartil da duração da subida/descida do *pitch*. Totaliza-se, portanto, 19 características.

3. **LPCC**: de cada frame do sinal, são calculados os coeficientes LPCC de 12 primeiras ordens. O vetor média, máximo, mínimo e variância desses coeficientes entre os frames formam um vetor de 48 dimensões.

4. **MPCC**: de cada frame do sinal, são calculados os coeficientes MPCC de 13 primeiras ordens. O vetor média, máximo, mínimo e variância desses coeficientes entre os frames formam um vetor de 52 dimensões.

5. **LPCMCC**: de cada frame do sinal, são calculados os coeficientes LPCMCC de 14 primeiras ordens. O vetor média, máximo, mínimo e variância desses coeficientes entre os frames formam um vetor de 56 dimensões.

Tabela 3 – Descrição das características estatísticas extraída do sinal de voz

Grupo	**Propriedade**	**Núm. Caracter.**
0	Energia	19
1	Pitch	19
2	LPCC	48
3	MFCC	52
4	LPCMCC	56
Total de características:		194

Fonte: Shen, Changjun e Chen (2011)

3.7.2 Treinamento de classificadores CNN-LSTM-1D e CNN-LSTM-2D

Para construção dos modelos de redes CNN-LSTM-1D utilizados neste trabalho, a seguinte metodologia foi adotada: de cada amostra de áudio do conjunto de dados, uma janela de tamanho de dois segundos foi centralizada sobre o sinal de forma que um conjunto de vetores de tamanho $48kHz \times 2s = 96000$ é extraído. Parte deste conjunto de vetores é então utilizada para treinar o classificador CNN-1D conforme a arquitetura apresentada na Seção 2.3, enquanto que outras são usadas para validação e teste. A parte convolucional da estrutura é responsável pela extração de características do sinal, enquanto a parte LSTM tem a finalidade de encontrar características temporais.

Quanto a construção dos modelos de redes CNN-LSTM-2D, a metodologia adotada foi: de cada amostra de áudio da base de dados, o espectrograma de cada sinal foi gerado. De todos as imagens de espectro geradas, parte foi utilizado para treinar o classificador CNN-LSTM-2D, conforme arquitetura apresentada na Seção 2.3, enquanto que outras são usadas para validação e teste.

Os espectrogramas gerados seguiram a metodologia proposta por Ververidis e Kotropoulos (2006 apud BADSHAH et al., 2017). Primeiramente, uma pré-ênfase é dada no sinal de voz, passando-se um filtro passa-alta. Sobre o sinal filtrado, uma janela *hamming* de tamanho de $15ms$ desliza com passo de $1ms$ e, sobre cada passo da janela, calculou-se a Transformada de Fourier com tamanho 256 do segmento do sinal. Pela simetria da Transformada, um vetor de tamanho 129 é formado em cada passo, de maneira que, após

a janela percorrer todo o sinal, um conjunto de vetores de tamanho 129 foi formado. Esse conjunto de vetores foi concatenado em matriz de tamanho $129 \times M$, onde M é o número de passos do deslizamento. Cada uma dessas matrizes constitui o espectrograma do sinal de áudio, que foram representados como uma imagem, conforme a Figura 7 ilustra. Como o comprimento do espectrograma varia com o tamanho do sinal de áudio, um valor de $M = 135$ foi estabelecido baseado no espectrograma de menor comprimento para fixar o comprimento entre todas as imagens. Com isso, selecionou-se apenas as 135 colunas centrais de cada espectrograma, de modo que todos as imagens passaram a ter dimensão fixa de 120×135.

Tanto para CNN-LSTM-1D quanto para CNN-LSTM-2D, dados de validação foram utlizados como critério de parada do treino. Ao final de cada época de treinamento, o desempenho do modelo é calculado também sobre o conjunto de dados de validação, sendo que nenhum desses dados estão presentes no conjunto de treino, nem de teste. Os critérios de parada do treinamento do modelo foram monitorados em cima do valor da perda (entropia cruzada) do conjunto de validação. Três foram os critérios: (1) limite máximo de épocas; (2) persistência, ou paciência, que é o número de épocas seguidas que o erro de validação fica sem aumentar e (3) delta mínimo, que assume a rede como estável se a diferença do valor monitoriado entre a época atual e a anterior é menor que um valor mínimo estabelecido. Os valores para critérios utilizados foram 45 número de épocas máxima, persistência de oito e delta mínimo de 0.05.

O tamanho da última camada de cada uma das redes foi adaptado para o tipo de rótulo que cada rede foi treinada. De maneira que, para redes treinadas com rótulos principais, excitação, valência, quadrantes, ADF, e BS o número de neurônios nas camadas finais foram, respectivamente, sete, três, três, quatro e dois.

Vale ressaltar aqui algumas experimentações que foram realizadas antes da realização dos testes finais. A primeira é observar que os modelos de rede foram testados com as topologias seguidas nos artigos citados. Contudo, como experimentação, foi realizado o uso de Redes Convolucionais sem a camada de LSTM. O resultado de classificação foi inferior daquele encontrado com o uso da camada LSTM, sendo que o mais discrepante foi na CNN-1D-LSTM. Além disso, muitos modelos apresentaram oscilação da curva de aprendizado. Devido à grande variedade de tipos de modelos testados, não foi objetivo desta pesquisa realizar a otimização de cada aprendizado específico. Parte-se do princípio que se um modelo está mal comportado, ele automaticamente não será escolhido pela árvore de decisão devido aos seu desempenho baixo. Em trabalhos futuros, sugere-se aplicar técnicas de regularização aplicadas a cada modelo individualmente.

Além disso, cabe observar que os dados foram estratificados em treino, validação e teste, mas não houve separação exclusiva de indivíduos entre esses grupos. Ou seja, os modelos não são descriminantes para os indivíduo, o que significa que tanto treino,

validação e teste possuem informação sobre todos os indíviduos. Quanto a distinção de individuos, muitos autores sugerem a separação da testagem em sexo masculino e feminino.

4 Resultados

Para discussão dos resultados nesta seção, as Tabelas 1 e 2 são reapresentadas (Tabelas 4 e 5, respectivamente) com os tipos de classificadores utilizados durante a pesquisa, bem como a distribuição dos tipos de rótulos e as quantidade de amostras da base de dados Berlin.

Tabela 4 – Rótulos dos sentimentos categorizados individualmente, por excitação, valência e por quadrantes e quantidade de amostras de cada classe (página 61)

	Sentimento	Categóricos	Excitação	Valência	Quadrantes	ADF	BS	Qnt.
0	Anger	A	H	(-)	Q2	A	-	127
1	Boredom	B	L	(-)	Q3	-	B	81
2	Disgust	D	H	(-)	Q2	D	-	46
3	Fear	F	H	(-)	Q2	F	-	69
4	Happiness	H	H	(-)	Q1	-	-	71
5	Sadness	S	L	(+)	Q3	-	S	62
6	Neutral	N	0	(0)	Q0	-	-	79

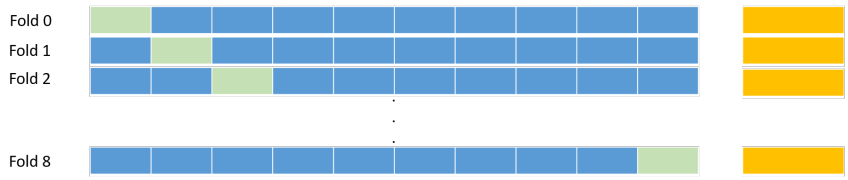
Tabela 5 – Quantidade de classes, número de classificadores utilizados e quantidade de valores F1 encontrados por tipo de rótulo (página 64)

Tipo de Rótulo	**Qnt. Classes**	**N° Classificadores Utilizados**	**Qnt. Valores F1 Encontrados**
Principais	7	4	28
Excitação	3	4	12
Valência	3	4	12
Quadrantes	4	4	16
BS	2	4	8
ADF	3	4	12
	22	24	88

Para avaliar a capacidade de generalização do método proposto, o método de validação cruzada k-*fold* foi utilizada. O conjunto com as 535 amostras de áudio da base de voz Berlin foi estratificado, ou seja, dividido mantendo as proporções das classes, em dez conjuntos. Um desses conjuntos foi separado como “conjunto de dados de teste”, de maneira que estas amostras não foram utilizadas em nenhuma etapa de treino e tampouco na construção de nenhuma das árvores de decisão hierárquica. Dos nove grupos restantes, um *fold* foi utilizado como conjunto de dados de validação enquanto os oito restantes foram agrupados em um único conjunto de dados para treinamento dos classificadores.. Este procedimento foi repetido nove vezes. Dessa forma, nove *folds* foram construídos, cada um possuindo um grupo distinto de dados de validação, sendo que todos os modelos

construídos foram testados sobre o mesmo conjunto de teste. A Figura 17 ilustra o exposto e nela, as cores azul, verde e amarelo representam os subconjuntos de treino, validação e teste, respectivamente, onde o conjunto de teste possui as mesmas amostras em todos os *folds*. A Tabela 6 apresenta a quantidade de amostras de treino, validação e teste para cada *fold*.

Figura 17 – Método de estratificação de dados em conjuntos de treino, validação e teste para nove *folds*



Fonte: Autor

Tabela 6 – Quantidade de amostras de treino, validação e teste para cada *fold*

	Quantidade de amostras		
fold	Treino	Val.	Teste
0	421	55	59
1	422	54	59
2	423	53	59
3	423	53	59
4	424	52	59
5	425	51	59
6	426	50	59
7	427	49	59
8	428	48	59

Para melhor explicação dos resultados obtidos, as seguintes seções discorrem sobre os resultados obtidos em validação, teste e árvore separadamente.

4.1 Resultados de Validação

Em cada *fold*, os classificadores CNN-1D, CNN-2D, SVM foram treinados para cada um dos seis tipos de rótulos, totalizando assim 18 treinamentos com 18 resultados. Seis novos classificadores são formados quando realizado o *Ensemble* dos resultados para cada tipo de rótulo, totalizando 24 resultados. Os classificadores com rótulos BS e ADF foram treinados e validados apenas com as amostras rotuladas nos sentimentos B, S e A, D, F respectivamente. Os dados de validação foram aplicados sobre os 24 modelos e obtidos 88 valores F1. A Tabela 7 apresenta a média e o desvio padrão dos 88 valores F1 encontrados para cada classe na validação do modelo em cada *fold*. Os valores em negritos na Tabela apontam o melhor desempenho entre os classificadores.

Tabela 7 – Média e desvio padrão do desempenho F1 encontrado para cada classe entre todos os *folds*

		Classificadores - Média (std)			
Tipo	Classes.	CNN_1D	CNN_2D	SVM	ENS
ADF	A	**0.90 (0.04)**	0.75 (0.11)	0.90 (0.06)	0.84 (0.07)
	D	0.49 (0.27)	0.20 (0.40)	**0.64 (0.22)**	0.41 (0.37)
	F	**0.67 (0.11)**	0.16 (0.33)	0.65 (0.22)	0.60 (0.18)
	F1-ponderado	0.76 (0.09)	0.48 (0.22)	**0.78 (0.11)**	0.69 (0.13)
BS	B	0.83 (0.23)	0.85 (0.15)	0.89 (0.08)	**0.92 (0.05)**
	S	0.84 (0.13)	0.78 (0.25)	0.86 (0.10)	**0.88 (0.06)**
	F1-ponderado	0.84 (0.17)	0.82 (0.19)	0.88 (0.08)	**0.90 (0.05)**
exc	H	0.89 (0.04)	0.38 (0.09)	**0.92 (0.04)**	0.86 (0.03)
	L	0.62 (0.11)	0.08 (0.17)	**0.85 (0.05)**	0.67 (0.09)
	N	0.00 (0.00)	0.10 (0.17)	**0.64 (0.13)**	0.12 (0.15)
	F1-ponderado	0.68 (0.05)	0.26 (0.09)	**0.86 (0.05)**	0.69 (0.06)
Princip.	A	0.78 (0.08)	0.75 (0.18)	0.78 (0.08)	**0.79 (0.11)**
	B	0.41 (0.16)	0.57 (0.22)	0.61 (0.10)	**0.65 (0.11)**
	D	0.33 (0.18)	**0.60 (0.29)**	0.45 (0.23)	0.56 (0.28)
	F	**0.48 (0.15)**	0.40 (0.28)	0.40 (0.21)	0.47 (0.22)
	H	0.35 (0.10)	0.42 (0.24)	**0.45 (0.14)**	0.42 (0.19)
	S	0.69 (0.20)	0.74 (0.16)	**0.81 (0.12)**	0.78 (0.12)
	N	0.62 (0.09)	0.57 (0.26)	0.65 (0.12)	**0.68 (0.14)**
	F1-ponderado	0.56 (0.07)	0.60 (0.18)	0.62 (0.06)	**0.64 (0.10)**
quad	Q0	0.34 (0.24)	0.09 (0.17)	**0.65 (0.13)**	0.39 (0.10)
	Q1	0.01 (0.03)	0.00 (0.00)	**0.02 (0.07)**	0.00 (0.00)
	Q2	0.74 (0.05)	0.34 (0.11)	**0.79 (0.04)**	0.69 (0.04)
	Q3	0.07 (0.09)	0.09 (0.18)	**0.84 (0.05)**	0.49 (0.15)
	F1-ponderado	0.41 (0.07)	0.19 (0.08)	**0.68 (0.05)**	0.50 (0.06)
val	(-)	0.70 (0.04)	0.71 (0.07)	**0.85 (0.03)**	0.83 (0.03)
	0	0.52 (0.08)	0.08 (0.15)	**0.56 (0.13)**	0.52 (0.14)
	(+)	**0.05 (0.09)**	0.02 (0.06)	0.02 (0.07)	0.03 (0.08)
	F1-ponderado	0.58 (0.04)	0.52 (0.06)	**0.69 (0.04)**	0.68 (0.05)

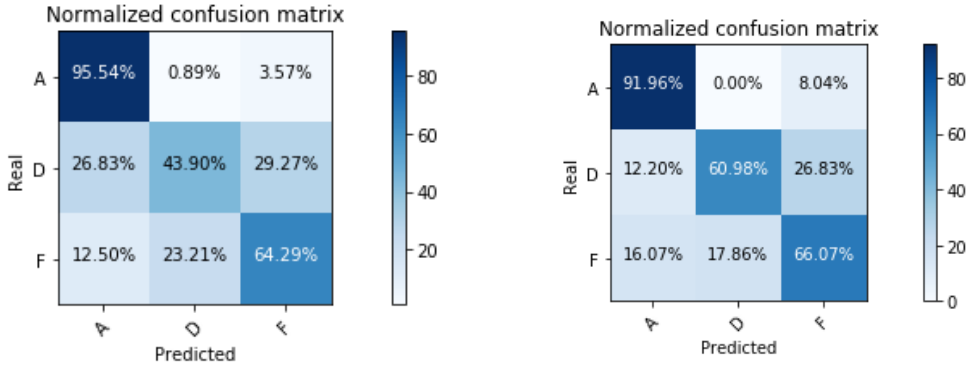
A assertividade de cada um dos classificadores em cada *fold* pode ser representada por uma Matriz de Confusão. Ao longo dos nove *folds*, 465 amostras de validação foram aplicadas aos classificadores, formando assim 465 duplas de valores real-previsto para cada tipo de rótulo. Essas duplas real-previstas foram representadas em uma Matriz de Confusão para cada tipo de rótulo com o intuito de dispor de um valor médio de assertividade dos modelos entre os *folds*.

A análise dos resultados de validação será feita conforme a ordem dos tipos de rótulos apresentados na Tabela 7. Durante os parágrafos que seguem, além da Matriz de Confusão, outras tabelas apresentam as métricas de acurácia, precisão e *recall*, além da F1 para cada classe obtida durante a validação de cada modelo, conforme explicadas na Seção 2.6.

4.1.1 Validação Rótulos ADF

Para classificadores rotulados em ADF, verifica-se que o classificador SVM foi aquele que obteve maior desempenho médio para as três métricas avaliadas, apesar do desempenho do classificado CNN-1D possuir resultados similares e com menor desvio padrão entre os *folds*. A Figura 18 apresenta a Matriz de Confusão para os resultados e a Tabela 8 aponta as métricas para as classes. Observa-se que a classe A possui alto valor de *recall*, indicando pouca presença de falsos negativos na classificação do rótulo, apesar de contar com a presença de muitos falsos positivos. Esse resultado pode ser explicado pelo fato desta classe ser majoritária em relação as demais (proporção da quantidade de rótulos ADF, conforme Tabela 6, é de 127:46:69 entre as classes), podendo ter tendenciado o modelo a ser polarizado para esta classe.

Figura 18 – Matriz de Confusão para classificadores tipo ADF em validação



(a) SVM-ADF (b) CNN-1D-ADF

Fonte: Autor

Tabela 8 – Métricas para classificadores tipo ADF e suas classes em validação

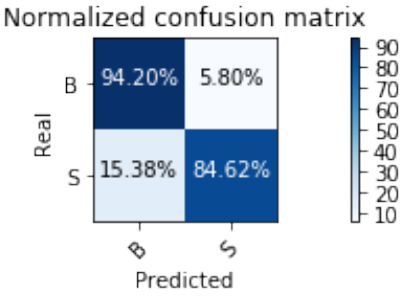
Rótulo.	Classes	CNN_1D				SVM			
		f1-score	precisão	recall	suporte	f1-score	precisão	*recall*	suporte
ADF	A	0.90	0.86	0.96	112	0.90	0.88	0.92	112
	D	0.49	0.56	0.44	41	0.66	0.71	0.61	41
	F	0.67	0.69	0.64	56	0.65	0.65	0.66	56
	acurácia	0.77	0.77	0.77	0	0.79	0.79	0.79	0
	macro	0.69	0.70	0.68	209	0.74	0.75	0.73	209
	ponderado	0.76	0.75	0.77	209	0.79	0.79	0.79	209

4.1.2 Validação Rótulos BS

Para classificadores rotulados em BS verifica-se, conforme Tabela 7, que os desempenhos são muito satisfatórios para todos os tipos de classificadores. O *Ensemble* possui resultado superior em relação a cada classificador individualmente. A Tabela 9 apresenta as métricas para o classificador *Ensemble*-BS, juntamente com a Matriz de Confusão na

Figura 19. Pode-se justificar os bons resultados pela quantidade equilibrada de amostras para classe (proporção BS, conforme Tabela 6, é de 81:62 entre as classes) e também por ser um classificador um-contra-um.

Figura 19 – Matriz de Confusão para classificadores tipo BS em validação



(a) *Ensemble*-BS

Fonte: Autor

Tabela 9 – Métricas para classificadores tipo BS e suas classes em validação

Rótulo.	Classes	*Ensemble*			
		f1-score	precisão	*recall*	suporte
BS	B	0.92	0.89	0.94	69
	S	0.88	0.92	0.85	52
	acurácia	0.90	0.90	0.90	0
	macro	0.90	0.90	0.89	121
	ponderado	0.90	0.90	0.90	121

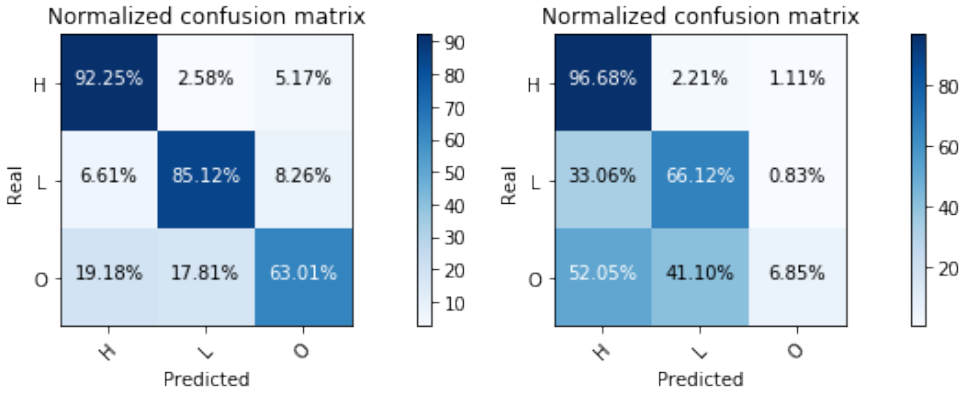
4.1.3 Validação Rótulos Excitação

Para classificadores rotulados em excitação, conforme Tabela 7, verifica-se que o melhor desempenho foi do classificador SVM. O segundo melhor classificador é o *Ensemble*, com um valor de F1 ponderado de 0.69, comparado a 0.85 do primeiro, conforme Tabela 10. O baixo desempenho do *Ensemble* se deve aos classificadores CNN-1D e CNN-2D, que possuíram resultados pouco satisfatórios para classificação da classe neutra. Conforme pode ser visto na matriz da Figura 20, o *Ensemble* possui muitos falsos negativos para classe neutra, fazendo com que o *recall* e o F1 sejam baixos. Um resultado interessante a ser observado pelos modelos de SVM-excitação são os bons valores de precisão e *recall* para a classe "alta". Isso corresponde dizer que modelos de excitação são precisos para classificar os sentimentos "A", "D", "F" e "H" dos demais, demonstrando ser um bom separador de excitação alta de excitação não alta.

4.1.4 Validação Rótulos Principais

Conforme Tabela 7, os classificadores CNN-1D, CNN-2D e SVM possuem na média dos *folds* desempenho similares: 0.55, 0.59, 0.61 de F1-ponderado para média de classes.

Figura 20 – Matriz de Confusão para classificadores tipo Excitação em validação



(a) SVM-excitação (b) *Ensemble*-excitação

Fonte: Autor

Tabela 10 – Métricas para classificadores tipo excitação e suas classes em validação

Rótulo.	Classes	SVM				*Ensemble*			
		f1-score	precisão	*recall*	suporte	f1-score	precisão	*recall*	suporte
exc	H	0.92	0.92	0.92	271	0.86	0.77	0.97	271
	L	0.84	0.84	0.85	121	0.68	0.69	0.66	121
	0	0.64	0.66	0.63	73	0.12	0.56	0.07	73
	acurácia	0.86	0.86	0.86	0	0.75	0.75	0.75	0
	macro	0.80	0.80	0.80	465	0.55	0.67	0.57	465
	ponderado	0.86	0.86	0.86	465	0.69	0.72	0.75	465

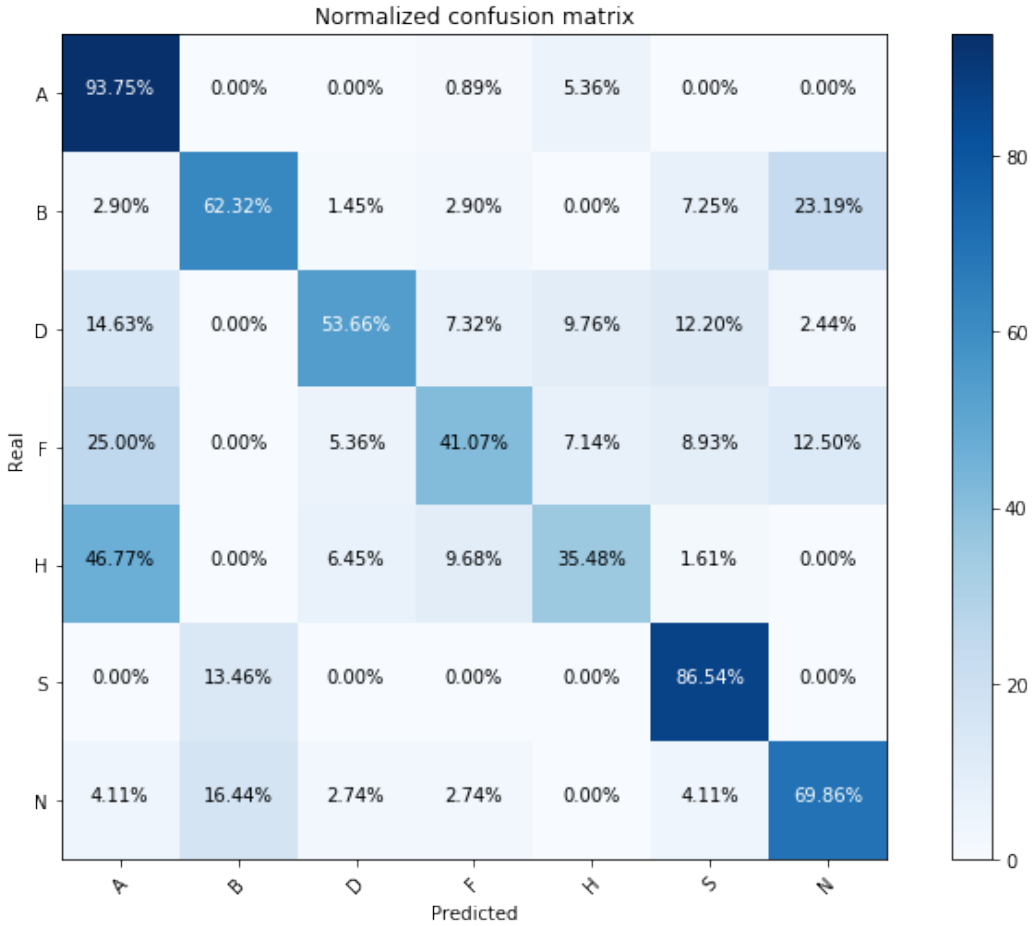
O *Ensemble* dos classificadores foi o classificador de maior desempenho, com média de 0.64 e variância de 0.09 de F1-ponderado. A variância dos resultados F1-ponderado para SVM e CNN-1D é de aproximadamente 0.06, sendo cerca de três vezes inferior aos 0.18 do classificador CNN-2D. Isso pode ser explicado por alguma má configuração de rede, ou pela má inicialização de pesos da rede, ou mesmo por hiperparâmetros que não foram bem ajustados. A Figura 21 e a Tabela 11 apresentam a Matriz de Confusão e as métricas para as classes do classificador *Ensemble*.

Algumas análises interessantes devem ser consideradas dos resultados de classificadores individuais. A primeira é quanto à classificação dos sentimentos pertencentes ao segundo quadrante de sentimentos. Pode-se observar pela matriz da Figura 21 que a classe A possui grande percentual de falsos positivos e baixo percentual de falsos negativos, inferindo em baixa precisão e alto *recall* para classe, conforme apresentado na Tabela 11. Importante observar que as classes que aparecem como falsos positivos para A são D, F e H, que são sentimentos de excitação alta, e que possuem certa similaridade e proximidade no diagrama de sentimentos de Russel. Isso se deve ao fato também da classe A ser majoritária. Classes neutras ou de baixa excitação quase não aparecem como falsos positivos para a classe A. Mesmo com muitos falsos positivos, a classe A é a de maior assertividade.

Situação semelhante ocorre para as classes de sentimento B e S. O sentimento B, como pode ser visto na Matriz de Confusão, possui como falsos positivos as classes S e N, que estão localizadas próximas umas às outras no diagrama de sentimentos e também são sentimentos de valência negativa. A classe S, apesar da sua alta assertividade, possui como falsos positivos outros sentimentos negativos, como D e F, além de B que está localizado em seu mesmo quadrante.

Exemplos como o da classe A, que possuem como falsos positivos sentimentos de excitação também alta, e B e S, que possuem como falsos positivos sentimentos de valência também negativa, motivam e justificam a construção de modelos de classificação de sentimentos em rótulos de excitação e valência, conforme proposta deste trabalho. A análise da Matriz de Confusão por meio das características dos sentimentos e conhecendo suas peculiaridades além de apenas uma análise numérica, enriquece o desenvolvimento de novos classificadores que buscam otimizar os resultados desenvolvidos em testes iniciais.

Figura 21 – Matriz de Confusão para classificadores tipo principais em validação



Fonte: Autor

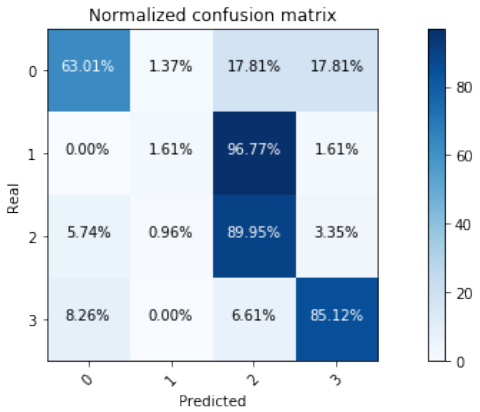
Tabela 11 – Métricas para classificadores tipo Principais e suas classes em validação

Rótulo.	Classes	*Ensemble*			
		f1-score	precisão	*recall*	suporte
indiv.	A	0.77	0.66	0.94	112
	B	0.66	0.69	0.62	69
	D	0.60	0.69	0.54	41
	F	0.49	0.62	0.41	56
	H	0.45	0.61	0.35	62
	S	0.78	0.70	0.87	52
	N	0.69	0.68	0.70	73
	acurácia	0.67	0.67	0.67	0
	macro	0.63	0.67	0.63	465
	ponderado	0.65	0.66	0.67	465

4.1.5 Validação Rótulos Quadrantes

Conforme pode ser visto na Tabela 7, o classificador SVM é o que possui melhor desempenho F1-ponderado para separação dos sentimentos em quadrantes, com média e desvio de 0.67 e 0.05, respectivamente. O segundo melhor desempenho é do classificador *Ensemble*, com 0.49 de média e 0.06 de desvio. A Figura 22 e a Tabela 12 apresentam a Matriz de Confusão e as métricas para esse classificador. Observa-se que a classe quadrante 01, constituída apenas do sentimento H, tem desempenho pouco satisfatório, sendo que praticamente todas as suas amostras foram classificadas como pertencentes ao quadrante 02. Esse resultado é explicado quando se recorre ao diagrama de sentimentos e assume-se que ambos quadrantes 01 e 02 são sentimentos de excitação alta, tendo características próximas de energia. Outro motivo é que a classe quadrantes 03 possui três vezes mais amostras que quadrantes 01, o que desequilibra o treino dos classificadores, podendo polarizar o resultado da classificação.

Figura 22 – Matriz de Confusão para classificadores tipo Quadrantes em validação



(a) SVM-quadrantes

Fonte: Autor

Tabela 12 – Métricas para classificadores tipo Quadrantes e suas classes em validação

Rótulo.	Classes	SVM			
		f1-score	precisão	*recall*	suporte
quad.	Q0	0.65	0.68	0.63	73
	Q1	0.03	0.25	0.02	62
	Q2	0.79	0.70	0.90	209
	Q3	0.84	0.83	0.85	121
	acurácia	0.73	0.73	0.73	0
	macro	0.58	0.61	0.60	465
	ponderado	0.68	0.67	0.73	465

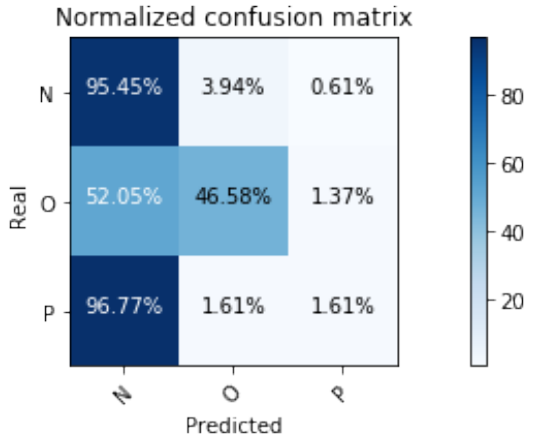
4.1.6 Validação Rótulos Valência

Conforme pode ser visto na Tabela 7, o classificador SVM é o que possui melhor desempenho F1-ponderado para separação dos sentimentos em valência, com média e desvio de 0.69 e 0.03, respectivamente, e o segundo melhor desempenho foi do classificador *Ensemble*, com o valor próximo de 0.67 de média e 0.04 de desvio. Todos os classificadores de valência encontrados possuíram a característica de apresentar altos valores de *recall* para a classe de sentimentos negativos, e baixo valor de precisão e *recall* para classe de sentimentos positivos. Como pode ser visto na Tabela 13, o valor F1 para classe Neutra é de 0.56, contra apenas 0.03 da classe positiva e sentimentos da classe negativa possuem F1 de 0.84. Os altos valores da primeira coluna da Matriz de Confusão da Figura 23 indica forte tendência da classe negativa em assumir falsos positivos.

Existe apenas uma entidade de sentimento de valência positiva, que é H, e também apenas uma entidade de sentimento na classe de valência neutra, representado pelo próprio sentimento N (Neutral). Esses sentimentos são classificados contra todos os outros sentimentos negativos (A, D, F, B e S). A proporção de amostras de sentimentos negativos, positivos e neutros é de 330:73:62, conforme Tabela 6.

Mesmo apresentando quantidades parecidas de amostras, as classes positivas e neutras possuem resultados bem distintos quando classificados contra uma classe de sentimentos com uma maior proporção de amostras que as suas. Esse resultado é interessante de ser explorado em contraponto com o pensamento de que classes de amostras minoritárias tendem a apresentar resultados inferiores quando classificadas contra classes majoritárias. O fato da classe positiva ter apenas 0.03 de F1 contra 0.56 da classe neutra pode ser também explicado pela escolha de características extraídas dos sentimentos, de maneira que os sentimentos neutros foram melhores representados que os sentimentos positivos com as características selecionadas. Dessa forma, a discrepância de resultados pode não ser somente explicada pelo desequilíbrio de classes, mas também pela escolha das características utilizadas para representar os dados.

Figura 23 – Matriz de Confusão para classificadores tipo Valência em validação



(a) SVM-valência

Fonte: Autor

Tabela 13 – Métricas para classificadores tipo Valência e suas classes em validação

Rótulo.	Classes	SVM			
		f1-score	precisão	*recall*	suporte
val	(-)	0.85	0.76	0.95	330
	(0)	0.56	0.71	0.47	73
	(+)	0.03	0.25	0.02	62
	acurácia	0.75	0.75	0.75	0
	macro	0.48	0.57	0.48	465
	ponderado	0.69	0.69	0.75	465

4.2 Resultados de Teste

A Tabela 14 está estruturada da mesma forma que a Tabela 7 e apresenta os 88 valores de F1 obtidos sobre os 24 classificadores aplicados sobre os dados de teste. As demais figuras e tabelas que seguem comparam o desempenho de classificação, para cada tipo de rótulo, dos dados de validação com os dados de teste. As métricas e as Matrizes de Confusão foram calculadas e construídas sobre os resultados de testes da mesma maneira que foram sobre os resultados de validação. Os valores em negritos na Tabela 14 apontam o melhor desempenho entre os classificadores.

Comparando os resultados das Tabelas 14 e 7, verifica-se que os classificadores de melhor desempenho em validação para cada tipo de rótulo são também os melhores para os resultados de teste, com exceção do tipo de rótulo BS, que possui *Ensemble* como o melhor modelo em validação e SVM sobre o conjunto de teste. Quanto a variância dos resultados entre *folds*, verifica-se que os resultados obtidos pelos classificadores SVM em teste são frequentemente os menores. Pode-se notar que a variância dos modelos convolucionais para os classificadores BS e ADF são altos se comparados ao SVM. Isso demonstra, como encontrado na literatura, que modelos convolucionais são mais sensíveis a quantidade de

dados do conjunto de treino se comparados ao classificador SVM, dado que a quantidade de dados de treino usado nesses classificadores é reduzida em relação aos demais.

No mais, pode-se verificar que o desempenho dos classificadores sobre os dados de teste no geral é inferior ao desempenho sobre os dados de validação, o que era esperado. Apenas o classificador BS possuiu desempenho de teste ligeiramente maior que o de validação.

Tabela 14 – Média e desvio padrão do desempenho F1 encontrado para cada classe entre todos os *folds*

		Classificadores - Média (std)			
Tipo	Classes.	CNN_1D	CNN_2D	SVM	ENS
ADF	A	0.87(0.02)	0.75(0.10)	**0.87(0.00)**	0.81(0.06)
	D	0.29(0.24)	0.16(0.31)	**0.67(0.08)**	0.33(0.23)
	F	0.65(0.10)	0.18(0.35)	**0.56(0.10)**	0.46(0.18)
	F1-ponderado	0.70(0.07)	0.48(0.21)	**0.75(0.04)**	0.62(0.10)
BS	B	0.93(0.16)	0.86(0.08)	0.90(0.02)	**0.95(0.05)**
	S	**0.95(0.10)**	0.79(0.15)	0.82(0.06)	0.92(0.09)
	F1-ponderado	0.94(0.14)	0.83(0.11)	0.87(0.04)	**0.94(0.06)**
exc	H	0.86(0.02)	0.30(0.06)	**0.91(0.02)**	0.88(0.03)
	L	0.69(0.05)	0.12(0.12)	**0.83(0.02)**	0.80(0.05)
	N	0.00(0.00)	0.04(0.08)	**0.66(0.10)**	0.51(0.23)
	F1-ponderado	0.74(0.03)	0.21(0.07)	**0.86(0.02)**	0.83(0.04)
Princip.	A	**0.76(0.02)**	0.73(0.11)	0.68(0.02)	0.72(0.08)
	B	0.53(0.09)	0.62(0.24)	0.65(0.02)	**0.75(0.14)**
	D	0.07(0.15)	0.36(0.18)	**0.70(0.06)**	0.42(0.11)
	F	**0.44(0.09)**	0.42(0.25)	0.23(0.11)	0.30(0.20)
	H	0.37(0.09)	0.19(0.13)	**0.42(0.06)**	0.25(0.08)
	S	**0.87(0.04)**	0.81(0.06)	0.79(0.04)	**0.87(0.06)**
	N	0.64(0.06)	0.60(0.28)	0.49(0.09)	**0.71(0.12)**
	F1-ponderado	0.57(0.03)	0.57(0.14)	0.59(0.03)	**0.61(0.08)**
quad	Q0	0.35(0.22)	0.00(0.00)	**0.64(0.09)**	0.45(0.19)
	Q1	0.00(0.00)	0.00(0.00)	**0.02(0.07)**	0.00(0.00)
	Q2	0.70(0.02)	0.30(0.05)	**0.75(0.03)**	0.69(0.04)
	Q3	0.09(0.12)	0.12(0.12)	**0.83(0.02)**	0.56(0.14)
	F1-ponderado	0.37(0.05)	0.17(0.06)	**0.67(0.02)**	0.53(0.07)
val	(-)	0.70(0.03)	0.77(0.04)	**0.87(0.01)**	0.86(0.02)
	0	0.25(0.05)	0.05(0.10)	**0.37(0.19)**	0.36(0.10)
	(+)	0.00(0.00)	**0.04(0.08)**	0.02(0.07)	0.02(0.07)
	F1-ponderado	0.58(0.03)	0.62(0.03)	**0.72(0.02)**	**0.72(0.03)**

Tabela 15 – Comparativo melhor modelo tipo ADF para dados de validação e teste

Rótulo.	Classes	SVM - Valid.				SVM - Test			
		f1-score	precisão	*recall*	suporte	f1-score	precisão	*recall*	suporte
indiv.	A	0.90	0.88	0.92	112	0.87	0.82	0.92	126
	D	0.66	0.71	0.61	41	0.67	0.66	0.69	45
	F	0.65	0.65	0.66	56	0.56	0.67	0.48	63
	acurácia	0.79	0.79	0.79	0	0.76	0.76	0.76	0
	macro	0.74	0.75	0.73	209	0.70	0.71	0.70	234
	ponderado	0.79	0.79	0.79	209	0.75	0.75	0.76	234

Tabela 16 – Comparativo melhor modelo tipo BS para dados de validação e teste

Rótulo.	Classes	*Ensemble* - Valid.				*Ensemble* - Test			
		f1-score	precisão	*recall*	suporte	f1-score	precisão	*recall*	suporte
BS	B	0.92	0.89	0.94	69	0.95	0.91	1.00	108
	S	0.88	0.92	0.85	52	0.93	1.00	0.86	81
	acurácia	0.90	0.90	0.90	0	0.94	0.94	0.94	0
	macro	0.90	0.90	0.89	121	0.94	0.95	0.93	189
	ponderado	0.90	0.90	0.90	121	0.94	0.95	0.94	189

Tabela 17 – Comparativo melhor modelo tipo excitação para dados de validação e teste

Rótulo.	Classes	SVM - Valid.				SVM - Test			
		f1-score	precisão	*recall*	suporte	f1-score	precisão	*recall*	suporte
exc	H	0.92	0.92	0.92	271	0.91	0.91	0.90	306.00
	L	0.84	0.84	0.85	121	0.83	0.89	0.78	189.00
	O	0.64	0.66	0.63	73	0.66	0.52	0.89	36.00
	acurácia	0.86	0.86	0.86	0	0.86	0.86	0.86	0.00
	macro	0.80	0.80	0.80	465	0.80	0.77	0.86	531.00
	ponderado	0.86	0.86	0.86	465	0.86	0.88	0.86	531.00

Tabela 18 – Comparativo melhor modelo tipo Principais para dados de validação e teste

Rótulo.	Classes	*Ensemble* - Valid.				*Ensemble* - Test			
		f1-score	precisão	*recall*	suporte	f1-score	precisão	*recall*	suporte
indiv.	A	0.77	0.66	0.94	112	0.71	0.57	0.96	126
	B	0.66	0.69	0.62	69	0.76	0.94	0.63	108
	D	0.60	0.69	0.54	41	0.42	0.67	0.31	45
	F	0.49	0.62	0.41	56	0.35	0.49	0.27	63
	H	0.45	0.61	0.35	62	0.25	0.36	0.19	72
	S	0.78	0.70	0.87	52	0.87	0.83	0.91	81
	N	0.69	0.68	0.70	73	0.70	0.56	0.94	36
	acurácia	0.67	0.67	0.67	0	0.64	0.64	0.64	0
	macro	0.63	0.67	0.63	465	0.58	0.63	0.60	531
	ponderado	0.65	0.66	0.67	465	0.61	0.65	0.64	531

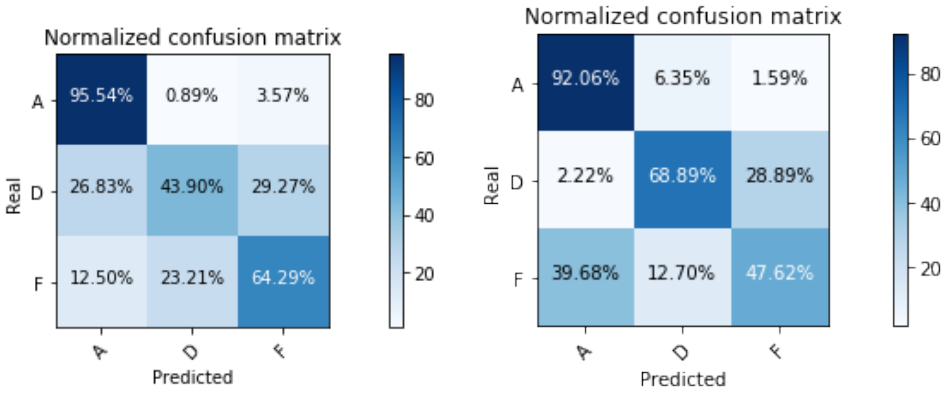
Tabela 19 – Comparativo melhor modelo tipo Quadrantes para dados de validação e teste

Rótulo.	Classes	SVM - Valid.				SVM - Test			
		f1-score	precisão	*recall*	suporte	f1-score	precisão	*recall*	suporte
quad.	0	0.65	0.68	0.63	73	0.64	0.50	0.89	36
	1	0.03	0.25	0.02	62	0.03	0.33	0.01	72
	2	0.79	0.70	0.90	209	0.75	0.67	0.85	234
	3	0.84	0.83	0.85	121	0.83	0.89	0.78	189
	acurácia	0.73	0.73	0.73	0	0.72	0.72	0.72	0
	macro	0.58	0.61	0.60	465	0.56	0.60	0.64	531
	ponderado	0.68	0.67	0.73	465	0.67	0.69	0.72	531

Tabela 20 – Comparativo melhor modelo tipo Valência para dados de validação e teste

Rótulo.	Classes	SVM - Valid.				SVM - Test			
		f1-score	precisão	*recall*	suporte	f1-score	precisão	*recall*	suporte
val.	(-)	0.85	0.76	0.95	330	0.87	0.81	0.94	423
	(0)	0.56	0.71	0.47	73	0.38	0.37	0.39	36
	(+)	0.03	0.25	0.02	62	0.03	0.33	0.01	72
	acurácia	0.75	0.75	0.75	0	0.78	0.78	0.78	0
	macro	0.48	0.57	0.48	465	0.42	0.50	0.45	531
	ponderado	0.69	0.69	0.75	465	0.72	0.72	0.78	531

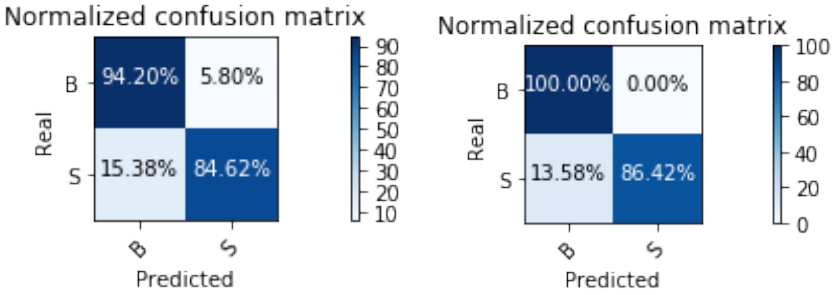
Figura 24 – Comparativo das Matrizes de Confusão SVM-ADF para validação e teste



(a) Validação (b) Teste

Fonte: Autor

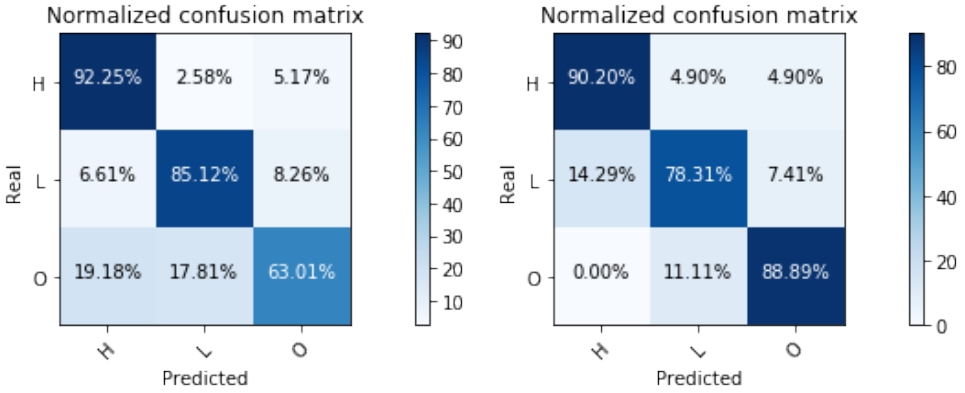
Figura 25 – Comparativo das Matrizes de Confusão *Ensemble*-BS para validação e teste



(a) Validação (b) Teste

Fonte: Autor

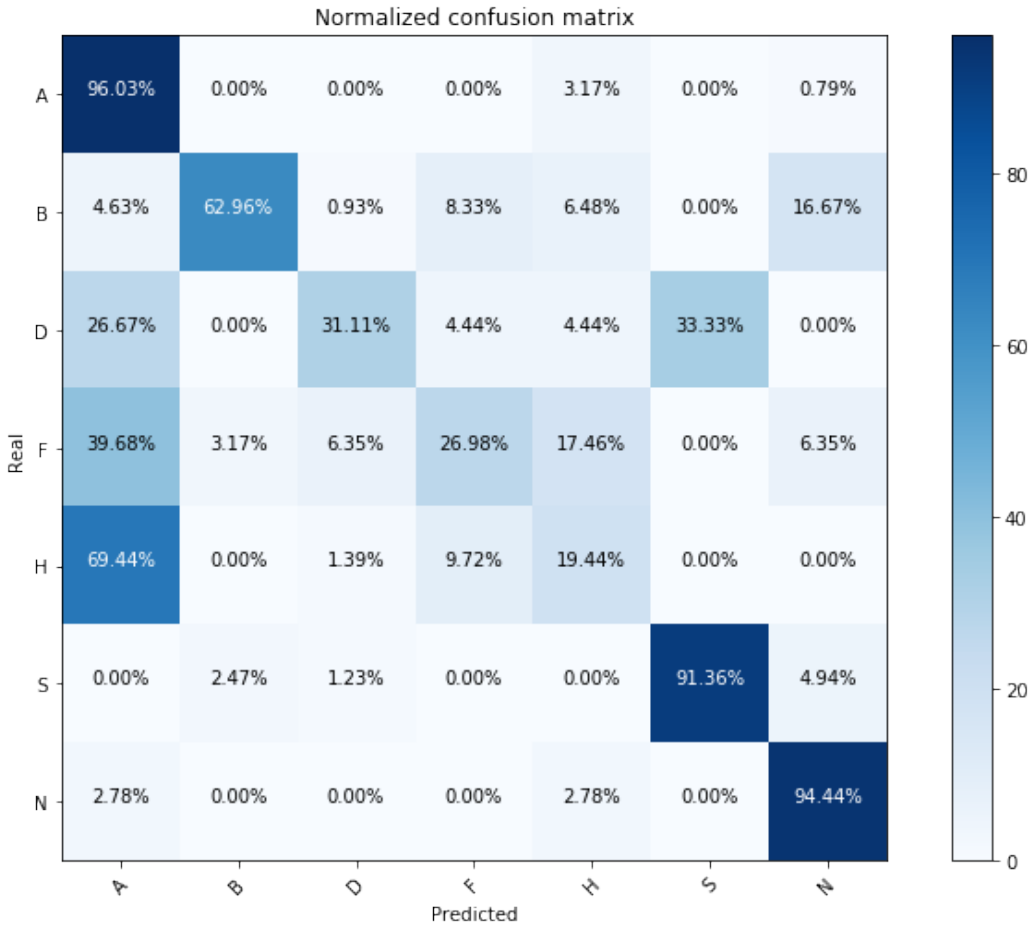
Figura 26 – Comparativo das Matrizes de Confusão SVM-excitação para validação e teste



(a) Validação (b) Teste

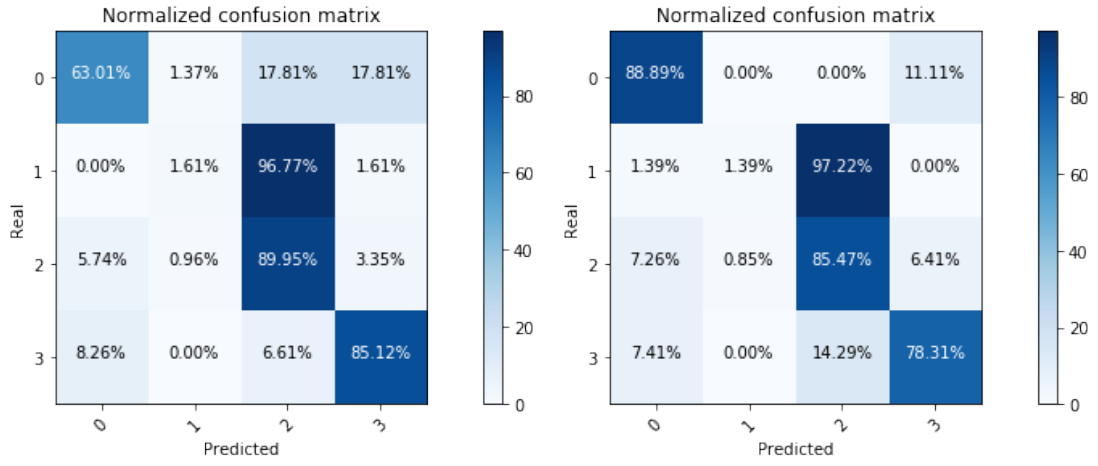
Fonte: Autor

Figura 27 – Comparativo das Matrizes de Confusão *Ensemble*-principais para validação e teste



Fonte: Autor

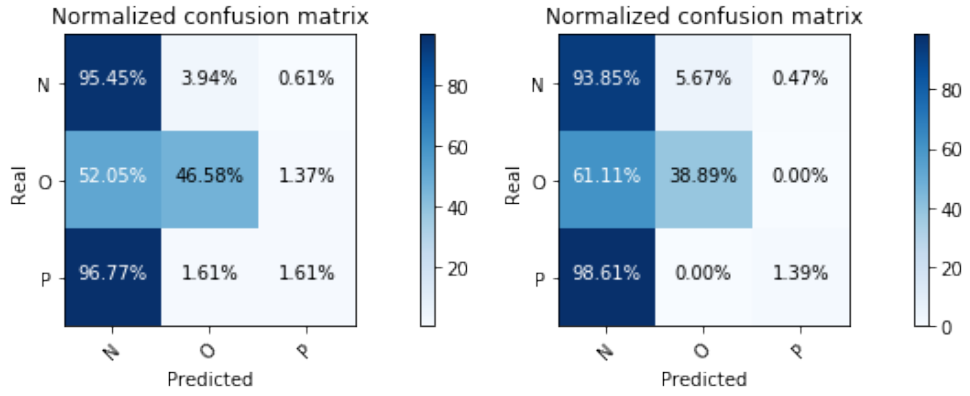
Figura 28 – Comparativo das Matrizes de Confusão SVM-quadrantes para validação e teste



(a) Validação (b) Teste

Fonte: Autor

Figura 29 – Comparativo das Matrizes de Confusão SVM-valência para validação e teste



(a) label 2 (b) label 2

Fonte: Autor

4.3 Resultados sobre as Árvores Hierárquicas

Para construção das árvores de decisão hierárquica foi utilizado os modelos de classificadores construídos sobre os dados de treino. A sequência que os classificadores foram montados sobre a árvore é baseada nos resultados F1-ponderado de validação obtidos por estes modelos, conforme metodologia explicada na Seção 3.3. Os dados de testes são aplicados sobre as árvores construídas para obter um novo conjunto de classificação de sentimentos de rótulos principais.

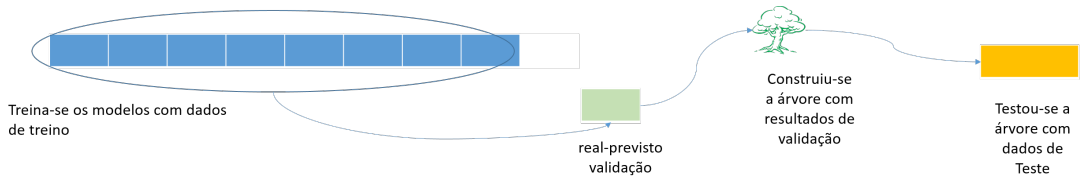
Duas abordagens foram utilizadas para a construção de árvores. A primeira foi construir uma árvore de decisão hierárquica para cada *fold*, baseada no desempenho de validação de seus modelos. A segunda abordagem unificou todas as duplas real-previsto dos resultados de validação dos classificadores de todos os *folds*, semelhante ao método descrito para a construção das Matrizes de Confusão apresentadas (ver Seção 4.1), e então construiu-se a árvore de decisão baseada no desempenho médio de validação desses resultados. Por motivos de simplificação, denomina-se cada árvore construída pela primeira abordagem como Árvores de Validação e a árvore construída sob a segunda abordagem como Árvore Geral. Os dados de teste foram aplicados sobre todas as Árvores de Validação e também sobre a Árvore Geral.

A Árvore Geral foi construída com base no desempenho médio de validação entre os *folds*. Entende-se que essa árvore é uma sequência de classificadores SVM, CNN-1D, CNN-2D e *Ensemble* que classificam, ao final, os sentimentos em rótulos principais. Todavia, cada um dos *folds* apresenta estruturas de classificadores iguais, mas com desempenhos distintos devido ao treinamento diferente em cada *fold*. Com isso, diz-se que a estrutura da Árvore Geral é a mesma entre todos os *folds*, e só se altera os pesos de treinamento

entre os *folds*, enquanto que a sequência de classificadores que aparecem nas Árvores de Validação varia entre os *folds*. Sendo assim, nove Árvores de Validação com sequências distintas de classificadores para cada *fold* foram construídas e nove Árvores Gerais foram construídas, com a mesma sequência de classificadores, sendo distintos entre si os modelos treinados em cada *fold*.

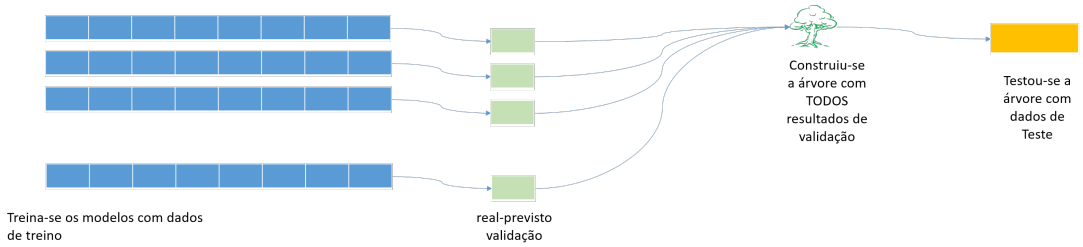
As Figuras 30 e 31 ilustram o esquema citado nos parágrafos acima. O objetivo da árvore é encontrar a sequência de modelos que ao serem combinados aumente a assertividade individual. Com isso, a Figura 30 mostra que uma árvore é construída para cada conjunto de validação de cada *fold*. A Figura 31 ilustra que todo os conjuntos de resultados de validação de todos os *folds* são usados para construir uma única árvore. Destaca-se o seguinte: mesmo que árvore tenha sido construída com base na combinação dos resultados de diferentes modelos de diferentes *folds*, para validar o desempenho da árvore geral construi-se oito árvores de sequência idênticas, porém utilizando modelos de cada *fold*. Não houve a mistura de modelos de diferentes *folds* justamente para evitar problemas de *overfitting*. Destaca-se mais uma vez que o conjunto de dados de teste não foi utilizado em nenhum momento para treino ou para validação dos modelos.

Figura 30 – Estrutura de classificadores da Árvore Validação



Fonte: Autor

Figura 31 – Estrutura de classificadores da Árvore Geral

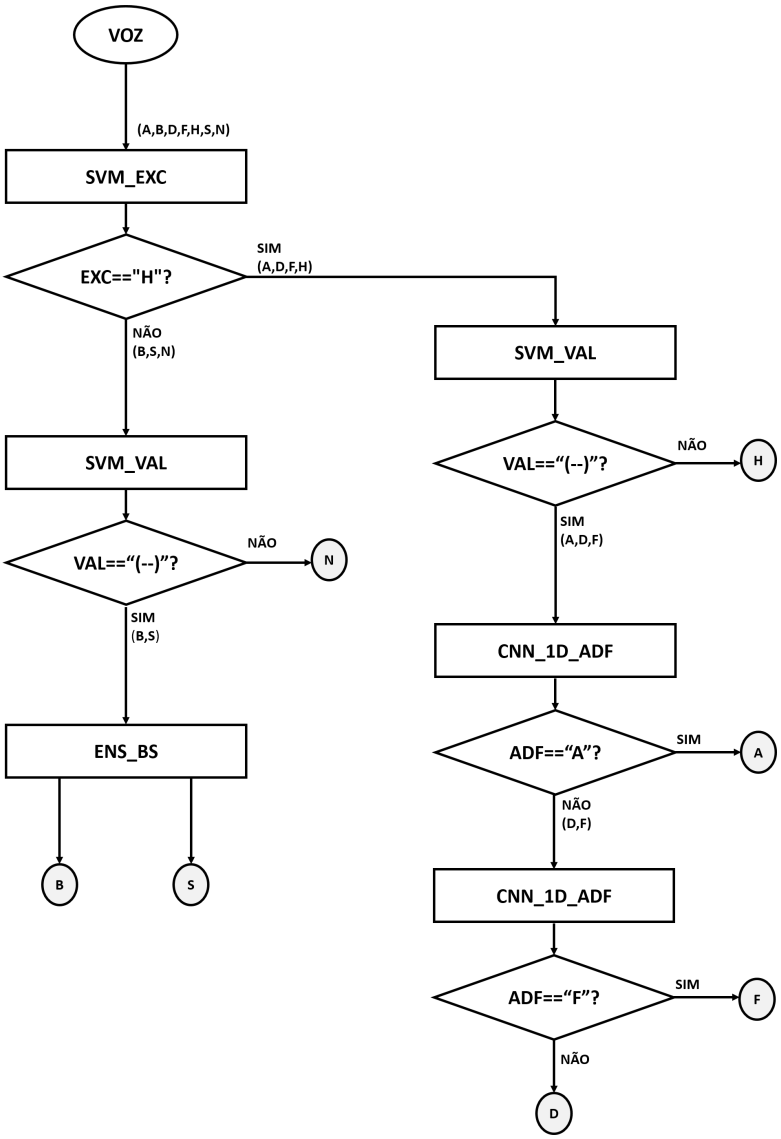


Fonte: Autor

A estrutura de classificadores escolhidas pela Árvore Geral está apresentada na Figura 32. As Árvores de Validação não são apresentadas por motivos de simplicidade, tendo seus resultados médios apresentados nas tabelas que seguem.

As Tabelas 21 e 22 apresentam a média dos resultados de classificação das Árvores de Validação aplicados sob dados de validação e dados de testes, respectivamente. Verifica-se na validação que o valor médio F1 obtido para todas as classes das árvores é superior

Figura 32 – Estrutura de classificadores da Árvore Geral



Fonte: Autor

a qualquer valor entre os classificadores. O desempenho F1 ponderado das classes foi de 0.74, sendo o *Ensemble* o de segundo maior desempenho médio com valor de 0.64. Contudo, o desempenho de teste sobre cada uma das árvores não é satisfatório, e conforme indica a tabela, a média do desempenho ponderado da árvore é de 0.58, sendo inferior ao desempenho do classificador SVM e *Ensemble* com 0.59 e 0.60, respectivamente. Apenas as classes D e F na árvore possuíram desempenho semelhante ao *Ensemble*.

Destes resultados, entende-se que construir árvores específicas para cada *fold* melhora significantemente o resultado da validação, mas piora sobre os resultados de teste, indicando assim que a ordem com que os modelos de classificação aparecem em cada árvore causou *overfitting* em dados de validação e reflete no baixo desempenho sob o conjunto de teste.

Tabela 21 – Árvore de Validação - Comparativo do desempenho F1 dos classificadores com dados de validação

		Classificadores - Média (std)				
Tipo	Classes.	CNN_1D	CNN_2D	SVM	ENS	ARV.
Princip.	A	0.78 (0.08)	0.75 (0.18)	0.78 (0.08)	0.79 (0.11)	**0.85 (0.08)**
	B	0.41 (0.16)	0.57 (0.22)	0.61 (0.10)	0.65 (0.11)	**0.79 (0.10)**
	D	0.33 (0.18)	0.60 (0.29)	0.45 (0.23)	0.56 (0.28)	**0.62 (0.19)**
	F	0.48 (0.15)	0.40 (0.28)	0.40 (0.21)	0.47 (0.22)	**0.67 (0.26)**
	H	0.35 (0.10)	0.42 (0.24)	0.45 (0.14)	0.42 (0.19)	**0.47 (0.26)**
	S	0.69 (0.20)	0.74 (0.16)	0.81 (0.12)	0.78 (0.12)	**0.92 (0.11)**
	N	0.62 (0.09)	0.57 (0.26)	0.65 (0.12)	0.68 (0.14)	**0.75 (0.14)**
	acurácia	0.58 (0.05)	0.63 (0.16)	0.63 (0.06)	0.67 (0.08)	**0.76 (0.09)**
	macro	0.52 (0.07)	0.58 (0.18)	0.59 (0.07)	0.62 (0.11)	**0.72 (0.12)**
	ponderado	0.56 (0.07)	0.6 (0.18)	0.62 (0.06)	0.64 (0.1)	**0.74 (0.11)**

Tabela 22 – Árvore de Validação - Comparativo do desempenho F1 dos classificadores com dados de teste

		Classificadores - Média (std)				
Tipo	Classes.	CNN_1D	CNN_2D	SVM	ENS	ARV.
Princip.	A	0.76 (0.02)	0.73 (0.11)	0.68 (0.02)	**0.72 (0.08)**	0.68 (0.03)
	B	0.53 (0.09)	0.62 (0.24)	0.65 (0.02)	**0.75 (0.14)**	0.69 (0.10)
	D	0.07 (0.15)	0.36 (0.18)	0.70 (0.06)	0.42 (0.11)	**0.48 (0.22)**
	F	0.44 (0.09)	0.42 (0.25)	0.23 (0.11)	0.30 (0.20)	**0.36 (0.16)**
	H	0.37 (0.09)	0.19 (0.13)	0.42 (0.06)	**0.25 (0.08)**	0.19 (0.12)
	S	0.87 (0.04)	0.81 (0.06)	0.79 (0.04)	**0.87 (0.06)**	0.85 (0.12)
	N	0.64 (0.06)	0.60 (0.28)	0.49 (0.09)	**0.71 (0.12)**	0.62 (0.13)
	acurácia	0.62 (0.02)	0.61 (0.12)	0.60 (0.02)	**0.64 (0.07)**	0.62 (0.04)
	macro	0.52 (0.03)	0.53 (0.15)	0.57 (0.03)	**0.57 (0.08)**	0.55 (0.06)
	ponderado	0.57 (0.03)	0.57 (0.14)	0.59 (0.03)	**0.61 (0.08)**	0.58 (0.05)

As Tabelas 23 e 24 apresentam o desempenho da classificação Árvore Geral sob dados de validação e também para os dados de teste, respectivamente, em cada *fold*. Analisando os resultados da Tabela 23, observa-se que a árvore possui um desempenho de validação F1-ponderado inferior se comparado a média dos desempenhos das árvores construídas especificamente para cada *fold*, na Tabela 21. Contudo, o desempenho dos dados de teste sobre a Árvore Geral se mostram mais satisfatórios que os dados obtidos em árvores específicas para cada *fold*.

Nos resultados de teste obtidos com a Árvore Geral, apresentados na Tabela 24, o desempenho global de acurácia se manteve o mesmo, e o desempenho F1 macro e ponderado aumentaram de 0.57 para 0.59 e de 0.60 para 0.62 do classificador de maior desempenho para o classificador Árvore. Os sentimentos B, S e N dos dados de teste sob a Árvore Geral foram classificados com desempenho superior a qualquer outro classificador. Classes de excitação alta tiveram desempenho ruim de classificação, sendo isso justificado talvez por uma má escolha das características dos sentimentos, ou mesmo o desequilíbrio entre as amostras de classes. As Figuras 33 e 34 apresentam Matrizes de Confusão para os dados de validação e testes aplicados na árvore.

Tabela 23 – Árvore Geral - Comparativo do desempenho F1 dos classificadores com dados de validação

		Classificadores - Média (std)				
Tipo	Classes.	CNN_1D	CNN_2D	SVM	ENS	ARV.
Princip.	A	0.78 (0.08)	0.75 (0.18)	0.78 (0.08)	**0.79 (0.11)**	**0.79 (0.12)**
	B	0.41 (0.16)	0.57 (0.22)	0.61 (0.1)	0.65 (0.11)	**0.76 (0.12)**
	D	0.33 (0.18)	**0.60 (0.29)**	0.45 (0.23)	0.56 (0.28)	0.46 (0.12)
	F	0.48 (0.15)	0.40 (0.28)	0.40 (0.21)	0.47 (0.22)	**0.54 (0.12)**
	H	0.35 (0.10)	0.42 (0.24)	0.45 (0.14)	0.42 (0.19)	**0.51 (0.12)**
	S	0.69 (0.20)	0.74 (0.16)	0.81 (0.12)	0.78 (0.12)	**0.82 (0.12)**
	N	0.62 (0.09)	0.57 (0.26)	0.65 (0.12)	0.68 (0.14)	**0.75 (0.12)**
	acurácia	0.58 (0.05)	0.63 (0.16)	0.63 (0.06)	0.67 (0.08)	**0.70 (0.12)**
	macro	0.52 (0.07)	0.58 (0.18)	0.59 (0.07)	0.62 (0.11)	**0.66 (0.12)**
	ponderado	0.56 (0.07)	0.6 (0.18)	0.62 (0.06)	0.64 (0.1)	**0.69 (0.12)**

Tabela 24 – Árvore Geral - Comparativo do desempenho F1 dos classificadores com dados de teste

		Classificadores - Média (std)				
Tipo	Classes.	CNN_1D	CNN_2D	SVM	ENS	ARV.
Princip.	A	0.76 (0.02)	0.73 (0.11)	0.68 (0.02)	**0.72 (0.08)**	0.63 (0.05)
	B	0.53 (0.09)	0.62 (0.24)	0.65 (0.02)	0.75 (0.14)	**0.85 (0.05)**
	D	0.07 (0.15)	0.36 (0.18)	**0.70 (0.06)**	0.42 (0.11)	0.28 (0.24)
	F	**0.44 (0.09)**	0.42 (0.25)	0.23 (0.11)	0.30 (0.20)	**0.44 (0.14)**
	H	0.37 (0.09)	0.19 (0.13)	**0.42 (0.06)**	0.25 (0.08)	0.24 (0.06)
	S	0.87 (0.04)	0.81 (0.06)	0.79 (0.04)	0.87 (0.06)	**0.92 (0.09)**
	N	0.64 (0.06)	0.6 (0.28)	0.49 (0.09)	0.71 (0.12)	**0.78 (0.07)**
	acurácia	0.62 (0.02)	0.61 (0.12)	0.60 (0.02)	**0.64 (0.07)**	**0.64 (0.02)**
	macro	0.52 (0.03)	0.53 (0.15)	0.57 (0.03)	0.57 (0.08)	**0.59 (0.04)**
	ponderado	0.57 (0.03)	0.57 (0.14)	0.59 (0.03)	0.61 (0.08)	**0.63 (0.03)**

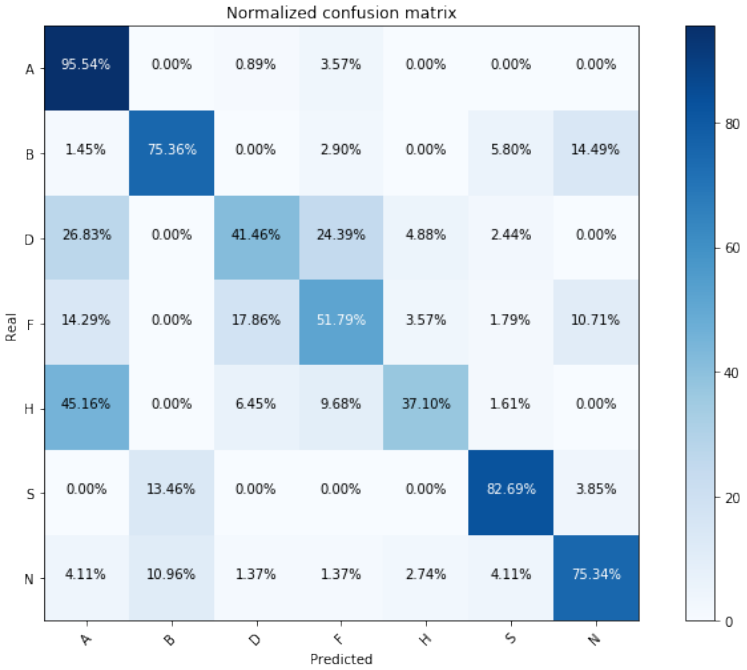
4.4 Comentários Gerais

Conforme Tabela 23, verifica-se que na classificação em rótulos principais, o *Ensemble* possui resultado médio superior para os demais classificadores, concordando assim com a hipótese de que as saídas dos classificadores SVM, CNN-1D e CNN-2D aumentam a assertividade de predição quando combinadas. Contudo, na Tabela 14 verifica-se que o classificador SVM possui melhor desempenho na classificação de outros rótulos além do tipo principais. Em muitos momentos os classificadores convolucionais apresentaram resultados não satisfatórios, o que pode também ter impactado no desempenho do *Ensemble*.

As arquiteturas de redes convolucionais CNN-1D e CNN-2D construídas foram as mesmas utilizadas para todos os tipos de rótulos. Isso pode ter influência nos resultados não satisfatórios, uma vez que talvez seria mais apropriado adaptar uma estrutura de rede para cada caso específico de tipo de rótulo. Importante mencionar também que a quantidade reduzida de dados de treinamento pode ter afetado no desempenho destas redes.

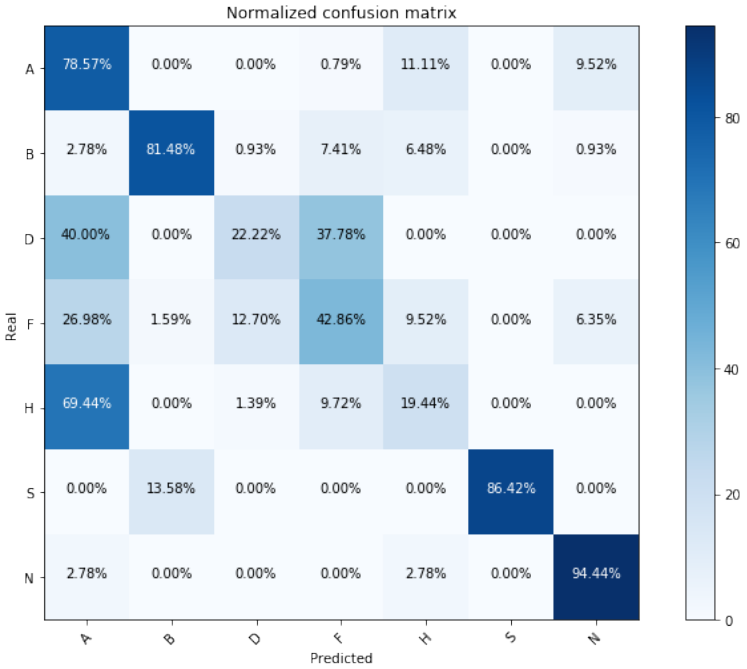
O *Ensemble* dos resultados foi calculado pela moda dos resultados dos classificadores SVM, CNN-1D e CNN-2D. O peso do resultado atribuído para todos os classificadores foi

Figura 33 – Matriz de Confusão para desempenho médio do classificador Árvore Geral - dados de validação



Fonte: Autor

Figura 34 – Matriz de Confusão para desempenho médio do classificador Árvore Geral - dados de teste



Fonte: Autor

igual. Na métrica F1-ponderada para dados de validação, conforme Tabela 23, o desempenho

do classificador *Ensemble* foi de 0.67 contra 0.63 do segundo melhor desempenho, enquanto que em dados de teste, conforme Tabela 24, o desempenho do *Ensemble* foi de 0.61 contra 0.59 do SVM, que obteve segundo melhor resultado. Uma proposta que pondera os pesos de cada classificador baseado em seus desempenhos durante o treino pode ser interessante, pois essa prática talvez garanta que o resultado final do *Ensemble* consiga ao mínimo repetir o melhor resultado entre os três classificadores.

O classificador Árvore Geral encontrou resultados médios superiores ao *Ensemble* sobre dados de validação e teste. Conforme apontam as Tabelas 23 e 24, para a métrica F1-ponderada, em dados de validação, o desempenho da Árvore foi de 0.69 contra 0.64 do *Ensemble*, enquanto que em dados de teste o desempenho da Árvore foi de 0.63 contra 0.61 do *Ensemble*. A melhoria, ainda que discreta, é promissora em indicar que os classificadores de sentimentos em rótulos intermediários foram capazes de aumentar o desempenho utilizando-se da combinação de diferentes informações a priori. Importante também citar desvantagens do uso do algoritmo, como o aumento do custo computacional e uma maior complexidade na construção da arquitetura da árvore.

A melhoria de desempenhos dos classificadores intermediários em excitação, valência e quadrante pode contribuir ainda para o melhor desempenho de classificação da árvore. Alguns classificadores intermediários, como os de valência e de quadrante, ainda apresentam problemas de *overfitting* e desequilíbrio de base. Se corrigidos, é possível que a construção da árvore aconteça por modelos mais fidedignos e, assim, alcançando melhores resultados.

5 Conclusões e Trabalhos Futuros

A proposta inicial deste trabalho foi identificar sentimentos em voz por meio de rótulos discretos utilizando, todavia, também as suas informações em excitação, valência e quadrantes presentes no mapa de afeto de Russel. Foi proposta a utilização de classificações intermediárias para auxiliar na classificação principal, e para tal abordagem foi utilizada a metodologia Multi-Tarefa. O intuito deste método foi fazer com que os classificadores aprendessem mais informações a respeito do sinal de voz que estão contidas também nos seus rótulos de sentimentos em excitação, valência e quadrantes. Para unificar os resultados de classificadores em diferentes rótulos, utilizou-se um algoritmo de Árvore de Decisão Hierárquica.

Além disso, outro objetivo foi extrair da voz uma gama maior de características do que as adotadas pelos classificadores atuais encontrados na literatura. Para isso, decidiu-se por utilizar os classificadores SVM, CNN-LSTM-1D, CNN-LSTM-2D, que extraem e classificam diferentes descritores a cerca do sinal de voz, fazendo com que a classificação final acontecesse através de várias informações extraídas do sinal. O método encontrado para unificar os resultados destes classificadores foi o *Ensemble*.

Conforme os resultados apresentados nas Tabelas 24 e 24, verifica-se que o *Ensemble* dos classificadores conseguiu desempenho geral igual, ou superior, aos resultados individuais de cada um destes, em todas as três métricas analisadas. Na métrica F1-ponderada, em dados de validação, o desempenho do classificador *Ensemble* foi de 0.64 contra 0.62 do segundo melhor desempenho, enquanto que em dados de teste, o desempenho do *Ensemble* foi de 0.63 contra 0.61 do segundo melhor. Estes valores foram obtidos na validação do modelo por *k-fold*. Isso mostra que a abordagem Multi-Características do método foi capaz de aprimorar o desempenho.

O algoritmo da Árvore de Decisão Hierárquica, responsável por unificar os resultados dos classificadores intermediários em excitação, valência e quadrantes, aumentou o desempenho geral de todos os classificadores individuais, inclusive o do próprio *Ensemble*. Conforme apontam as Tabelas 23 e 24, para a métrica F1-ponderada, em dados de validação, o desempenho da Árvore foi de 0.69 contra 0.64 do *Ensemble*, enquanto que em dados de teste o desempenho da Árvore foi de 0.63 contra 0.61 do *Ensemble*. Estes valores foram obtidos na validação do modelo por *k-fold*.

Como análise final dos resultados, pode-se observar também a Figura 32 que apresenta a estrutura de classificadores escolhidos para a Árvore de Decisão Hierárquica. Nela se observa que diferentes tipos de classificadores, cada um extraindo e classificando características distintas do sinal de voz, foram utilizados para incrementar o resultado

final da classificação. De modo geral, o fato desses classificadores utilizarem múltiplas características da voz para uma classificação final, demonstra que esse algoritmo também funciona em uma abordagem Multi-Características. Ou seja, a abordagem Multi-Tarefa apresentada pela árvore e pelos classificadores de outros tipos de rótulos está muito atrelada também com a abordagem Multi-Características exposta. Visto isso, o autor entende que Multi-Características auxiliou a encontrar as melhores características não somente do sinal de voz em si, mas também do sinal de voz atrelado ao tipo de rótulo que estava sendo analisado.

Considerando que a base de dados Berlin é desbalanceada, em trabalhos futuros pretende-se aplicar algum método para amenizar esse problema. Pretende-se ainda também aplicar a metodologia utilizada para outras bases de dados de tamanhos maiores, já que a base Berlin possui poucas amostras. Além disso, a estrutura dos classificadores foi a mesma independente do tipo de rótulo.

Quanto a estrutura dos modelos de redes convolucionais, sugere-se a aplicação de técnicas de *data augmentation* nos dados de voz, gerando novas amostras por meio do aumento da taxa de reprodução, variação aleatória do volume, entre outras, dando assim uma quantidade maior de dados no treino das redes. Além disso, o uso da técnica de *transfer learning* pode ser utilizado para definir os pesos iniciais da arquitetura neural e conseguir melhores resultados de classificação nesta pesquisa, por meio de *fine-tuning*. Uma pesquisa mais aprofundada para encontrar a melhor arquitetura, específica para cada tipo de rótulo, também é interessante. A técnica de *Ensemble* de resultados também pode ser aprimorada, por exemplo, utilizando ao invés da moda do resultados, algum método que seja capaz de dar um peso a cada voto, baseado no desempenho individual de cada classificador.

Quanto a separação de dados, é encontrado na literatura algumas pesquisas que fazem distinção de indivíduos e de sexo na hora de treino e teste. Essa abordagem não foi aplicada neste trabalho, podendo ser analisada em trabalhos futuros.

Referências

ALBADAWY, E. A.; KIM, Y. Joint Discrete and Continuous Emotion Prediction Using Ensemble and End-to-End Approaches. v. 18, p. 366–375, 2018. Disponível em: <https://doi.org/10.1145/3242969.3242972>. Citado na página 29.

AYADI, M. E.; KAMEL, M. S.; KARRAY, F. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, Elsevier, v. 44, n. 3, p. 572–587, 2011. ISSN 00313203. Disponível em: <http://dx.doi.org/10.1016/j.patcog.2010.09.020>. Citado 2 vezes nas páginas 23 e 25.

BADSHAH, A. M.; AHMAD, J.; RAHIM, N.; BAIK, S. W. Speech Emotion Recognition from Spectrograms with Deep Convolutional Neural Network. *2017 International Conference on Platform Technology and Service (PlatCon)*, p. 1–5, 2017. Disponível em: <http://ieeexplore.ieee.org/document/7883728/>. Citado 6 vezes nas páginas 26, 49, 51, 52, 53 e 70.

BESTELMEYER, P. E.; KOTZ, S. A.; BELIN, P. Effects of emotional valence and arousal on the voice perception network. *Social Cognitive and Affective Neuroscience*, v. 12, n. 8, p. 1351–1358, 2017. ISSN 17495024. Citado 2 vezes nas páginas 23 e 26.

BISHOP, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387310738. Citado 5 vezes nas páginas 37, 38, 39, 40 e 42.

BURKHARDT, F.; PAESCHKE, A.; ROLFES, M.; SENDLMEIER, W. F.; WEISS, B. A Database of German Emotional Speech. *Interspeech*, n. January, p. 1517–1520, 2005. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.130.8506&rep=rep1&type=pdf>. Citado 3 vezes nas páginas 51, 59 e 60.

CHANG, C.-c.; LIN, C.-j. LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, v. 2, p. 1–39, 2013. ISSN 21576904. Citado 2 vezes nas páginas 42 e 68.

DASGUPTA, P. B. Detection and analysis of human emotions through voice and speech pattern processing. *International Journal of Computer Trends and Technology*, Seventh Sense Research Group Journals, v. 52, n. 1, p. 1–3, Oct 2017. ISSN 2231-2803. Disponível em: <http://dx.doi.org/10.14445/22312803/IJCTT-V52P101>. Citado 2 vezes nas páginas 24 e 25.

FAYEK, H. M.; LECH, M.; CAVEDON, L. Evaluating deep learning architectures for Speech Emotion Recognition. *Neural Networks*, Elsevier Ltd, v. 92, p. 60–68, 2017. ISSN 18792782. Disponível em: <http://dx.doi.org/10.1016/j.neunet.2017.02.013>. Citado 2 vezes nas páginas 27 e 47.

FLACH, P. A.; KULL, M. Precision-recall-gain curves: Pr analysis done right. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. Cambridge, MA, USA: MIT Press, 2015. (NIPS'15), p. 838–846. Disponível em: <http://dl.acm.org/citation.cfm?id=2969239.2969333>. Citado 2 vezes nas páginas 56 e 57.

GADHE, R. P.; NILOFER, S.; WAGHMARE, V. B.; SHRISHRIMAL, P. P.; DESHMUKH, R. R. Emotion Recognition from Speech: A Survey. *International Journal of Scientific & Engineering Research*, v. 6, n. 4, p. 632–635, 2015. ISSN 2229-5518. Disponível em: <http://www.ijser.org>. Citado na página 23.

GHARAVIAN, D.; SHEIKHAN, M.; NAZERIEH, A.; GAROUCY, S. Speech emotion recognition using fcbf feature selection method and ga-optimized fuzzy artmap neural network. *Neural Computing and Applications - NCA*, v. 21, p. 1–12, 11 2011. Citado na página 24.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado 5 vezes nas páginas 43, 44, 45, 46 e 49.

GRAVES, A. *Supervised Sequence Labelling with Recurrent Neural Networks*. [S.l.: s.n.], 2011. Citado 6 vezes nas páginas 43, 44, 45, 46, 47 e 49.

HUANG, Z.; DONG, M.; MAO, Q.; ZHAN, Y. Speech Emotion Recognition Using CNN. *Proceedings of the ACM International Conference on Multimedia - MM '14*, p. 801–804, 2014. Disponível em: <http://dl.acm.org/citation.cfm?doid=2647868.2654984>. Citado na página 52.

LEE, C. C.; MOWER, E.; BUSSO, C.; LEE, S.; NARAYANAN, S. Emotion recognition using a hierarchical binary decision tree approach. *Speech Communication*, v. 53, n. 9-10, p. 1162–1171, 2011. ISSN 01676393. Citado 2 vezes nas páginas 31 e 54.

LIU, W.; WANG, Z.; LIU, X.; ZENG, N.; LIU, Y.; ALSAADI, F. E. A survey of deep neural network architectures and their applications. *Neurocomputing*, Elsevier B.V., v. 234, n. December 2016, p. 11–26, 2017. ISSN 18728286. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2016.12.038>. Citado 3 vezes nas páginas 31, 54 e 55.

LIVINGSTONE, S. R.; RUSSO, F. A. *The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north American english*. [S.l.: s.n.], 2018. v. 13. ISSN 19326203. ISBN 1111111111. Citado 2 vezes nas páginas 23 e 67.

LORENA, A.; COGNIçãO, C. e; CARVALHO, A.; COMPUTAçãO, D. de Ciências de; COMPUTAçãO, I. de Ciências Matemáticas e de; , U. . Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada; Vol. 14, No 2 (2007); 43-67*, v. 14, 01 2007. Citado 6 vezes nas páginas 36, 37, 39, 40, 41 e 42.

LYON, R.; SHAMMA, S. Auditory representations of timbre and pitch. In: _____. *Auditory Computation*. New York, NY: Springer New York, 1996. p. 221–270. ISBN 978-1-4612-4070-9. Disponível em: <https://doi.org/10.1007/978-1-4612-4070-9_6>. Citado na página 25.

MAO, Q.; WANG, X.; ZHAN, Y. Speech Emotion Recognition Method Based on Improved Decision Tree and Layered Feature Selection. *International Journal of Humanoid Robotics*, v. 07, n. 02, p. 245–261, 2010. ISSN 0219-8436. Disponível em: <http://www.worldscientific.com/doi/abs/10.1142/S0219843610002088>. Citado na página 54.

MAO, S. Automatic Speech Emotion Recognition Using Deep Learning. v. 118, n. 20, p. 1–134, 2018. ISSN 1607-551X. Citado 3 vezes nas páginas 31, 54 e 55.

PARTHASARATHY, S.; BUSSO, C. Jointly predicting arousal, valence and dominance with multi-Task learning. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, v. 2017-Augus, p. 1103–1107, 2017. ISSN 19909772. Citado na página 23.

PASCANU, R.; MIKOLOV, T.; BENGIO, Y. On the difficulty of training recurrent neural networks. In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. JMLR.org, 2013. (ICML'13), p. III–1310–III–1318. Disponível em: <http://dl.acm.org/citation.cfm?id=3042817.3043083>. Citado na página 47.

PATHAK, S.; KOLHE, V. A Survey on Emotion Recognition from Speech Signal. *International Journal of Advanced Research in Computer and Communication Engineering*, v. 5, n. 7, p. 447–450, 2016. ISSN 2278-1021. Citado 2 vezes nas páginas 23 e 26.

QUIROS-RAMIREZ, M. A.; POLIKOVSKY, S.; KAMEDA, Y.; ONISAWA, T. INTERNATIONAL CONFERENCE ON KANSEI ENGINEERING AND EMOTION RESEARCH A Spontaneous Cross-Cultural Emotion Database :. p. 1127–1134, 2014. Citado na página 68.

RABINER, L. R.; SCHAFER, R. W. *Theory and Applications of Digital Speech Processing*. [s.n.], 2011. 1056 p. ISBN 9780136034285. Disponível em: <https://books.google.com.au/books?id=W9leOgAACAAJ>. Citado 3 vezes nas páginas 24, 25 e 26.

REDDY, A. P.; VIJAYARAJAN, V. Extraction of Emotions from Speech - A Survey. *International Journal of Applied Engineering Research ISSN*, v. 12, n. 16, p. 973–4562, 2017. ISSN 09739769. Citado 4 vezes nas páginas 23, 25, 26 e 35.

RODELLAR-BIARGE, V.; PALACIOS-ALONSO, D.; NIETO-LLUIS, V.; GõMEZ-VILDA, P. Towards the search of detection in speech-relevant features for stress. *Expert Systems*, v. 32, n. 6, p. 710–718, 2015. ISSN 14680394. Citado na página 23.

RUSSELL, J. A. A circumplex model of affect. *Journal of Personality and Social Psychology*, v. 39, n. 6, p. 1161–1178, 1980. ISSN 00223514. Citado 2 vezes nas páginas 24 e 29.

SAILUNAZ, K.; DHALIWAL, M.; ROKNE, J.; ALHAJJ, R. Emotion detection from text and speech: a survey. *Social Network Analysis and Mining*, Springer Vienna, v. 8, n. 1, 2018. ISSN 18695469. Disponível em: <https://doi.org/10.1007/s13278-018-0505-2>. Citado na página 25.

Santiago-Omar Caballero-Morales. Recognition of Emotions in Mexican Spanish Speech : An Approach Based on Acoustic Modelling of Emotion-Specific Vowels. v. 2013, p. 13–16, 2013. Citado na página 68.

SATT, A.; ROZENBERG, S.; HOORY, R.; RESEARCH-HAIFA, I. B. M. Efficient Emotion Recognition from Speech Using Deep Learning on Spectrograms. *Interspeech2017*, p. 1089–1093, 2017. Disponível em: <https://pdfs.semanticscholar.org/de47/fc09bc8dcd032c8b3450a0b2a816c376e07e.pdf>. Citado 2 vezes nas páginas 49 e 53.

SHEN, P.; CHANGJUN, Z.; CHEN, X. Automatic Speech Emotion Recognition using Support Vector Machine. *Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference*, v. 2, p. 621–625, 2011. Citado 5 vezes nas páginas 25, 26, 35, 69 e 70.

SHIH, P.-Y.; CHEN, C.-P.; WU, C.-H. SPEECH EMOTION RECOGNITION WITH ENSEMBLE LEARNING METHODS Po-Yuan. p. 2756–2760, 2017. Citado 4 vezes nas páginas 27, 28, 55 e 56.

SOLEYMANI, M.; LICHTENAUER, J.; PUN, T.; PANTIC, M. A multimodal database for affect recognition and implicit tagging. *IEEE Transactions on Affective Computing*, Institute of Electrical and Electronics Engineers (IEEE), v. 3, n. 1, p. 42–55, jan 2012. Disponível em: <https://doi.org/10.1109/t-affc.2011.25>. Citado na página 28.

SWAIN, M.; ROUTRAY, A. Databases , features and classifiers for speech emotion recognition : a review. *International Journal of Speech Technology*, Springer US, v. 0, n. 0, p. 0, 2018. ISSN 1572-8110. Disponível em: <http://dx.doi.org/10.1007/s10772-018-9491-z>. Citado na página 68.

THRUN, S. Multitask Learning *. v. 75, p. 41–75, 1997. Citado na página 30.

TUAROB, S.; TUCKER, C. S.; SALATHE, M.; RAM, N. An ensemble heterogeneous classification methodology for discovering health-related knowledge in social media messages. *Journal of Biomedical Informatics*, Elsevier Inc., v. 49, p. 255–268, 2014. ISSN 15320464. Disponível em: <http://dx.doi.org/10.1016/j.jbi.2014.03.005>. Citado 2 vezes nas páginas 28 e 56.

VARSAMOPOULOS, S.; BERTELS, K. Designing neural network based decoders for surface codes. n. December, 2018. Citado 3 vezes nas páginas 47, 48 e 49.

VERVERIDIS, D.; KOTROPOULOS, C. Emotional speech recognition: Resources, features, and methods. *Speech Communication*, v. 48, n. 9, p. 1162–1181, 2006. ISSN 01676393. Citado 2 vezes nas páginas 25 e 70.

VRYZAS, N.; VRYSIS, L.; KOTSAKIS, R.; DIMOULAS, C. Speech emotion recognition adapted to multimodal semantic repositories. IEEE, sep 2018. Disponível em: <https://doi.org/10.1109/smap.2018.8501881>. Citado na página 28.

WANG, X. W.; NIE, D.; LU, B. L. Emotional state classification from EEG data using machine learning approach. *Neurocomputing*, Elsevier, v. 129, p. 94–106, 2014. ISSN 09252312. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2013.06.046>. Citado 3 vezes nas páginas 23, 24 e 29.

XIA, R.; LIU, Y. A Multi-Task Learning Framework for Emotion Recognition Using 2D Continuous Space. *IEEE Transactions on Affective Computing*, IEEE, v. 8, n. 1, p. 3–14, 2017. ISSN 19493045. Citado 4 vezes nas páginas 23, 30, 31 e 55.

ZHANG, S.; ZHANG, S.; HUANG, T.; GAO, W. Convolutional Neural Network and Discriminant. v. 20, n. 6, p. 1576–1590, 2018. Citado 5 vezes nas páginas 26, 27, 28, 29 e 55.

ZHANG, X. Ensemble System for Multimodal Emotion Recognition Challenge (MEC 2017). *2018 First Asian Conference on Affective Computing and Intelligent Interaction (ASCII Asia)*, n. Mec, p. 1–6, 2018. Citado na página 29.

ZHANG, Y.; LIU, Y.; WENINGER, F. MULTI-TASK DEEP NEURAL NETWORK WITH SHARED HIDDEN LAYERS : BREAKING DOWN THE WALL BETWEEN EMOTION REPRESENTATIONS Department of Computing , Imperial College London , London , United Kingdom Nuance Communications , Ulm , Germany. v. 645378, n. 645378, p. 4990–4994, 2017. Citado 2 vezes nas páginas 25 e 30.

ZHAO, J.; MAO, X.; CHEN, L. Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomedical Signal Processing and Control*, Elsevier Ltd, v. 47, p. 312–323, 2019. ISSN 17468094. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1746809418302337>. Citado 9 vezes nas páginas 26, 27, 46, 47, 49, 50, 51, 52 e 53.

ZHOU, Z.-H. *Ensemble Methods: Foundations and Algorithms*. 1st. ed. [S.l.]: Chapman & Hall/CRC, 2012. ISBN 1439830037, 9781439830031. Citado 2 vezes nas páginas 53 e 54.