UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

André Furlan

Monografia de estudos especiais

São José do Rio Preto

2019

André Furlan

Caracterização de *voice spoofing* para fins de verificação de locutores com base na transformada wavelet e na análise paraconsistente de características

Monografia apresentada para cumprimento da disciplina de estudos especiais do curso de Mestrado em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Campus de São José do Rio Preto.

Orientador: Prof. Dr. Rodrigo Capobianco Guido

São José do Rio Preto, SP

2019

André Furlan

Caracterização de *voice spoofing* para fins de verificação de locutores com base na transformada wavelet e na análise paraconsistente de características

Monografia apresentada para cumprimento da disciplina de estudos especiais do curso de Mestrado em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Campus de São José do Rio Preto.

Orientador: Prof. Dr. Rodrigo Capobianco Guido

Comissão Examinadora

Professor Dr. Rodrigo Capobianco Guido
UNESP - Campus de São José do Rio Preto
Co-Orientador

Professora Dra. Renata Spolon Lobato
UNESP - Campus de São José do Rio Preto

Professor Dr. Aleardo Manacero Júnior
UNESP - Campus de São José do Rio Preto

São José do Rio Preto, SP
2019

RESUMO

Este documento constitui a monografia produzida como resultado dos estudos especiais realizados pelo autor visando promover um levantamento bibliográfico inicial do tema de sua dissertação de mestrado. Foram inclusas as descrições essenciais de 15 trabalhos científicos na área de *voice spoofing detection*, acrescidos ainda, das direções que caracterizam o tema do trabalho em questão visando a monografia de qualificação.
No capítulo 1 é feita uma breve introdução. Iniciando as revisões de conceitos apresenta-se a engenharia paraconsistente de características, filtros digitais *wavelets*, amostragem, quantização e o formato do arquivo *wave* e caracterização dos processos de produção da voz humana no capítulo 2 seção 1, a seguir, na seção 2 deste mesmo capítulo apresenta-se a revisão bibliográfica inicial. Finalmente no capítulo 3 é mostrado um cronograma com os trabalhos já realizados e uma previsão da realização dos próximos.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

Os sistemas de autenticação de usuário por voz e o processo de falseamento usando *voice spoofing* é o tema de estudo que se pretende explorar. Um sistema de reconhecimento **idealmente** não deve se deixar enganar por, como exemplo, uma voz gravada, neste documento será apresentada uma revisão de conceitos que visa preparar as bases para a construção de métodos que reconheçam esse ataque.

No capítulo 2 se realizará uma revisão dos seguintes conceitos:

- Engenharia paraconsistente.
- Filtros digitais usando wavelets.
- Caracterização dos processos de produção da voz humana.
- Amostragem, quantização, entre outros.

Em seguida apresentar-se-á uma revisão bibliográfica finalizando com o cronograma previsto.

Capítulo 2

Revisão de Bibliográfica

2.1 Conceitos utilizados

2.1.1 Engenharia paraconsistente de características

Dentro do processo de classificação frequentemente surge a questão:
Os vetores de características criados proporcionam uma boa separação de classes?

O método de cálculo do plano paraconsistente é uma ferramenta que pode ser usada para responder essa questão.

O processo inicia-se após a aquisição dos vetores de características para cada classe (C_n) onde n é o índice de cada uma delas. Se o número de classes presentes for, por exemplo, quatro então estas poderão ser representadas por C_1, C_2, C_3, C_4.

Em seguida será necessário o cálculo de duas grandezas:

- A menor similaridade intraclasse (α).
- A razão de sobreposição interclasse (β)

α indica o quanto de similaridade os dados têm entre si dentro de uma mesma classe, β mostra a razão de sobreposição entre diferentes classes. Idealmente α deve ser maximizada e β minimizada para um desempenho ótimo dos classificadores.

Inicialmente é necessária a normalização dos vetores de características de forma que a soma de todos os seus valores seja um.

Em seguida a obtenção de α se dá selecionando-se os maiores e os menores valores de cada uma das posições de todos os vetores de características de cada classe gerando assim um vetor para os valores maiores e outro para os menores.

O **vetor de similaridade**(svC_n) é obtido fazendo-se a diferença item a item dos maiores em relação aos menores.

Finalmente e para cada classe é tirada a média dos valores de cada vetor de similaridade, α é o menor valor dentre essas médias.

A figura 2.1 ilustra este processo.

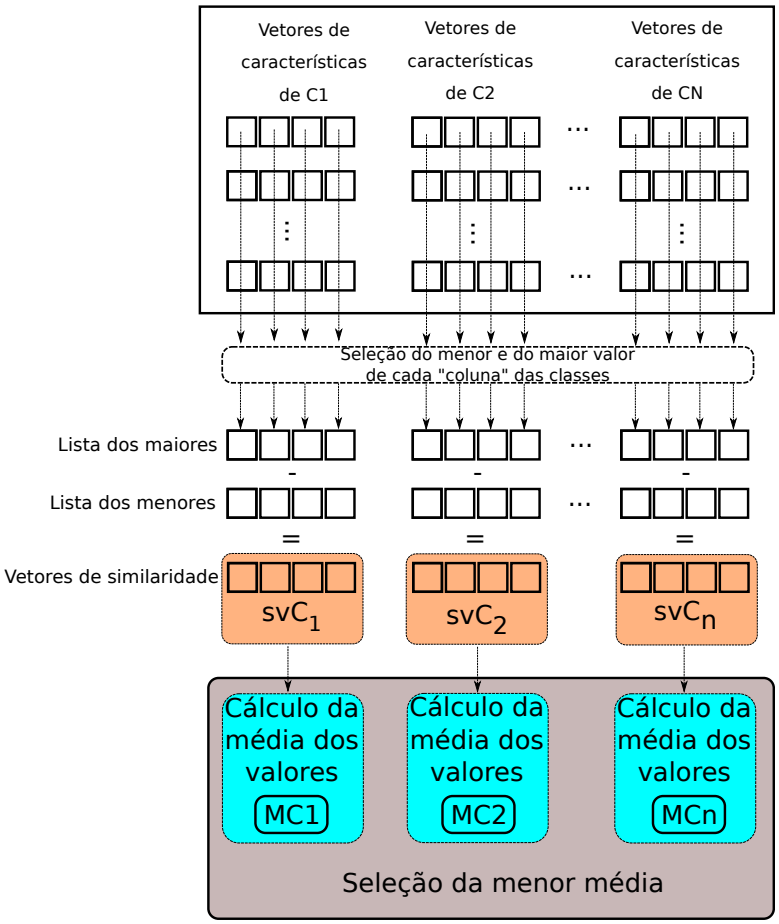

Figura 2.1: Cálculo de α

A obtenção de β, assim como ilustrado na figura 2.2, também se dá selecionando-se os maiores e os menores valores de cada uma das posições de todos os vetores de características de cada classe gerando assim um vetor para os valores maiores e outro

para os menores.

Na sequência se segue com o cálculo de R cujo valor é a quantidade de vezes que um valor do vetor de características de uma classe se encontra no intervalo de valores maiores e menores de outra classe.

É necessário o cálculo de F que é o número máximo de sobreposições possíveis entre classes e é dado por:

$$F = N.(N-1).X.T \tag{2.1}$$

onde:

- N é a quantidades de classes.
- X é quantidade de vetores de características por classe.
- T é o tamanho do vetor de características.

Finalmente, β é calculado:

$$\beta = \frac{R}{F} \tag{2.2}$$

Nesse ponto é importante notar que $\alpha = 1$ sugere fortemente que os vetores de características de cada classe são similares e representam suas respectivas classes precisamente. Complementarmente $\beta = 0$ sugere os vetores de características de classes diferentes não se sobrepõe [Gui19].

- Verdade → Fé total ($\alpha = 1$) e nenhum descrédito ($\beta = 0$)
- Ambiguidade → Fé total ($\alpha = 1$) e descrédito total ($\beta = 1$)
- Falsidade → Fé nula ($\alpha = 0$) e descrédito total ($\beta = 1$)
- Indefinição → Fé nula ($\alpha = 0$) e descrédito total ($\beta = 0$)

No entanto, raramente α e β terão tais valores, na maioria do tempo $0 \leqslant \alpha \leqslant 1$ e $0 \leqslant \beta \leqslant 1$, por isso, se torna necessário o cálculo do **grau de certeza**(G_1) e do **grau de contradição**(G_2).

$$G_1 = \alpha - \beta \tag{2.3}$$

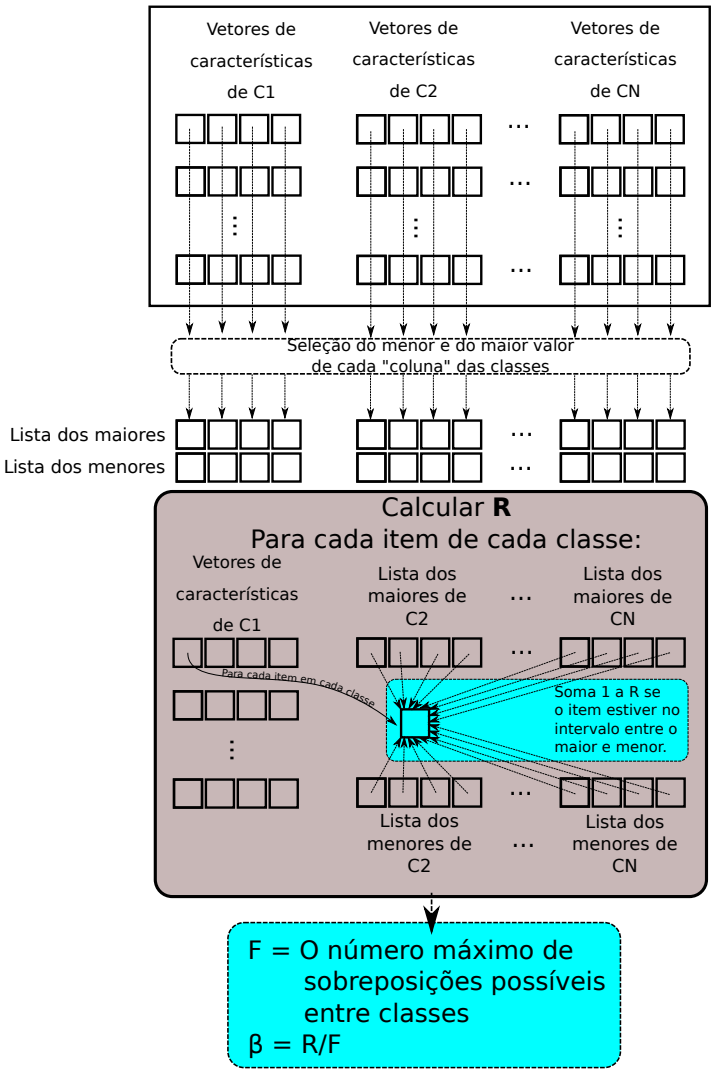

Figura 2.2: Cálculo de β

$$G_2 = \alpha + \beta - 1 \qquad , \tag{2.4}$$

onde: $-1 \leqslant G_1$ e $1 \geqslant G_2$.

Os valores de G_1 e G_2 em conjunto definem os graus entre verdade e falsidade, ou seja, $G_1 = -1$ e $G_1 = 1$ respectivamente e também os graus entre indefinição e ambiguidade, ou seja, $G_2 = -1$ e $G_2 = 1$ respectivamente.

O plano paraconsistente para fins de visualização e maior rapidez na avaliação dos resultados como ilustrado na figura 2.3 tem quatro cantos definidos:

- (-1,0) → Falsidade.

- (1,0) → Verdade.
- (0,-1) → Indefinição.
- (0,1) → Ambiguidade.

É importante perceber que na figura 2.3 existe um pequeno círculo, este indica onde se encontram as classes nos graus explicados da listagem anterior.

Para se ter ideia em que área exatamente se encontram as classes avaliadas se deve calcular as distâncias(D) do ponto $P = (G_1, G_2)$ dos limites supracitados. Tal cálculo pode ser feito da seguinte forma:

$$D_{-1,0} = \sqrt{(G_1 + 1)^2 + (G_2)^2} \quad , \tag{2.5}$$

$$D_{1,0} = \sqrt{(G_1 - 1)^2 + (G_2)^2} \quad , \tag{2.6}$$

$$D_{0,-1} = \sqrt{(G_1)^2 + (G_2 + 1)^2} \quad , \tag{2.7}$$

$$D_{0,1} = \sqrt{(G_1)^2 + (G_2 - 1)^2} \quad , \tag{2.8}$$

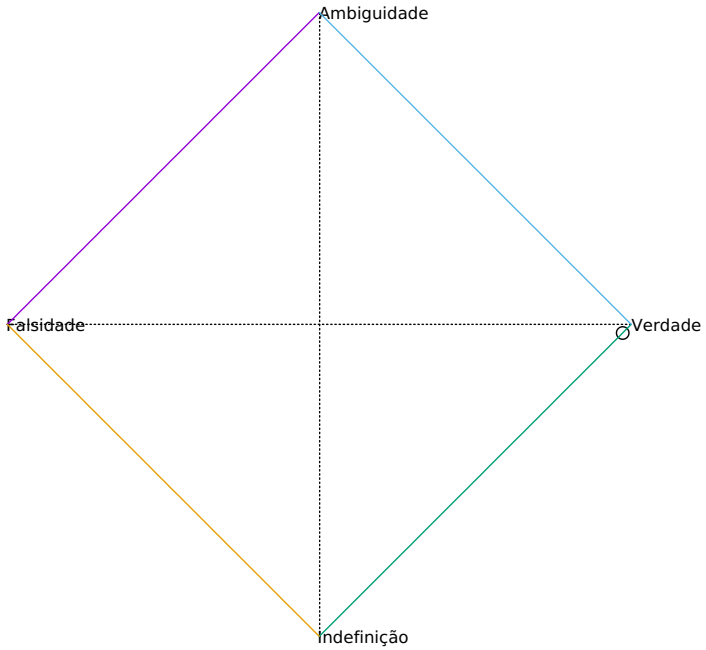

Figura 2.3: O plano paraconsistente.

2.1.2 Filtros digitais wavelets

Filtros digitais *wavelet* vem para suprir as deficiências de janelamento de sinal apresentadas pelas transformadas de Fourier e pelas transformadas curtas de Fourier. *Wavelets* contam com variadas funções filtro e tem tamanho de janela variável o que permite uma análise multirresolução [AWG09].

As *wavelets* proporcionam a análise do sinal de forma detalhada tanto no espectro de baixa frequência quanto no de alta frequência.

É importante observar que, quando se trata de transformada *wavelet* seis elementos estão presentes: dois filtros de análise, dois filtros de síntese e as funções ortogonais *scaling* e *wavelet*. No tocante a sua aplicação só a transformada direta, e não a inversa, será usada na construção dos vetores de características então os filtros de síntese, a função *scaling* e a função *wavelet* não serão elementos abordados aqui, pois, esses só interessariam caso houvesse a necessidade da transformada inversa. A abordagem usada será baseada nos filtros de análise digitais que proporcionarão a decomposição do sinal com o uso de filtros passa baixas e passa altas estritamente no domínio discreto.

No contexto dos filtros digitais baseados em *wavelets* o tamanho da janela recebe o nome de **suporte**. Janelas definem o tamanho do filtro que será aplicado ao sinal quando esse é pequeno se diz que a janela tem **um suporte compacto** [P+96].

Se diz que uma *wavelet* tem boa **resposta em frequência** quando, na aplicação da mesma na filtragem das frequências não são causadas muitas pertubações indesejadas ao sinal, as wavelets de Daubechies se destacam neste quesito por serem *maximamente planas* (Maximally-flat) nos platôs de resposta em frequência como indicado na figura 2.4.

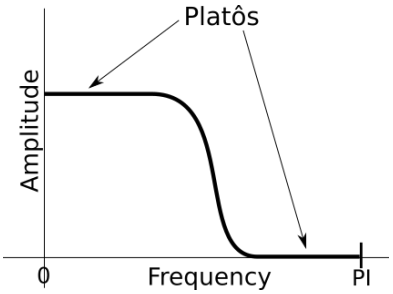

Figura 2.4: Platôs maximamente planos em um filtro digital

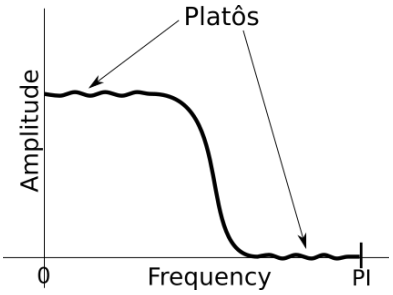

Figura 2.5: Platôs não maximamente planos de um filtro digital

Além da resposta em frequência a aplicação de um filtro digital *wavelets* também pode gerar o que se chama de **resposta em fase**, esse deslocamento pode ser **linear**, **quase linear** ou **não linear**.

- Na resposta em fase **linear** há o mesmo deslocamento de fase para todos os componentes do sinal.
- Quando a resposta em fase é **quase linear** existe uma pequena diferença no deslocamento dos diferentes componentes do sinal.
- Finalmente, quando a resposta é **não linear** acontece um deslocamento significativamente heterogêneo para as diferentes frequências formantes do sinal.

Idealmente é desejável que todo filtro apresente boa resposta em frequência e resposta em fase linear.

Wavelet	**Resposta em frequência**	**Resposta em fase**
Haar	Pobre	Linear
Daubechies	Quanto maior o suporte, melhor. *Maximally-flat*	Não linear
Symmlets	Quanto maior o suporte, melhor. Não *Maximally-flat*	Quase linear
Coiflets	Quanto maior o suporte, melhor. Não *Maximally-flat*	Quase linear

Tabela 2.1: Algumas wavelets mais populares e suas propriedades

O algoritmo de Malat

O algoritmo de Malat torna aplicação das *wavelets* no sinal em uma simples multiplicação de matrizes, o sinal que deve ser transformado se torna uma matriz linear vertical já

os filtros passa-baixa e passa-alta tornam-se, nessa ordem, linhas de uma matriz quadrada que será completada segundo regras que serão mostradas mais adiante.

É importante que essa matriz quadrada tenha de aresta a mesma quantidade de itens que o nosso sinal, ou seja, se o sinal tem quatro elementos então a matriz de filtros deve ser uma de 4x4.

Algo interessante a se notar é que, para que seja possível a transformada wavelet, basta ter disponível o filtro passa-baixa construído a partir da *mother wavelet* já que o filtro passa-alta pode ser construído a partir da ortogonalidade do primeiro.

A título de exemplo considere:

O filtro passa baixa baseado na wavelet Haar: $h[\cdot] = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$,

E seu respectivo valor ortogonal: $g[\cdot] = [\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}]$,

Considere também o seguinte sinal: $sinal = [1, 2, 3, 4]$.

Se o tamanho do sinal a ser tratado é quatro, ou seja, o sinal tem quatro pulsos, e se pretende-se aplicar o filtro Haar, a seguinte matriz é construída:

$$\begin{pmatrix} \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0 \\ \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, 0 \\ 0, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \\ 0, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \end{pmatrix} \tag{2.9}$$

No entanto, filtros Haar tem apenas dois valores e, necessariamente, a linha da matriz deve ter quatro itens. Para resolver este problema basta completar cada uma das linhas com zeros. A matriz é montada de forma que a mesma seja ortogonal.

Montada a matriz de filtros segue-se com os cálculos da transformada:

$$\begin{pmatrix} \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0 \\ \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, 0 \\ 0, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \\ 0, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} \frac{3}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \\ \frac{7}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix} \tag{2.10}$$

realizada a multiplicação é necessário agora montar o sinal filtrado, isso é feito esco-

lhendo, dentro do resultado, valores alternadamente de forma que o vetor resultante seja:

$$resultado = [\frac{3}{\sqrt{2}}, \frac{7}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}] \quad . \tag{2.11}$$

2.1.3 Amostragem, quantização e o formato do arquivo Wave

Serão usados arquivos no formato *wave* usando *pulse-code modulation* (PCM), neste esquema os dados são armazenados sem perdas. O arquivo, segundo [Sap19], se estrutura como o ilustrado na figura 2.6.

A taxa de amostragem de 44100hz permite, segundo o teorema de Nyquist, que seja realizada a quantização de frequências de até 22050hz a uma resolução de 16bits.

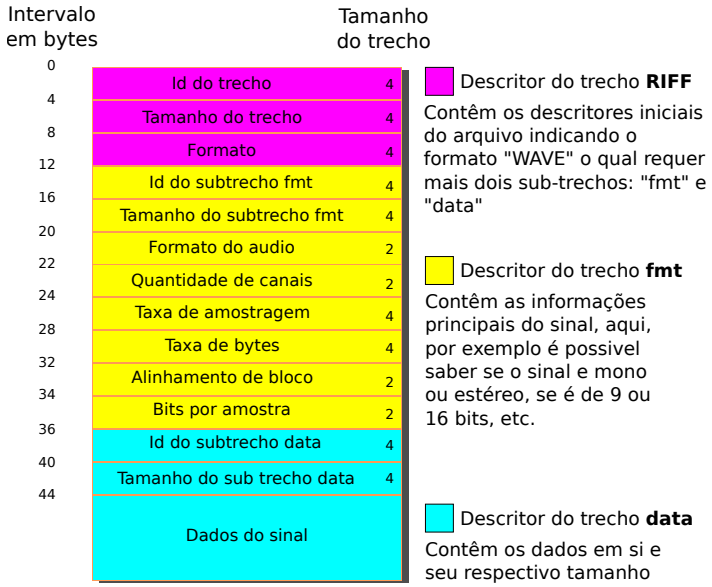

Figura 2.6: Estrutura do arquivo Wave

A estrutura de interesse se localiza na última parte do arquivo, mais especificamente no bloco "data", aqui os dados são organizados como um grande vetor de números, cada um deles, indicando a intensidade do sinal naquele ponto.

2.1.4 Caracterização dos processos de produção da voz humana

A fala possui três grandes áreas de estudo: fisiológica (ou fonética articulatória), acústica (ou fonética acústica) e perceptual (ou comumente chamada percepção da fala) [KG14].

Neste trabalho o foco será apenas na acústica, já que não serão analisados aspectos da fisiologia relacionada a voz e sim o sinal sonoro propriamente dito.

Vozeada versus não-vozeada

Quando da análise de voz se pode levar em consideração as partes vozeadas e/ou não-vozeadas do sinal. As partes vozeadas são aquelas produzidas com ajuda das pregas vocais, as partes não-vozeadas não tem participação desta estrutura.

Frequência fundamental da voz

Também conhecida como f_0 é o componente periódico resultante da vibração das pregas vocais, em termos de percepção se pode interpretar f_0 como o tom da voz (pitch) [KG14].

Vozes agudas tem um pitch alto, vozes mais graves tem um pitch baixo, a alteração do pitch durante a fala é definido como entonação.

A frequência fundamental da voz é a velocidade na qual uma forma de onda se repete por unidade de tempo, ou seja, o número de ciclos vibratórios produzidos pelas pregas vocais, num segundo, sendo assim, as medidas de f_0 geralmente são apresentadas em Hz [Fre13].

A medição de f_0 está sujeita a contaminações surgidas das variações naturais de *pitch* tipicas da voz humana [Fre13].

A importância de se medir f_0 corretamente vem do fato de que, além de carregar boa parte da informação da fala, f_0 é a base para construção das outras frequências, pois essas são múltiplas de f_0.

Formantes

O primeiro formante (f_1), relaciona-se à amplificação sonora na cavidade oral posterior e à posição da língua no plano vertical; o segundo formante (f_2) à cavidade oral anterior e à posição da língua no plano horizontal. O terceiro formante (f_3) relaciona-se às cavidades à frente e atrás do ápice da língua; o quarto formante (f_4), ao formato da laringe e da faringe na mesma altura [V+14].

2.2 Trabalhos correlatos

No artigo [RFLC19] foi apresentado um esquema de diferenciação entre a fala comum e aquela vinda de um dispositivo reprodutor. O foco da análise se dá na distorção causada pelo alto-falante segundo a energia e outras várias características do espectro do sinal. Uma base com 771 sinais de fala foi criada para cada um dos quatro dispositivos de gravação usados totalizando 3084 trechos de áudio. Uma *support vector machine* (SVM) foi usada como classificador. De acordo com os experimentos a *taxa de verdadeiros positivos* é de 98,75% e a *taxa de verdadeiros negativos* é de 98,75%.

Em [YXWW19] é mostrado um método para diferenciar a voz de um locutor verdadeiro da voz gerada por sistemas usando sintetizadores baseados no *modelo oculto de Markov* (HMM). SAS[YJ19] foi a escolha de base de dados. Este método usa coeficientes de características logarítmicos extraídos de wavelets que são apresentados a um classificador SVM. Os resultados obtidos tiverem, em média, mais de 99% de acurácia.

Usando uma decomposição por espalhamento baseada em wavelets e convertendo o resultado em coeficientes cepstrais (SCCs) o artigo [SSAL17] cria um vetor de características que é avaliado por modelos de mistura Gaussiana (GMM). SAS e ASVspoof 2015 [ZW15] foram as bases de dados escolhidas para testes. Em relação aos resultados foram usadas a *taxa de falsos verdadeiros* (FAR) que representa a taxa de ocorrências falsas classificadas como verdadeiras e a *taxa de falsos falsos* (FRR) que é a taxa de ocorrências verdadeiras classificadas como falsas. Aos pontos em que FAR é igual a FRR chamou-se de pontos de taxa de erros iguais e a *taxa de erros iguais* (ERR) é o valor de $\frac{FAR}{FRR}$.

Considerando isso nos experimentos foi obtida uma ERR geral de 0,18.

Já em [AV19] os autores usam o *"Zero time windowing"* ou janelamento de tempo zero (ZTW), conceito esse que deve ser melhor entendido durante a confecção da dissertação, para, em conjunto com a análise cepstral do espectro gerado, fazer a análise do sinal. Os experimentos foram feitos usando-se a base ASVspoof 2017[TK17] com um classificador GMM, a taxa geral de ERR dos experimentos foi de 0,1475.

Em [YLYL19] é citado que existe uma diferença entre as propriedades espectrais da voz original e da voz gravada. No escrito são usados coeficientes cepstrais sobre os quais são aplicados uma média e uma normaliza de variância para diminuir o impacto dos ruídos na classificação. Uma GMM foi usada como classificador. A base de dados usada é a ASVspoof 2017. Quanto aos resultados se obteve uma EER geral menor que 0,1.

A proposta de [Han18] é usar sinais residuais de predição linear, para, juntamente com coeficientes cepstrais criar características que serão apresentas a um classificador GMM. Novamente, a base de dados usada foi a ASVspoof 2015 e os resultados em ERR geral foram de 5,249.

Para detecção de voice spoofing [RBABA19] importa do campo de processamento de imagens o conceito de textura, para o processamento de voz esse conceito é chamado de "texturas de voz". Padrões binários locais (LBP) e seu respectivo histograma são usados para a construção do vetor de características que será avaliado por uma SVM. A base de dados usada para testes foi a ASVspoof 2015. A taxa máxima de acurácia conseguida foi de 0,7167.

Uma abordagem que combina análise de sinal de fala usando a *transformada de constante Q* (CQT) com o processamento cepstral é mostrada em [TDE17], essa técnica resulta no que se chama *coeficientes cepstrais de constante Q*(CQCCs), segundo o artigo, a vantagem destes coeficientes é a resolução espectro temporal variável. As base de dados usadas foram a RedDots [Pro], ASVspoof 2015 e AVSpoof. Foram usados três classificadores:

- DA-IICT: Uma fusão de dois classificadores GMM, sendo que um deles usa *coeficientes cepstrais de frequência MEL*(MFCC) e o outro usa características CFCC-IF

[PP15].

- STC [NKL+16].
- SJTU [KMM+16].

Na seção de experimentos são feitos testes para cada uma das bases com os seguintes resultados: ASVspoof 2015 → EER geral de 0.026; AVspoof → EER geral de 0; RedDots → EER geral de 0,185.

O artigo [SPM18] propõe uma aproximação usando reverberação e as partes não vozeadas da fala, três GMMs foram definidos para a classificação, esses classificadores votam se uma ocorrência é ou não verdadeira, ganhado sempre a classificação que obtiver mais votos. A base de dado utilizada foi a ASVSpoof 2017. O sistema de avaliação de desempenho escolhido, novamente, foi o ERR e esta alcançou um valor de 2,99.

A principal ideia de [KP18] é capturar a amplitude instantânea vinda de flutuações instantâneas de energia. Segundo o artigo as modulações de amplitude são mais suscetíveis ao ruído inserido no sinal original por uma fonte reprodutora. O estudo usa a base de dados ASVSpoof 2017 e GMM como classificador. Os resultados apresentados chegaram a uma EER de 0.0019.

No trabalho [WIAE18] foram usadas as diferenças entre bandas de frequências específicas para diferenciar um sinal legítimo de um usado em ataques de falsificação. Neste trabalho é proposta a *predição linear em domínio de frequência*(FDLP) juntamente com GMMs para classificação dos dados presentes na base ASVspoof 2017. Os resultados apresentados chegaram a uma EER de 0.0803.

Em [SSWA18] se propõe duas novas características que visam interpretar as componentes estáticas e dinâmicas do sinal, essas características complementam as características de tempo restrito no espectro. São elas a *"modulation spectral centroid frequency"* e a *long term spectral average*. O sistema usa como classificador um GMM juntamente com a base dados ASVSpoof 2017. Os resultados chegaram a um valor de EER de 0,0654.

Considerando o envelopamento das amplitudes e das frequências instantâneas em cada banda estreita filtrada, [KP17] discute como diferenciar um sinal legítimo de um falso.

A base de dados usada foi a ASVSpoof 2015. Um GMM foi usado como classificador e, em relação ao desempenho, o método chegou a ter um EER de 0,045.

Neste trabalho [DGK+16] é proposto o uso do *gammatone frequency cepstral coefficients*(MGFCC). O gammatone é o produto de uma distribuição gamma com um sinal senoide e é usado na construção de filtros auditivos que, neste caso, são usados para extrair características do sinal de voz. A base de dados usada foi a ASVspoof 2015. O classificador usado foi um GMM e o EER chegou a 0,02556.

Segundo [AKY+18] *Hashing* sensível a locus(LSH) é frequentemente usado como um classificador para problemas relacionados a *big data*, neste trabalho é proposto uma junção de MFCC e LSH a fim de se reconhecer o locutor. Neste método o MFCC é extraído dos arquivos de sinal para posterior aplicação do LSH gerando assim uma tabela *hash*, estes valores de *hash* são então comparados identificando assim o locutor ou locutora. Nos testes realizados houve uma acurácia de 92,66%. A base de dados usada foi a TIMIT 2018 [Con18].

2.2.1 Contextualização

No trabalho proposto para dissertação de mestrado do autor a intenção é encontrar um conjunto de características que demonstrem ser as mais disjuntas possíveis para fins de separação entre as classes, com o objetivo de melhorar a acurácia de classificadores para detecção de ataques de *voice spoofing*. Caraterísticas essas com base na transformada *wavelet*, devido a sua boa resolução em relação às dimensões de tempo e frequência. Essas características serão avaliadas usando a análise paraconsistente de acordo com o trabalho [Gui19] recentemente publicado.

Capítulo 3

Cronograma para conclusão do curso de mestrado

Até a presente data foram realizados os primeiros levantamentos para construção da base de dados com as vozes que serão objeto de pesquisa. Essa base conta com um total de 21 gravações originais e outras 21 de *playback* gravadas de pessoas com variados gêneros e idades pronunciando os dígitos de zero a nove em Inglês. A ideia é que essa base possa crescer e abarcar boa parte dos tipos de vozes existentes na região geográfica próxima ao campus da UNESP de São José do Rio Preto. Também foram estudados e implementados filtros passa baixas e passa altas, além de testadas metodologias de criação de vetores de características usando os intervalos espectrais pré-definidos pelas bandas MEL e BARK.

Foi melhorada a biblioteca criada e fornecida pelo orientador deste trabalho, a qual facilita a manipulação dos arquivos de áudio no formato *wave*. Do mesmo modo, código-fonte para análise paraconsistente de características foi também desenvolvido e complementado com a devida documentação, a qual foi norteada por instruções recebidas do orientador.

Quanto aos trabalhos futuros a tabela 3.1 mostra o cronograma previsto.

09/03 – 31/03	Início dos trabalhos: Coleta de dados para constituição da base de dados e estudo da base bibliográfica. Escrita da dissertação.
01/04 – 15/04	Coleta de dados para constituição da base de dados e estudo da base bibliográfica. Início dos experimentos. Escrita da dissertação.
16/04 – 16/04	Reunião de validação com o orientador.
17/04 – 29/04	Experimentos e escrita da dissertação.
30/04 – 30/04	Reunião de validação com o orientador.
01/05 – 20/05	Experimentos e escrita da dissertação.
21/05 – 28/05	Reunião de validação com o orientador.
29/05 – 10/06	Experimentos e escrita da dissertação.
11/06 – 11/06	Reunião de validação com o orientador.
12/06 – 17/06	Experimentos e escrita da dissertação.
18/06 – 18/06	Reunião de validação com o orientador.
19/06 – 24/06	Experimentos e escrita da dissertação.
25/06 – 25/06	Reunião de validação com o orientador.
26/06 – 01/07	Experimentos e escrita da dissertação.
02/07 – 02/07	Reunião de validação com o orientador.
03/07	Entrega da dissertação.

Tabela 3.1: Cronograma

Bibliografia

[AKY+18] A. Awais, S. Kun, Y. Yu, S. Hayat, A. Ahmed, and T. Tu. Speaker recognition using mel frequency cepstral coefficient and locality sensitive hashing. In *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pages 271–276, May 2018.

[AV19] KNRK Raju Alluri and Anil Kumar Vuppala. Replay spoofing countermeasures using high spectro-temporal resolution features. *International Journal of Speech Technology*, 22(1):271–281, 2019.

[AWG09] P. S. Addison, J. Walker, and R. C. Guido. Time–frequency analysis of biosignals. *IEEE Engineering in Medicine and Biology Magazine*, 28(5):14–29, Sep. 2009.

[Con18] Linguistic Data Consortium. Timit acoustic-phonetic continuous speech corpus, 2018.

[DGK+16] K. Arun Das, Kuruvachan K. George, C. Santhosh Kumar, S. Veni, and Ashish Panda. Modified Gammatone Frequency Cepstral Coefficients to Improve Spoofing Detection. pages 50–55, 345 E 47TH ST, NEW YORK, NY 10017 USA, 2016. LNM Inst Informat Technol; IEEE Commun Soc; IEEE Syst Man & Cybernet Soc, IEEE. International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, INDIA, SEP 21-24, 2016.

[Fre13] Susana Freitas. Avaliação acústica e áudio percetiva na caracterização da voz humana. 2013.

[Gui19] R. C. Guido. Paraconsistent feature engineering [lecture notes]. *IEEE Signal Processing Magazine*, 36(1):154–158, Jan 2019.

[Han18] Cemal Hanilçi. Linear prediction residual features for automatic speaker verification anti-spoofing. *Multimedia Tools and Applications*, 77(13):16099–16111, Jul 2018.

[KG14] Robinson Luis Kremer and ML d C GOMES. A eficiência do disfarce em vozes femininas: uma análise da frequência fundamental. *ReVEL*, 12:23, 2014.

[KMM+16] Pavel Korshunov, Sébastien Marcel, Hannah Muckenhirn, André R Gonçalves, AG Souza Mello, RP Velloso Violato, Flávio O Simoes, M Uliani Neto, Marcus

de Assis Angeloni, José Augusto Stuchi, et al. Overview of btas 2016 speaker anti-spoofing competition. pages 1–6. IEEE, 2016.

[KP17] Madhu R. Kamble and Hemant A. Patil. Novel Energy Separation Based Frequency Modulation Features For Spoofed Speech Classification. pages 326–331, 2017. 9th International Conference on Advances in Pattern Recognition (ICAPR), Indian Stat Inst Bangalore, Bangalore, INDIA, DEC 27-30, 2017.

[KP18] Madhu R. Kamble and Hemant A. Patil. Novel Variable Length Energy Separation Algorithm using Instantaneous Amplitude Features For Replay Detection. Interspeech, pages 646–650. Int Speech Commun Assoc, 2018. 19th Annual Conference of the International-Speech-Communication-Association (INTERSPEECH 2018), Hyderabad, INDIA, AUG 02-SEP 06, 2018.

[NKL+16] S. Novoselov, A. Kozlov, G. Lavrentyeva, K. Simonchik, and V. Shchemelinin. Stc anti-spoofing systems for the asvspoof 2015 challenge. pages 5475–5479, March 2016.

[P+96] Robi Polikar et al. The wavelet tutorial, 1996.

[PP15] Tanvina Patel and Hemant Patil. Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech. 09 2015.

[Pro] RedDots Project. Reddots database.

[RBABA19] Raoudha Rahmeni, Anis Ben Aicha, and Yassine Ben Ayed. On the contribution of the voice texture for speech spoofing detection. pages 501–505, 345 E 47TH ST, NEW YORK, NY 10017 USA, 2019. IEEE.

[RFLC19] Yanzhen Ren, Zhong Fang, Dengkai Liu, and Changwen Chen. Replay attack detection based on distortion by loudspeaker for voice authentication. *Multimedia Tools and Applications*, 78(7):8383–8396, Apr 2019.

[Sap19] Craig Stuart Sapp. Wave pcm soundfile format, 2019.

[SPM18] M. S. Saranya, R. Padmanabhan, and Hema A. Murthy. Replay Attack Detection in Speaker Verification Using non-voiced segments and Decision Level Feature Switching. International Conference on Signal Processing and Communications SPCOM, pages 332–336. IEEE, 2018. 12th International Conference on Signal Processing and Communications (SPCOM), Indian Inst Sci, Bangalore, INDIA, JUL 16-19, 2018.

[SSAL17] K. Sriskandaraja, V. Sethu, E. Ambikairajah, and H. Li. Front-end for antispoofing countermeasures in speaker verification: Scattering spectral decomposition. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):632–643, June 2017.

[SSWA18] Gajan Suthokumar, Vidhyasaharan Sethu, Chamith Wijenayake, and Eliathamby Ambikairajah. Modulation dynamic features for the detection of replay attacks. pages 691–695, 2018.

[TDE17] Massimiliano Todisco, Héctor Delgado, and Nicholas Evans. Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. *Computer Speech & Language*, 45:516 – 535, 2017.

[TK17] Junichi Yamagishi et al Tomi Kinnuneni, Nicholas Evans. Asvspoof 2017, 2017.

[V+14] Eugênia Hermínia Oliveira Valença et al. Análise acústica dos formantes em indivíduos com deficiência isolada do hormônio do crescimento. 2014.

[WIAE18] Buddhi Wickramasinghe, Saad Irtza, Eliathamby Ambikairajah, and Julien Epps. Frequency Domain Linear Prediction Features for Replay Spoofing Attack Detection. Interspeech, pages 661–665, C/O EMMANUELLE FOXONET, 4 RUE DES FAUVETTES, LIEU DIT LOUS TOURILS, BAIXAS, F-66390, FRANCE, 2018. Int Speech Commun Assoc, ISCA-INT SPEECH COMMUNICATION ASSOC. 19th Annual Conference of the International-Speech-Communication-Association (INTERSPEECH 2018), Hyderabad, INDIA, AUG 02-SEP 06, 2018.

[YJ19] et al Yamagishi Junichi, Todisco Massimiliano. Spoofing and anti-spoofing (sas) corpus, 2019.

[YLYL19] Y. Ye, L. Lao, D. Yan, and L. Lin. Detection of replay attack based on normalized constant q cepstral feature. pages 407–411, April 2019.

[YXWW19] Diqun Yan, Li Xiang, Zhifeng Wang, and Rangding Wang. Detection of hmm synthesized speech by wavelet logarithmic spectrum. *Automatic Control and Computer Sciences*, 53(1):72–79, Jan 2019.

[ZW15] Nicholas Evans Junichi Yamagishi Zhizheng Wu, Tomi Kinnunen. Asvspoof 2015, 2015.