UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

WELLITON JHONATHAN LEAL BABINSKI

CLASSIFICAÇÃO AUTOMÁTICA DE INSTRUMENTOS MUSICAIS UTILIZANDO UMA ABORDAGEM DE CLASSIFICAÇÃO MULTIVARIADA DE SÉRIES TEMPORAIS

PATO BRANCO

2022

WELLITON JHONATHAN LEAL BABINSKI

CLASSIFICAÇÃO AUTOMÁTICA DE INSTRUMENTOS MUSICAIS UTILIZANDO UMA ABORDAGEM DE CLASSIFICAÇÃO MULTIVARIADA DE SÉRIES TEMPORAIS

AUTOMATIC MUSICAL INSTRUMENTS CLASSIFICATION USING A MULTIVARIATE TIME SERIES CLASSIFICATION APPROACH

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do título de Bacharel em Engenharia de Computação da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Dalcimar Casanova
Coorientador: Prof. Dr. Rafael Cardoso

PATO BRANCO

2022

WELLITON JHONATHAN LEAL BABINSKI ✉

CLASSIFICAÇÃO AUTOMÁTICA DE INSTRUMENTOS MUSICAIS UTILIZANDO UMA ABORDAGEM DE CLASSIFICAÇÃO MULTIVARIADA DE SÉRIES TEMPORAIS

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do título de Bacharel em Engenharia de Computação da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de Aprovação: 27 de junho de 2022.

Prof. Dr. Dalcimar Casanova
Universidade Tecnológica Federal do Paraná

Prof. Dr. Rafael Cardoso
Universidade Tecnológica Federal do Paraná

Prof. Dr. Gustavo Weber Denardin
Universidade Tecnológica Federal do Paraná

Prof. Dr. Jefferson Tales Oliva
Universidade Tecnológica Federal do Paraná

PATO BRANCO

2022

Dedico este trabalho as minhas amadas mãe, avó e irmã, que são a base que me sustenta, a força que me empurra e a energia que me mantém em movimento.

AGRADECIMENTOS

Agradeço primeiramente a minha mãe Miriam Leal, avó Enoeli Leal e irmã Ketlin Leal, pois sempre e estiveram ao meu lado dando todo o suporte possível, apoiando e incentivando em todos os momentos. Também sou grato aos demais familiares por sempre acreditarem em mim durante toda essa longa trajetória.

Aos grandes amigos Rafael Anderson Dalmolin, Emanuel de Cesaro, Gabriel Dalla Vechia, Jordan Chan, Marcelo Miguel Peluso, Gabriel Salvatti, Wesley Almeida, Elivelto Sauzen Muller e Cristian Pastro, com quem convivi e compartilhei vários momentos ao longo desses anos tornardo esse período mais leve e divertido.

Aos meus orientadores Dalcimar Casanova e Rafael Cardoso por todo suporte, direcionamento, ensinamentos, pela confiança depositada e também por aceitarem me conduzir neste trabalho de conclusão de curso.

À Universidade Tecnológica Federal do Paraná – UTFPR pela oportunidade e qualidade de ensino oferecida, essencial no meu processo de formação pessoal e profissional.

"O importante é não parar de questionar. A curiosidade tem sua própria razão de existir. Não se pode deixar de se maravilhar ao contemplar os mistérios da eternidade, da vida, da maravilhosa estrutura da realidade. Basta tentar apenas compreender um pouco deste mistério todos os dias." (EINSTEIN, 1955, tradução)[1].

[1] "*The important thing is not to stop questioning. Curiosity has its own reason for existing. One cannot help but be in awe when he contemplates the mysteries of eternity, of life, of the marvelous structure of reality. It is enough if one tries merely to comprehend a little of this mystery every day.*" (EINSTEIN, 1955).

RESUMO

Este trabalho apresenta o desenvolvimento de um algoritmo essencial para sistemas inteligentes de mixagem e reparo automático de áudio, é um modelo de aprendizado de máquina que classifica automaticamente instrumentos musicais utilizando de uma abordagem de classificação multivariada de séries temporais. A partir de dois bancos de dados de sinais de áudio digital com diversas amostras de instrumentos musicais, o objetivo inicial foi extrair características temporais e espectrais instantâneas ao longo do tempo, que são representadas por séries temporais. As características são utilizadas para treinar de modelos de aprendizado de máquina supervisionado, que identificam padrões de séries temporais utilizando de algoritmos adaptados para essa tarefa, como o *K-Nearest Neightbours* em conjunto com o algoritmo de alinhamento *Dynamic Time Warping*, ou o *Support Vector Machines* em conjunto com o algoritmo de alinhamento *Global Alignment Kernel*. O objetivo final foi utilizar esses modelos para realizar a tarefa de classificação de novas bases de sinais de áudio de instrumentos musicais desconhecidos, como também realizar análises para entender quais efeitos diferentes durações de sinais e parâmetros de extração de características tem sobre os resultados.

Palavras-chave: classificação; instrumentos musicais; sinais de áudio; séries temporais; aprendizado de máquina.

ABSTRACT

This work presents a development of an essential algorithm to automatic audio mixing and repair intelligent systems, is a machine learning model that classifies automatically musical instruments using a multivariate time series classification approach. Starting from two databases of audio digital signals with various musical instruments samples, the first goal is to extract both temporal and spectral instantaneous features, which are represented by temporal series. The series are used to train the supervised machine learning models, that are responsable to identify patterns of temporal series using adapted algorithms to do this task, like the K-Nearest Neightbours with the Dynamic Time Warping algorithm, or the Support Vector Machines with Global Alignment Kernel algorithm. The main goal is to utilize this models to do the task of feature classification of new databases of unknow musical instruments audio signal, as well to analysis and understand which effects different signal durations and feature extraction parameters have in the results.

Keywords: classification; musical instruments; audio signals; time series; machine learning.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

ABREVIATURAS

ins.	Instrumento

SIGLAS

ADC	Conversor Analógico-Digital, do Inglês *Analog-to-Digital Converter*
AE	Envelope de Amplitude, do Inglês *Amplitude Envelope*
BW	Largura de Banda, do Inglês *Bandwidth*
CPU	Unidade Central de Processamento, do Inglês *Central Process Unit*
DAW	Estação de Trabalho de Áudio Digital, do Inglês *Digital Audio Workstation*
DFT	Transformada Discreta de Fourier , do Inglês *Discrete Fourier Transform*
DSP	Processamento de Sinal Digital, do Inglês *Digital Signal Processing*
DTW	Distorção Temporal Dinâmica, do Inglês *Dynamic Time Warping*
FFT	Transformada Rápida de Fourier , do Inglês *Fast Fourier Transform*
GAK	Kernel de Alinhamento Global, do Inglês *Global Alignment Kernel*
IMP	Produção Musical Inteligente, do Inglês *Intelligent Music Production*
k-NN	K-Vizinhos Mais Próximos, do Inglês *K-Nearest Neighbors*
LDA	Análise Discriminate Linear, do Inglês *Linear Discriminant Analysis*
MFCC	Coeficientes Cepstrais de Frequência Mel, do Inglês *Mel-Frequency Cepstral Coefficients*
MIDI	Interface Digital de Instrumentos Musicais, do Inglês *Musical Instrument Digital Interface*
MIR	Recuperação de Informação Musical, do Inglês *Music Information Retrieval*
PCA	Análise de Componentes Principais, do Inglês *Principal Component Analysis*
RMS	Raiz Quadrada Média, do Inglês *Root-Mean-Square*
SC	Centróide Espectral, do Inglês *Spectral Centroid*
STFT	Transformada de Fourier de Curta Duração, do Inglês *Short-Time Fourier Transform*
SVM	Máquina de Vetores de Suporte, do Inglês *Support Vector Machine*
TFR	Representação Tempo-Frequência, do Inglês *Time-Frequency Representation*
UTFPR	Universidade Tecnológica Federal do Paraná
VST	Tecnologia de Estúdio Virtual, do Inglês *Virtual Studio Technology*
ZCR	Taxa de Cruzamentos no Zero, do Inglês *Zero-Crossing Rate*

SUMÁRIO

1 INTRODUÇÃO

A indústria fonográfica já passou por diversas mudanças seguindo o fluxo da evolução tecnológica. A maior delas pode ser considerada a migração para a era digital, proporcionada desde meados dos anos 80 até os dias de hoje. Ela ocorreu devido ao aumento do poder computacional e capacidade de processamento de informações, o que tornou possível que várias ferramentas analógicas de áudio, fossem emuladas em *softwares* de forma virtual. Assim também surgiram novas ferramentas e aplicações graças aos esforços de pesquisa e desenvolvimento nos campos de estudo de processamento de sinais, musical e de fala.

Para uma melhor contextualização é necessário pontuar também alguns desenvolvimentos muito importantes no áudio digital. Como na década de 90, quando uma pioneira no desenvolvimento de *plugins* de áudio deu um importante passo, a Steinberg lançou em 1996 o Cubase, um *software* que permitia processar áudio em tempo real usando a CPU do computador (IZHAKI, 2012, p. 13). Desde então, a maneira como a música era produzida, gravada, mixada e masterizada mudou radicalmente, passou a não ser mais necessário contratar estúdios ou ter vários equipamentos de alto custo para se gravar e produzir uma música com boa qualidade sonora. Além disso, depois que essa praticidade ficou aparente, o processamento de sinais de áudio digital em *softwares* de computador, hoje conhecidos como *Digital Audio Workstation* (DAW), possibilitou o desenvolvimento de novas técnicas de transformação de sons, que no domínio analógico eram impraticáveis ou de qualidade inferior (DEAN *et al.*, 2011, p. 21).

Sinais de áudio no domínio digital tornam-se dados, e nesse formato ficam indistinguíveis de qualquer outro do mesmo tipo, o que permitiu sistemas e técnicas desenvolvidas em outros setores e para outros motivos serem usados no áudio (WATKINSON, 2001, p. 9). No início de 2000, a pesquisa musical ainda era recente, na sua maioria realizada principalmente com base em representações simbólicas, usando notação musical ou representações MIDI (*Musical Instrument Digital Interface*) (MÜLLER, 2015, Preface). Porém, com o aumento do poder computacional dessa década, como a melhora da capacidade de armazenamento e conexões de banda larga, foi gerado uma grande disponibilidade de áudio digitalizado, esses eventos proporcionaram novas oportunidades e evidenciaram a possibilidade de sistemas automáticos para analisar o conteúdo de áudio, resultando em uma crescente animação no foco dos esforços de pesquisa (LERCH, 2012, p. 2).

Nos últimos anos, o ramo do áudio adotou fortemente ferramentas orientadas a dados na sua área de pesquisa, uma das subáreas é relativamente nova e tem sido chamada de *Intelligent*

Music Production (IMP), focada em utilizar inteligência artificial na produção musical. A área de estudo em questão tem apresentado diversas novas contribuições e aplicações comerciais na indústria fonográfica. Empresas como iZotope, CEDAR Audio e Accusonus estão liderando a aplicação do aprendizado de máquina em produtos de áudio. Existe uma grande crescente no surgimento novos de sistemas inteligentes, ferramentas de alta controlabilidade, configurações automáticas de parâmetros e aplicações de aprendizado de máquina em produtos de áudio comerciais, como nos DAW's e *plugins VST* (*Virtual Studio Technology*). Essas abordagens usam técnicas do conhecimento da engenharia, psicoacústica, avaliação perceptual e aprendizado de máquina para automatizar vários aspectos do processo de produção musical no geral (DE MAN; STABLES; REISS, 2019, p. 3). Como exemplo, durante a produção de uma música nas etapas processamento, mixagem e masterização, tarefas executadas tradicionalmente por um engenheiro de áudio especializado no processo, agora podem contar com a possibilidade de apenas uma intervenção mínima do mesmo. Essas tarefas estão sendo lentamente substituídas por ferramentas baseadas em inteligência artificial, que fazem uso de algoritmos baseados em combinações de modelos estatísticos, redes neurais, entre outros (ALESSIO, 2019).

Na grande maioria desses *softwares* comerciais que estão se tornando muito populares e são chamados de *plugins* inteligentes, uma ferramenta em específico se destaca, explícita ou implicitamente ela faz parte de uma etapa essencial nos algoritmos por trás das interfaces modernas, é o classificador automático de instrumentos musicais, seja ele feito com abordagens de aprendizado de máquina supervisionado ou com abordagens não supervisionadas. Por se tratarem de *softwares* utilizados exclusivamente e estensivamente na produção musical, com excessão de exemplos como vocais, efeitos especiais e sons ambientes, a grande maioria de sinais de áudio manipulados são provenientes da síntese sonora ou gravação de algum instrumento musical. Todos esses *plugins* realizam algum tipo de análise, processamento, sugestão de parâmetros e até correções, tudo depende da etapa de produção e propósito do *plugin*, mas em algum momento, seja na sua concepção ou na sua utilização em tempo real por um usuário final, eles utilizam um classificador automático de instrumentos musicais. É nesse contexto da produção musical inteligente que o problema deste trabalho se insere, a tarefa de classificação de sinais de áudio de instrumentos musicais é essencial para o propósito muitos *softwares* comerciais de processamento automático de áudio. Apesar de ser um campo de estudo já com bastante história, as pesquisas no ramo estão sempre sendo renovadas com novos algoritmos e técnicas utilizadas, assim como neste trabalho a proposta é utilizar de abordagens focadas em classificação multivariada de séries temporais para a realização da tarefa.

1.1 MOTIVAÇÃO

A maioria das ferramentas de processamento analógico de sinais de áudio do meio musical exigem intervenção manual, mas com o advento das novas tecnologias inteligentes, essa situação vem mudando. O objetivo de muitas ferramentas de *Intelligent Music Production (IMP*) é reduzir a carga de trabalho do profissional da área e explorar até que ponto várias tarefas podem ser automatizadas. O processo de mixagem de uma música, por exemplo, tem muitas aplicações imediatas conhecidas e aplicações variam de sistemas de mixagem completamente autônomos a ferramentas mais assistivas que aprimoram o fluxo de trabalho. Os limites entre essas categorias são vagos e a maioria dos sistemas podem ser adaptados para menos ou mais controle do usuário (DE MAN; STABLES; REISS, 2019, p. 30).

Um exemplo comercial no ramo da produção musical comentado por Alessio (2019), é a empresa iZotope e alguns de seus produtos. Nela, o aprendizado de máquina é usado para tarefas como identificar instrumentos musicais automaticamente, detectar estruturas musicais, reparar irregularidades causadas em gravação, pelo ambiente, por erro humano, como também para melhorar a navegação entre os sinais de áudio. O *plugin* de áudio Neutron 3 da empresa, contém um assistente de mixagem que ao usar aprendizado de máquina cria um ponto de partida para um processo de mixagem de nível inicial, economizando tempo, energia, e possibilitando que o usuário tome mais decisões criativas e menos decisões técnicas de mixagem. Esse assistente de mixagem faz parte de um conjunto de ferramentas que a empresa chama de *Assistive Audio Technology*, as quais analisam de uma forma inteligente seu áudio e fornecem sugestões como predefinições personalizadas, adaptadas sob medida ao som desejado. Resultado de anos de combinação de desenvolvimento de algoritmos inteligentes de *Digital Signal Processing (DSP)*, associado a técnicas modernas de aprendizado de máquina (SHAHAN, 2018).

Analisando de uma forma mais ampla ainda no contexto específico da mixagem de sinais de áudio, a classificação de instrumentos musicais tem sido usada em mais aplicações interessantes e a maioria delas são soluções comerciais privadas. O propósito dessas aplicações só é possível tendo o conhecimento básico inicial de quais instrumentos musicais se encontram nos sinais que serão processados, o que torna essa classificação dos sinais uma etapa fundamental nessas ferramentas inteligentes. Na sequência são listados tópicos referentes as plugins inteligentes, em que um classificador de instrumentos musicais é utilizado direta ou indiretamente no processo, algumas dessas também já foram comentadas por Lerch (2012, p. 3). Ao lado de cada tópico estão exemplos de plugins comerciais que tem esse tipo de aplicação comentada, as principais

funções e benefícios dessas ferramentas são:

- Controle inteligente, possibilitando maior quantidade e controlabilidade de parâmetros do processamento de áudio em e *plugins* (parâmetros como: de intensidade, velocidade, tempo, efeitos, tonalidade, *cross fades* inteligentes, etc). Exemplo: (IZOTOPE, 2022a).
- Automatização, predefinições, assistência e sugestões imediatas em processos padrões de mixagem, tornando o processamento de áudio digital menos suscetível ao erro humano (ajuda a evitar: suposições equivocadas, ouvidos cansados, má afinação, tratamento acústico ruim, etc). Exemplos: (IZOTOPE, 2022a) e (LEAPWING, 2022).
- Maior assertividade e facilidade na tomada de decisões de mixagem de instrumentos musicais por profissionais da área (decisões sobre: equilíbrio de níveis, de frequências, de posição e profundidade, etc). Exemplos: (IZOTOPE, 2022a) e (LEAPWING, 2022).
- Recuperação espectral de frequências, que estão com pouca energia ou intensidade tonal desejada no som característico de um instrumento musical. Reparo e remoção de ruídos e irregularidades causadas na gravação de um instrumento sendo tocado. Esses problemas ocasionados por motivos como ambiente acústico, ação humana e equipamentos (irregularidades como: ruídos de gravações em vinil, rangidos em cordas, reverberação em excesso, etc) Exemplo:(IZOTOPE, 2022b)
- Remoção completa de um instrumento musical ou grupo de instrumentos específicos, de uma música já consolidada em apenas uma faixa de áudio (remoção de: vocais, baterias, guitarra, baixo, etc). Exemplos: (IZOTOPE, 2022b) e (NICK, 2020).
- Organização automática do conteúdo de áudio de grandes bibliotecas, assim como pesquisa e recuperação de arquivos de áudio com características bem específicas (biblioteca de: *samples* de instrumentos, notas singulares, *loops* gravados, etc). Exemplos: (ALGONAUT, 2022) e (XLN, 2022).

Analisando esses exemplos, compreende-se uma das motivações da proposta deste trabalho, o classificador automático de instrumentos musicais está presente em vários *plugins* inteligentes da indústria fonográfica, mais especificamente na área da produção musical. Ele se faz muito importante em sistemas assistentes de mixagem automática, sistemas assistentes de reparo e organizadores de bibliotecas de áudio. Sua relevância se dá pelo fato de que para qualquer um desses tipos de *plugins* é fundamental para a o sucesso das suas aplicações, ter a informação prévia de quais instrumentos musicais estão nos sinais que serão posteriormente processados. Além das novas abordagens e novos parâmetros de avaliação, a aplicação desse

algoritmo classificador pelas ferramentas mencionadas, gera economia de tempo e energia impactando diretamente no trabalho de profissionais da indústria fonográfica, como também impacta indiretamente na qualidade musical entregue aos consumidores de música.

Segundo Weihs *et al.* (2016, p. 464), esses classificadores podem ser considerados um problema típico de classificação supervisionada do campo de aprendizado de máquina. O nível de dificuldade vai depender da aplicação real, no entanto, para todos os tipos de reconhecimento de padrões de instrumentos musicais, o principal desafio é uma escolha apropriada das características a serem extraídas e utilizadas. Klapuri e Davy (2006, p. 184) afirmaram que um problema que os pesquisadores geralmente se deparam ao projetar um classificador para sons de instrumentos musicais, é o de selecionar um único classificador para todas as classes de sons de instrumentos, ou inversamente, usar vários classificadores "focados" em diferentes características. No primeiro caso, os classificadores tradicionais são chamados de classificadores planos, ao passo que os classificadores do segundo caso podem ser uma das várias soluções diferentes, que vão desde classificadores hierárquicos até conjuntos de classificadores. Diversos trabalhos científicos já foram realizados abordando a tarefa e dentre essas abordagens comentadas na sequência, houve tanto trabalhos com foco explorando a aplicação de diferentes algoritmos ao problema, quanto explorando as mais variadas características de sinais.

Vários esquemas de características já foram propostos, nos trabalhos precursores classificando sinais de instrumentos, o foco desses foi puramente estatístico sobre os sinais como pode-se notar nas pesquisas de Kaminsky e Materka (1995), Martin e Kim (1998). Ao passar dos anos cada vez mais foram adicionadas novas características com interpretações temporais, espectrais e perceptuais, assim como as utilizadas por Agostini, Longari e Pollastri (2001), Herrera-Boyer, Peeters e Dubnov (2003). Estudos feitos por Deng, Simmermacher e Cranefield (2008), Chandwadkar e Sutaone (2012) demonstraram que dos vários esquemas de características individuais analisados, a *Mel-Frequency Cepstral Coefficients (MFCC)* (é uma unidade de medida de intensidade sonora, teve como objetivo construir uma escala que refletisse exatamente como as pessoas ouvem os tons musicais) foi a que apresentou o melhor desempenho de classificação. Essa é uma das mais utilizadas até os dias atuais pela grande qualidade de informação sobre o timbre de um instrumento, que em resumo, define a forma como humanos percebem dois instrumentos tocando notas musicais na mesma frequência, mas soando diferentemente.

Ao abordar como grande parte das características eram representadas, em geral na maioria dos trabalhos, independentemente de como foram extraídas, elas foram representadas por valores únicos globais, provavelmente devido a dependência de modelos classificadores

tradicionais. A diferença está no processo adotado, se esses valores foram provenientes de uma abstração estatística, como média, desvio padrão, variância, como as utilizadas por Eronen e Klapuri (2000) e Deng, Simmermacher e Cranefield (2008). Podem ser também coeficientes extraídos globalmente, como os *Linear Predictive Coding (LPC)* computados por Chetry e Sandler (2006), e também os MFCC's utilizados por Chandwadkar e Sutaone (2012). E por último, se o valor foi resultado de uma redução de dimensionalidade de uma série temporal, com técnicas de redução como *Principal Component Analysis (PCA)* e *Linear Discriminant Anaylis (LDA)* utilizadas respectivamente por Kaminsky e Materka (1995), Chakraborty e Parekh (2018).

Portanto, abstrações como métricas estatísticas, extração global ou redução de dimensionalidade, abstraem também os comportamentos das características ao longo tempo, consequentemente essa informação de "quando aconteceram" certos eventos significantes é também abstraída do modelo de classificação. Já nos trabalhos mais recentes, principalmente nos que utilizam técnicas de redes neurais, é comum a abordagem de extrair características perceptuais mas utilizar suas representações gráficas como característica de entrada de modelos. Pois essas imagens carregam consigo a informação da variação temporal intrínseca na sua matriz, elas são em geral espectrogramas, cepstrogramas, coeficientes, wavelets, entre outros.

Para mencionar alguns trabalhos que já relacionaram sinais de áudio digital a séries temporais e medidas de distância dinâmicas, os precursores vem da pesquisa de reconhecimento de fala. Foram Sakoe e Chiba (1978) que criaram o algoritmo *Dynamic Time Warping (DTW)* para o propósito do reconhecimento diferentes sinais de palavras faladas. Pikrakis, Theodoridis e Kamarotos (2003) utilizaram uma variação do algoritmo de DTW dependente de contexto para o reconhecimento do padrão de frequências fundamentais e específicamente no contexto de músicas tradicionais gregas. Mais tarde, Muda, Begam e Elamvazuthi (2010) fizeram uso da mesma técnica DTW para reconhecimento de fala, porém utilizando os MFCC's dos sinais. Já Esling e Agon (2013), propuseram uma forma inovadora de consultar bancos de dados genéricos de sinais de áudio otimizando simultaneamente o que foi chamado pelo autor de *multiobjective time series matching*, um técnica que mescla otimização multiobjetivo e correspondência de séries temporais. E por fim, Bhalke, Rama Rao e Bormane (2011) que fizeram uso de DTW pra classificar notas de instrumentos musicais, o que se aproxima um pouco da proposta deste trabalho, entretanto eles utilizaram uma abordagem univariada com apenas um tipo de característica (MFCC).

Os sinais áudio digital são um exemplo por excelência de dados representados por séries temporais, e estão no centro de muitos aplicativos de aprendizado de máquina do mundo real. Normalmente, interage-se com dados de séries de áudio em sua forma univariada no domínio

de tempo ou frequência, e isso apresenta uma oportunidade para avaliar novas abordagens que podem alavancar dimensões extras como acontecem nos casos multivariados (RUIZ *et al.*, 2021).

A partir de todo esse contexto analisado, fica evidente que a tanto a pesquisa sobre classificação de sinais de áudio digital quanto a de classifiação de séries temporais já tem muita história, estudos e abordagens aplicadas, inclusive também utilizando DTW como medida de similaridade para classificação univariada de séries. Como abordado, já existem alguns trabalhos aplicando técnicas univariadas ou medidas de similaridade em sinais de áudio, mas não especificamente realizando uma classificação multivariada, possívelmente porque a cada característica adicionada, aumenta também a dimensão do problema, o que requer maior uso de recurso computacional para tal tarefa. Também pelo fato de que os algoritmos com essa abordagem estão se tornando populares apenas nos últimos anos e até agora foram mais aplicados a problemas com séries temporais relativamente curtas, que não tem uma alta amostragem de um sinal de áudio musical. Portanto, a principal motivação deste trabalho é aplicar diferentes métodos de classificação multivariada de séries temporais, sobre a tarefa de classificação de sinais de instrumentos musicais.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O desenvolvimento de modelos de aprendizado de máquina supervisionados, para a tarefa de classificação automática de instrumentos musicais utilizando uma abordagem de classificação multivariada de séries temporais.

1.2.2 Objetivos Específicos

Além do objetivo principal dessa proposta, são propostos alguns objetivos específicos secundários para este trabalho, são eles:

- Analisar os modelos para diferentes durações dos sinais de áudio.
- Analisar os modelos para diferentes parâmetros de janelas de análise.
- Analisar os modelos para diferentes instrumentos musicais.
- Analisar os modelos para diferentes algoritmos propostos.

1.3 ESTRUTURA DO TRABALHO

Este trabalho está organizado da seguinte forma, no presente capítulo 1 foi introduzido o problema da tarefa de classificação de instrumentos musicais, o contexto a qual ela impacta e os objetivos a se alcançar propostos por este trabalho. Já no capítulo 2 são apresentados conceitos teóricos importantes os quais fazem um panorama geral passando por tópicos como instrumentos musicais, sinais de áudio digital e suas características, até os algoritmos de aprendizado de máquina utilizados. Na sequência no capítulo 3 além de elencar as tecnologias e ferramentas utilizadas, são abordados os métodos utilizados, e também a forma como são segmentadas e detalhadas todas as etapas necessárias para o implementação do trabalho. No capítulo 4 estão os resultados da implementação e também discussão de cada análise feita sobre os mesmos. Por fim, no capítulo 5 estão as conclusões obtidas sobre as análises e verificações dos resultados.

2 REFERENCIAL TEÓRICO

2.1 CONSIDERAÇÕES INICIAIS

Neste capítulo são evidenciadas algumas noções fundamentais necessárias para o melhor entendimento dos vários temas envolvidos no escopo geral deste trabalho, começando pelas definições de um instrumento musical e quais são suas categorizações na seção 2.2, seguindo com uma breve introdução sobre sinais de áudio digital e seus domínios de representações na seção 2.3. Na sequência são expostas as definições das características clássicas extraídas de sinais de áudio digital na seção 2.4, e por fim na seção 2.5 são abordados os conceitos teóricos dos dois métodos de aprendizado de máquina utilizados junto com suas respectivas medidas de distância a serem utilizadas.

2.2 INSTRUMENTOS MUSICAIS

Para a maioria das pessoas, reconhecer instrumentos musicais é considerada uma tarefa natural e inconsciente, já que eles são elementos simples de se perceber e fundamentais para compreendermos músicas dos mais variados gêneros e estilos musicais. Alguns instrumentos são mais facilmente identificados pela sua ampla utilização no meio musical, enquanto outros dependem muito do contexto fornecido pela música, como também pela cultura em que alguém se encontre envolvido, para só então ser possível realizar esse tipo de associação.

Uma das definições mais comuns de um instrumento musical é a de um instrumento acústico que é membro da orquestra clássica. Esses instrumentos podem ser classificados em famílias, e dentro de cada uma podem ser subdivididos em instrumentos com base no alcance das notas e nos meios articulatórios. Mas a definição de instrumento musical vai além da orquestra clássica e inclui instrumentos usados em músicas populares e étnicas, e mais recentemente instrumentos não acústicos, como o sintetizadores. Em todos os casos, está implícito que o instrumento musical é um objeto autônomo e produtor de som, que permite ao músico tocar em uma situação ao vivo. Os desenvolvimentos nos estilos estenderam o uso da palavra instrumento para incluir novos dispositivos musicais, às vezes desafiando a própria natureza da palavra (DEAN *et al.*, 2011, p. 236).

Conclui-se então que praticamente qualquer objeto que é usado para produzir som pode ser chamado de instrumento musical. Eles não se limitam às ferramentas que usamos para criar a

música em salas de concerto, teatros, salões de dança e outros lugares onde ouve-se música hoje, mas a todos os criadores de som, até mesmo os criadores de ruído que se estendem e derivam deles (MONTAGU, 2007, p. 2). Na Figura 1 estão alguns dos principais e mais conhecidos instrumentos utilizados pela orquestra clássica ocidental.

Figura 1 – Instrumentos musicais utilizados pela orquestra clássica ocidental

Fonte: (FREDERICK, 2019).

2.2.1 Categorização dos Instrumentos

Existem dois métodos de categorização que se destacam, são considerados os mais contemporâneos e também os mais utilizados. O primeiro é o sistema de Von Hornbostel e Sachs (1961), que utilizou da geometria dos instrumentos, seus materiais, suas peças e a forma como o som é produzido para construir uma taxonomia única de instrumentos musicais. Esse sistema pode ser aplicado a instrumentos de várias culturas e é o esquema de classificação mais amplamente aceito. Nesse sistema os instrumentos são classificados nos seguintes grupos: idiofones, membranofones, aerofones, cordofones e eletrofones.

O segundo é a categorização por relação de famílias, um dos sistemas destacados por Kartomi (1990), que surgiu do esquema da orquestra clássica ocidental, em que se classifica um instrumento dependendo do seu som, como o som é produzido e como o instrumento é projetado. Essa divisão em grande parte foi feita devido ao desenvolvimento histórico dos instrumentos,

inicialmente essa classificação era dividida nas famílias de cordas, sopros, percussões e metais. É importante notar que as famílias não são distinções bem definidas, pois nem todo instrumento se encaixa perfeitamente em uma família. Os pianos são um bom exemplo disso, ao mesmo tempo que eles tem cordas, eles também possuem martelos, então não é possível definir com clareza se pertence a família das cordas ou das percussões. Por esse motivo eles são geralmente separados em uma família própria de teclados, que generalizando mais ainda também é chamada de família de teclas. Nela incluem-se todos os tipos de instrumento que são tocados a partir de teclas, como orgãos, pianos, teclados, acordeons entre outros. Na tabela 1 abaixo é exibida a classificação pelo esquema das relações de famílias.

Tabela 1 – Relações de famílias de instrumentos musicais

Família	Termo em Inglês	Instrumentos
Instrumentos de Percussão	*Percussion*	Baterias, Tambores...
Instrumentos de Cordas	*Strings*	Violão, Violino, Cello...
Instrumentos de Sopro	*Woodwinds*	Flauta, Clarinete...
Instrumentos de Metal	*Brass*	Trombone, Trompete...
Instrumentos de Teclas	*Keys*	Teclado, Piano, Acordeon...

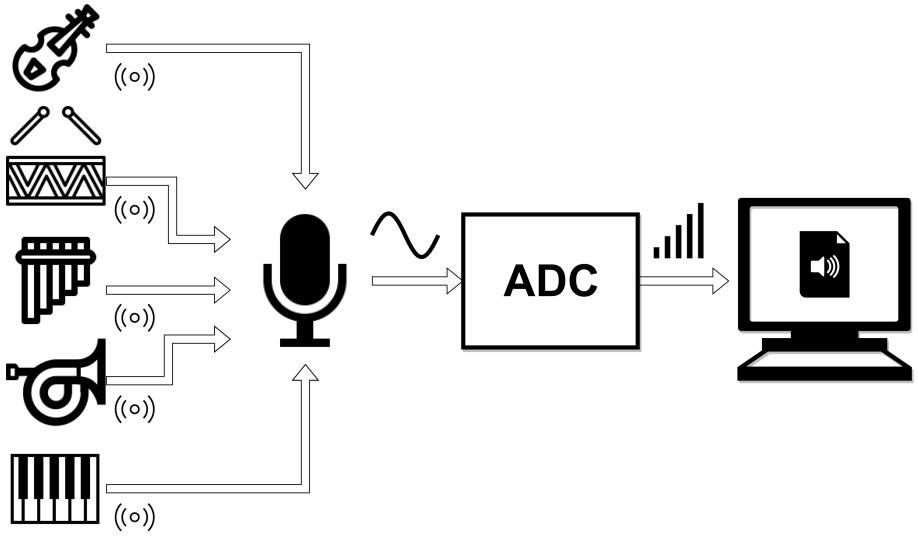
2.3 SINAIS DE ÁUDIO

Os sinais podem descrever uma grande variedade de fenômenos físicos. Embora os sinais possam ser representados de várias maneiras, em todos os casos a informação em um sinal, está contida em um padrão de variações de alguma forma (OPPENHEIM; WILLSKY; NAWAB, 1996, p. 1). Do ponto de vista físico, tocar instrumentos resulta em sons ou ondas acústicas, que são transmitidas pelo ar como oscilações de pressão. Quando um som é gerado por um objeto vibrante, como cordas de um instrumento ou até mesmo cordas vocais de um cantor, essas vibrações causam deslocamentos e oscilações das moléculas de ar, resultando em regiões locais de compressão e rarefação. Essa pressão alternada percorre o ar como uma onda, desde sua fonte até um ouvinte ou um microfone. Um sinal de áudio é uma representação de um som, e o termo “áudio” é usado para se referir à transmissão, recepção ou reprodução de sons que estão dentro dos limites da audição humana (20 Hz até 20000 Hz) (MÜLLER, 2015, p 19).

O processo de captação de som de um instrumento musical, até convertê-lo em sinal de áudio digital, em geral acontece em algumas etapas padrões. Primeiro, microfones são posicionados estratégicamente na direção do instrumento que será tocado. O microfone é

chamado de elemento transdutor do sistema, pois nele há uma membrana muito sensível, que vibra de acordo com a pressão do ar gerada pelo instrumento musical. Essa vibração captada é convertida em um sinal elétrico, que por sua vez é amplificado e enviado para um conversor analógico-digital (ADC). Já na segunda etapa, este conversor recebe o sinal analógico, que é contínuo no tempo e o converte em sinal digital que é discreto no tempo. Por fim, o sinal de áudio digital já está pronto para ser processado e armazenado na memória de um dispositivo, como um computador. Os arquivos quando armazenados geralmente estão nos tipos de formatos padrões de áudio, como os conhecidos "*wav*" e "*mp3*". Na Figura 2 é representada graficamente uma abstração do processo de conversão descrito acima, e na Figura 3, um exemplo da forma de onda de um sinal de áudio digital.

Figura 2 – Abstração simplificada de captação de som e conversão pra sinal de áudio



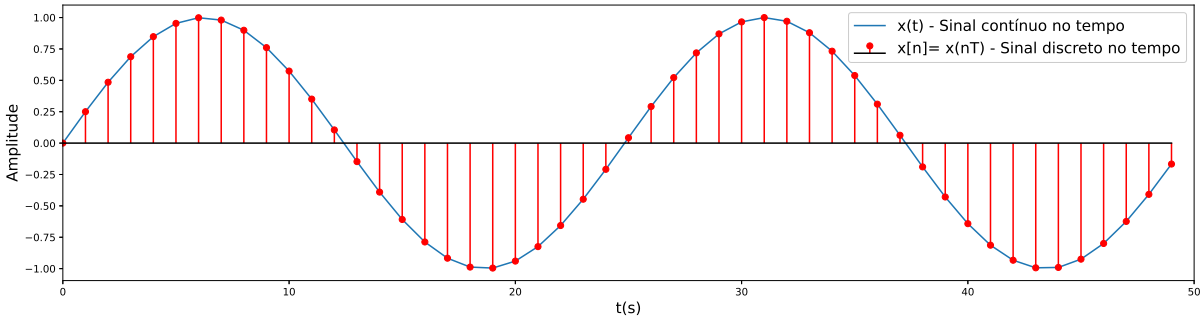
Fonte: Autoria própria.

2.3.1 Sinais de Áudio Digital

Um sinal é uma descrição de como um parâmetro mensurável está relacionado a outro parâmetro mensurável, e um bom exemplo é o tipo de sinal mais comum na eletrônica analógica, uma tensão que varia com o tempo. Como ambos os parâmetros podem assumir uma faixa contínua de valores, o sinal é chamado de sinal contínuo no tempo. Em comparação, após processar este mesmo sinal através de um conversor analógico-digital (ADC), cada um dos dois parâmetros são forçados a serem discretizados, e o sinal passa a ser chamado de sinal discreto no tempo (SMITH, 2003, p 11).

Portanto, de forma resumida, o sinal analógico e contínuo no tempo $x(t)$ é discretizado tanto na amplitude quanto no tempo. Em que a quantização se refere à discretização das amplitudes, e a amostragem se refere à discretização no tempo. O sinal resultante é uma série de valores de amplitude quantizados $x[n]$ (LERCH, 2012, p. 9). Na Figura 3, há um exemplo de sinal analógico, contínuo no tempo, e um sinal digital, já amostrado e discreto no tempo.

Figura 3 – Exemplo de sinais contínuo e discreto no tempo



Fonte: Autoria própria.

2.3.2 *Short-Time Fourier Transform (STFT)*

A transformada de Fourier gera informações de frequência que são calculadas sobre todo o domínio do tempo de um sinal em consideração. No entanto, as informações sobre quando essas frequências acontecem ao longo do tempo ficam ocultas na transformação. Para recuperar as informações de tempo ocultas, Dennis Gabor introduziu no ano de 1946 a transformada de Fourier de curta duração. Em vez de considerar todo o sinal, a ideia é considerar apenas uma pequena parte do sinal. Para isso, fixa-se a chamada função janela, que é uma função diferente de zero por um curto período de tempo. O sinal original é então multiplicado pela função de janela para produzir um sinal da janela em questão. Para se obter informações de frequência em diferentes instâncias de tempo, deve se deslocar a função de janela ao longo do tempo, como uma janela móvel, e calcular transformada de Fourier para cada um dos sinais de janela resultantes (MÜLLER, 2015, p 53). A STFT discreta de um sinal é calculada pela equação (1) na sequência.

$$STFT(k,m) = \sum_{n=0}^{N-1} w(n)x(n+mH)e^{(-i\frac{2\pi}{N}kn)} \tag{1}$$

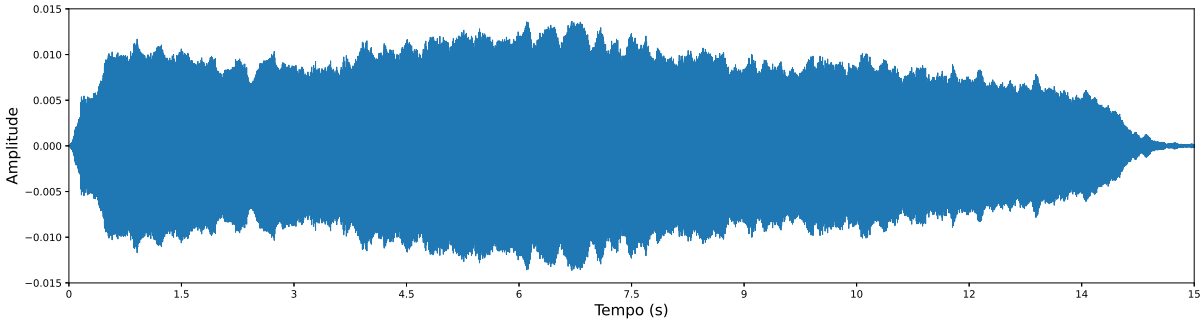
O número complexo $STFT(m,k)$ denota o k-ésimo coeficiente de Fourier para o m-ésimo período de tempo. Para cada frame de tempo fixo m, obtém-se um vetor espectral de tamanho $k+1$ dado pelos coeficientes $STFT(m,k)$ para $k \in [0:k]$. O número $k = N/2$ (assumindo que

N é par) é o índice de frequência correspondente à frequência de Nyquist, e o parâmetro de comprimento N determina a duração das janelas consideradas. $H = [0 : N]$, é definido como o tamanho do salto, ele é especificado em amostras e determina o tamanho do passo no qual a janela móvel deve ser deslocada através do sinal. O cálculo de cada vetor espectral equivale a uma *discrete Fourier transform* (DFT) de tamanho N, que pode ser feita eficientemente usando a *fast Fourier transform (FFT)* (MÜLLER, 2015, p 53).

2.3.3 Domínios de Representação dos Sinais

Os dois tipos de representações de sinais de áudio digital mais utilizados, são a representação no domínio do tempo e a representação no domínio da frequência. A variável independente de um sinal é determinada pelo parâmetro em que seu comportamento, independe do comportamento do outro parâmetro. Um sinal que usa o tempo como variável independente na sua representação, é dito estar no domínio do tempo. Para outro sinal comum em que se use a frequência como variável independente, é dito estar no domínio da frequência (SMITH, 2003, p 12). Na Figura 4 é possível observar um sinal no domínio de representação do tempo, enquanto na Figura 5 está uma representação de um sinal no domínio da frequência.

Figura 4 – Representação global de um sinal de áudio no domínio do tempo

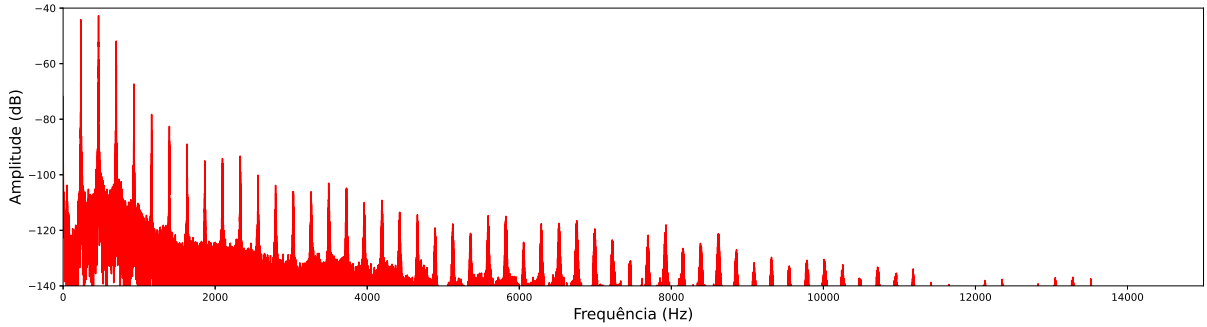


Fonte: Autoria própria.

Na Figura 6 está uma representação no domínio da frequência de um sinal segmentado por diversas janelas, em que cada linha representa o espectro de magnitudes extraído pela STFT de apenas uma janela móvel de análise. Cada uma das linhas conectadas formam uma abstração de bandas de energia e cada coeficiente de Fourier está representado pelos seus vértices. A figura resultante final é formada pela STFT de cada uma das janelas segmentadas do sinal global.

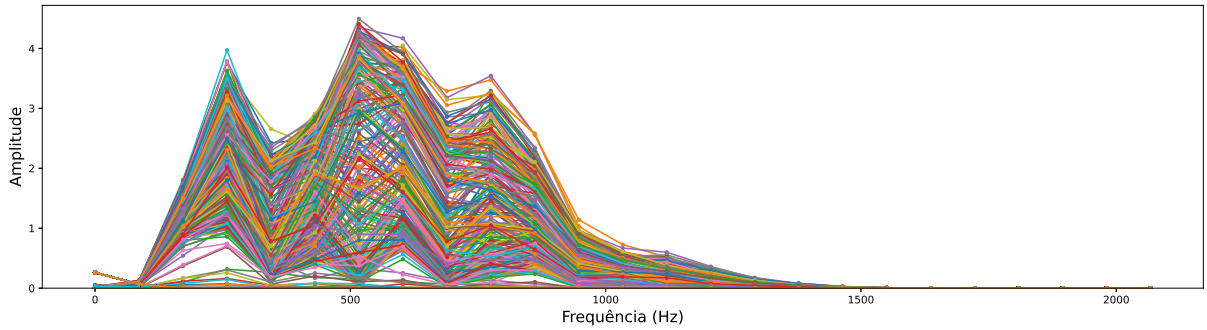
É no domínio do tempo em que as ondas sonoras são reproduzidas e gravadas, a frequência é o outro domínio em que elas podem ser representadas e compreendidas. Na música

Figura 5 – Representação global de um sinal de áudio no domínio da frequência



Fonte: Autoria própria.

Figura 6 – Representação segmentada da variação das amplitudes dos coeficientes de Fourier ao longo do tempo



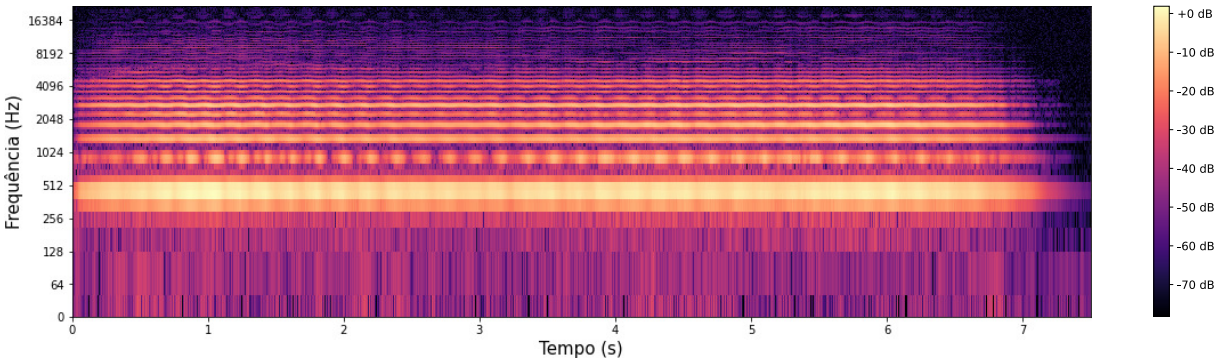
Fonte: Autoria própria.

ocidental, a altura de uma nota em uma partitura, representa sua frequência fundamental. Como o tempo sozinho ou a frequência sozinha não são suficientes para representar uma música, precisamos pensar em termos de representações conjuntas de tempo e frequência. As partituras musicais ocidentais são, na verdade, representações de tempo-frequência (TFRs) com uma codificação específica. Existe apenas uma transformada de Fourier de um determinado sinal, no entanto, existe um grande número de representações de tempo-frequência (TFRs), o mais popular é o espectrograma, que é uma representação de energia, definido como a transformada de Fourier de quadros sucessivos de um sinal (KLAPURI; DAVY, 2006, p. 23). Na Figura 7 é exibido um exemplo de representação de tempo-frequência, um espectrograma.

2.4 CARACTERÍSTICAS DE SINAIS DE ÁUDIO

A extração de características de sinais de áudio digital é uma tarefa fundamental e conhecida por ser a base do campo de estudo chamado *Music Information Retrieval (MIR)*. Esse campo tem seu foco voltado exclusivamente para a extração de informações musicais de sinais, que estão codificadas em arquivos digitais, de formatos como *wav, mp3, aiff, flac*, entre

Figura 7 – Representação tempo-frequência de um sinal de áudio



Fonte: Autoria própria.

outros. Essas informações são chamadas de características, recursos ou *features*, que é o termo amplamente usado em inglês.

O termo "característica", é usado para referir-se a uma descrição numérica ou nominal do sinal musical em consideração, normalmente sendo o resultado de um processo de extração de informações disponíveis nos sinais de áudio. O objetivo das ferramentas de extração é transformar os dados brutos em representações mais descritivas, idealmente descrevendo os aspectos musicais como eles são fisicamente ou na forma como são percebidos por humanos. Alguns exemplos de aspectos perceptuais, são os relacionados à instrumentação, estrutura rítmica, melodia ou harmonia (KNEES; SCHEDL, 2016, p. 33).

2.4.1 Categorização das Características

As categorizações a seguir foram descritas por Knees e Schedl (2016) se baseando em quatro tipos de possíveis análises sobre as características de sinais de áudio digital, são elas o nível de abstração, o escopo temporal, aspecto musical e domínio de representação.

2.4.1.1 Nível de Abstração

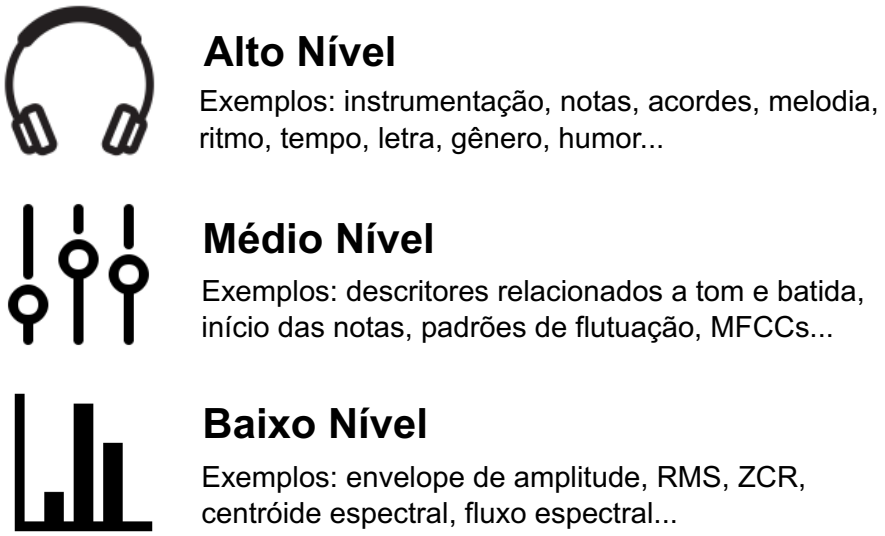
O nível de abstração de uma característica é uma das formas de categorização de sinais de áudio, e é dividida em uma escala de três níveis. Os níveis estão representados abaixo, e a Figura 8 contém alguns exemplos dos mesmos.

1. Alto Nível
   - São as que descrevem a música em termos de como nós humanos a percebemos, esses aspectos incluem instrumentação, ritmo, melodia e letras de músicas.

- São compreendidas por usuários finais de música, ouvintes comuns.

2. Médio Nível
   - São as que capturam aspectos que já são musicalmente mais significativos, como propriedades relacionadas a notas ou batidas por minuto. Frequentemente também são resultado de combinações de características de baixo nível, ou a aplicação de algum modelo do ramo psicoacústico (ramo da ciência que estuda como os seres humanos percebem os sons, e as respostas psicológicas associadas ao som).
   - São compreendidas por especialistas em música e instrumentação.
3. Baixo Nível
   - São normalmente calculados diretamente da forma de onda e dos dados brutos de sinais, são como resumos estatísticos, físicos e matemáticos de formas de ondas.
   - São compreendidas por profissionais que trabalham com processamento de sinais.

Figura 8 – Níveis de abstração das características de áudio



Fonte: Autoria própria.

2.4.1.2 Escopo Temporal

Essa estratégia de categorização se aplica a qualquer tipo de sinal de áudio sendo ele musical ou não. O foco de análise é voltado para o tamanho da janela de extração das características. Podemos diferenciar as características como:

1. Instantâneas
   - São calculadas para um determinado momento, ou janela rolante de análise.

- Seu intervalo de tempo está entre $\approx 10\ ms$ e no máximo $\approx 100\ ms$.

2. Nível de Segmento
   - São calculados em janelas de comprimento fixo ou usando uma definição semanticamente mais significativa, como uma frase musical, uma melodia ou refrão.
   - Seu intevalo de tempo está na escala de segundos.
3. Globais
   - Descrevem todo o item musical de seu interesse, como uma música, um movimento ou um trecho de áudio.
   - Seu intervalo de tempo é o mesmo que totalidade do sinal, a característica se extende por todo o sinal, por toda uma música por exemplo.

2.4.1.3 Aspecto Musical

Essas características de áudio estão mais relacionadas a sinais músicais do que sinais de áudio em geral. Elas são categorizadas em termos dos aspectos musicais e perceptivos que as descrevem. Alguns exemplos são: instrumentação, timbre, batida, ritmo, *pitch*, melodia, acordes, harmonia, entre outros.

2.4.1.4 Domínio de Representação

Esse tipo de categorização, foi abordado na subseção 2.3.3, e está diretamente ligado ao domínio em que as características são computadas, analisadas e representadas. A categorização distingue as características pela: representação no domínio do tempo, representação no domínio da frequência ou representação tempo-frequência.

2.4.2 Características Instantâneas Clássicas

Os termos característica instantânea e características de curto prazo são normalmente usados para medidas que geram um valor por uma janela de análise curta de amostras de áudio. Uma característica instantânea não é necessariamente musicalmente ou perceptivelmente significativa por si só, e é freqüentemente referida como uma característica de baixo nível (LERCH, 2012, p 31).

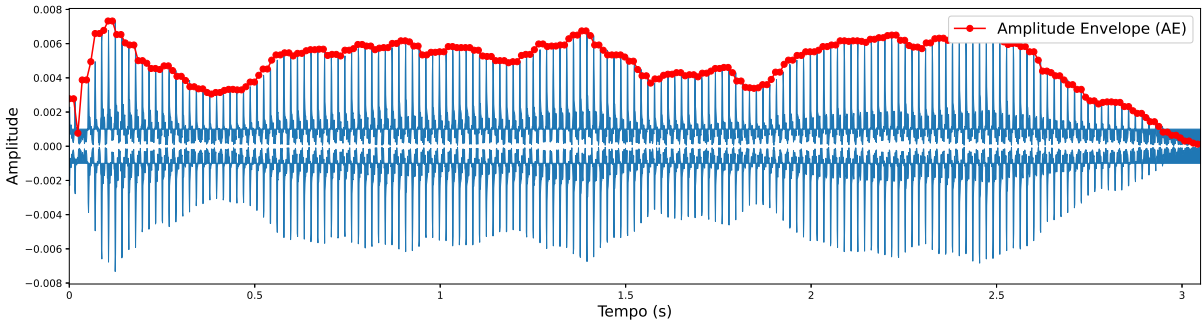
2.4.2.1 Envelope de Amplitude (AE)

Essa característica é o envelope de amplitude de uma sinal em questão, é definida como o valor máximo de amplitude de todas as amostras em uma determinada janela de análise t. Descreve as formas de ondas dos sinais de áudio. É computada na representação de sinais no domínio do tempo e considerada de baixo nível. Essa característica é sensitiva a valores *outliers*. Sua definição formal segue abaixo na Equação (2) e está representada graficamente na Figura 9.

$$AE_t = max_{k=t.K}^{(t+1)K-1} s(k) \tag{2}$$

AE_t descreve o envelope de amplitude na janela de análise t. A função $s(k)$ denota a amplitude da k-ésima amostra, enquanto o K se refere o tamanho da janela de análise, ou em outras palavras o número de amostras dentro de uma janela. Por fim, a função max calcula qual é a amostra com maior amplitude dentro de cada uma janelas de análise. Essa equação é utilizada para cada nova janela t segmentada de um sinal global.

Figura 9 – Sinal no domínio do tempo e AE de uma nota de uma tuba



Fonte: Autoria própria.

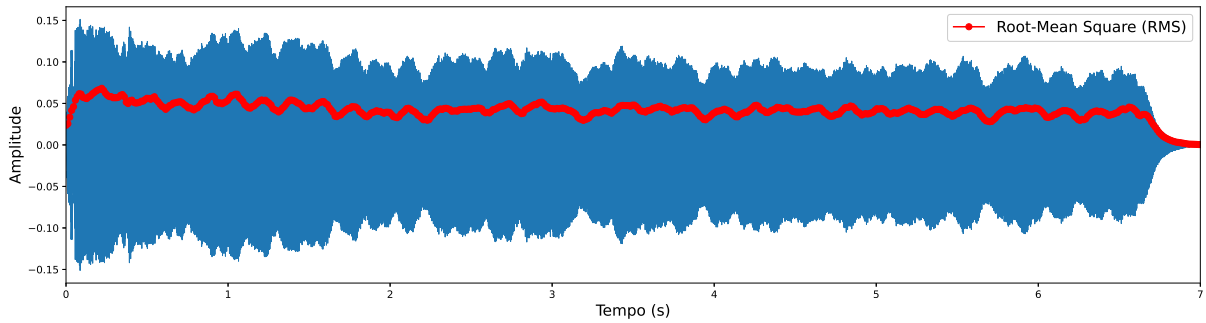
2.4.2.2 *Root-Mean-Square (RMS)*

Se trata da energia quadrática média em uma janela de análise t, que normalmente é denotada como energia RMS ou nível RMS. É descrita como uma estimativa de volume ou também intensidade sonora percebida. É outra característica extraída da representação do sinal no domínio do tempo e categorizada como de baixo nível. Essa característica é pouco sensitiva a valores *outliers*. Seu cálculo se da pela Equação (3) abaixo, e na sequência segue a representação gráfica do sinal de uma nota de um cello na Figura 10.

$$RMS_t = \sqrt{\frac{1}{K} \sum_{k=t.K}^{(t+1)K-1} s(k)^2} \tag{3}$$

RMS_t é a raiz do valor quadrático médio na janela de análise t. A função $s(k)$ denota a energia da k-ésima amostra, enquanto o K denota o tamanho da janela de análise. Ao multiplicar a somatória com $1/K$, é obtido a média da soma dos valores de energia. E quando aplicado a raiz quadrada, é obtida a raiz do valor quadrático médio. Essa equação é utilizada para cada nova janela t segmentada de um sinal global.

Figura 10 – Sinal no domínio do tempo e RMS de uma nota de um cello



Fonte: Autoria própria.

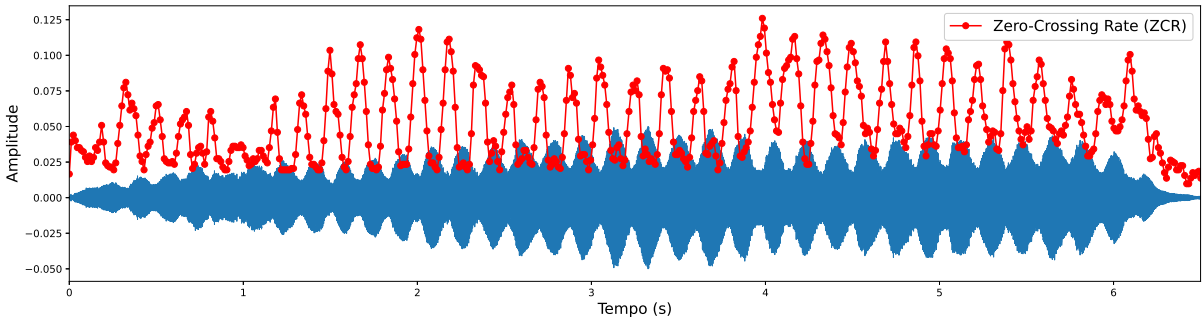
2.4.2.3 *Zero-Crossing Rate (ZCR)*

É a medida de quantas vezes os valores de amplitude mudam de sinal, ou seja cruzam o valor zero em uma janela de análise t. É utilizado para identificar o quão oscilatório é um sinal. É mais uma característica extraída da representação do sinal no domínio do tempo e categorizada como de baixo nível. É definido pela Equação (4) abaixo, e representada na Figura 11.

$$ZCR_t = \frac{1}{2} \sum_{k=t.K}^{(t+1)K-1} \mid sgn(s(k)) - sgn(s(k+1)) \mid \tag{4}$$

ZCR_t é a quantidade de vezes que o sinal cruzou o valor zero na janela de análise t. A função $sgn(s(k))$ é comparada com $sgn(s(k+1))$ da amostra seguinte, e a relação determina o sinal de saída de uma amplitude dependendo se $s(k) > 0 \rightarrow +1$, $s(k) < 0 \rightarrow +1$, ou $s(k) = 0 \rightarrow 0$. Se o valor absoluto resultante da operação for 0, os sinais se subtarem e quer dizer que o valor da amostra atual e da próxima estão no mesmo plano, senão os sinais se somam e o resultado será 2, que é um indicativo que o sinal atravessou o eixo. O valor final é dividido por $1/2$ para não computar como 2 passagens. O parâmetro K denota o tamanho da janela de análise e essa equação é utilizada para cada nova janela t segmentada de um sinal global.

Figura 11 – Sinal no domínio do tempo e ZCR de uma nota de um violino



Fonte: Autoria própria.

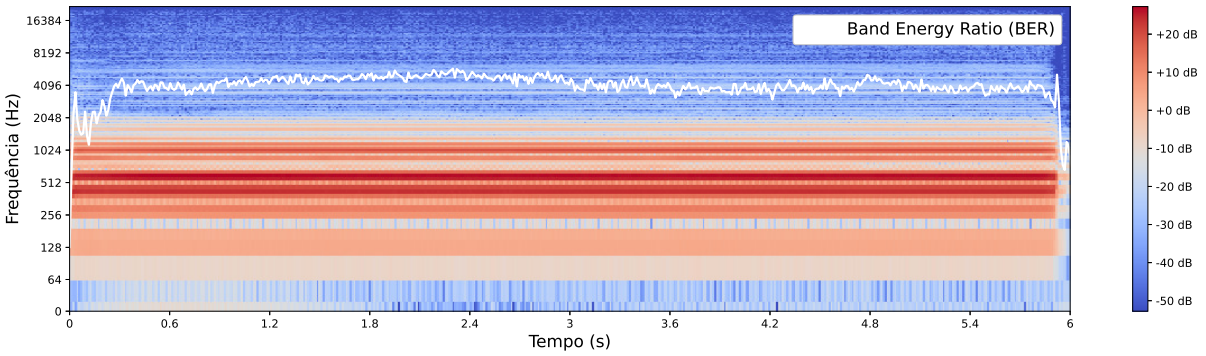
2.4.2.4 *Band Energy Ratio (BER)*

Representa a razão da energia entre bandas, ela relaciona a energia nas bandas de baixa frequência com a energia nas bandas de alta frequência. Utilizando essa característica é possível mensurar o quão dominantes as baixas frequências são para cada janela de análise t.A característica é considerada de baixo nível e extraída em uma representação de sinais no domínio da frequência, porém é mais fácil visualizá-la em uma representação tempo-frequência como na Fig. 12. A equação que a define é a (5) abaixo.

$$BER_t = \frac{\sum_{n=1}^{F-1} m_t(n)^2}{\sum_{n=F}^{N} m_t(n)^2} \tag{5}$$

BER_t é a razão entre duas bandas de frequências na janela de análise t.O parâmetro $m_t(n)$ representa a magnitude de um sinal em banda de frequências em uma janela t. N é o número de bandas de frequências. O parâmetro F da equação denota a o valor de frequência que determina a separação das bandas. Apesar de ser arbitrário, segundo Knees e Schedl (2016, p 48) o valor padrão mais utilizado de F para dividir as baixas e altas frequências é de 2000 Hz. Observando a equação como um todo, no numerador está a potência das magnitudes nas baixas frequências, e no denominador a potência das magnitudes nas altas frequências, o resultado dessa divisão retorna uma razão entre as bandas. Essa equação é utilizada para cada nova janela t segmentada de um sinal global, e as magnitudes $m_t(n)$ são obtidas após a aplicação da STFT, também a cada nova janela.

Figura 12 – Sinal no domínio da frequência e BER de uma nota de um fagote



Fonte: Autoria própria.

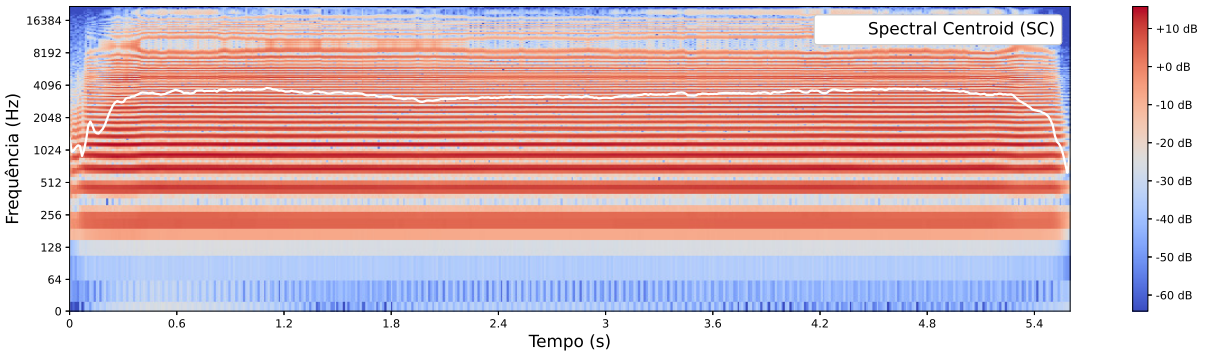
2.4.2.5 *Spectral Centroid (SC)*

O *spectral centroid* ou centróide espectral representa o centro de gravidade de um espectrograma, ou seja, é a média ponderada das magnitudes das frequência para cada janela de análise t. Essa característica tem relação com o timbre da música e no ramo da psicoacústica é utilizada como uma forma de mensurar o "brilho" de um timbre musical. É extraída na representação do sinal no domínio da frequência e também facilmente visualizada na representação tempo-frequência na Fig. 13. A equação que determina o cálculo da mesma segue em (6).

$$SC_t = \frac{\sum_{n=1}^{N} m_t(n).n}{\sum_{n=1}^{N} m_t(n)} \tag{6}$$

O parâmetro n é uma banda de frequência, $m_t(n)$ representa a magnitude de um sinal em banda de de frequências para uma janela t. A equação no geral é a média da soma das magnitudes em uma janela t, em que o numerador é multiplicado por um peso, que a própria banda de frequências. Essa equação é utilizada para cada nova janela t segmentada de um sinal global, e as magnitudes $m_t(n)$ são obtidas após a aplicação da STFT, também a cada nova janela.

Figura 13 – Sinal no domínio da frequência e SC de uma nota de um trompete



Fonte: Autoria própria.

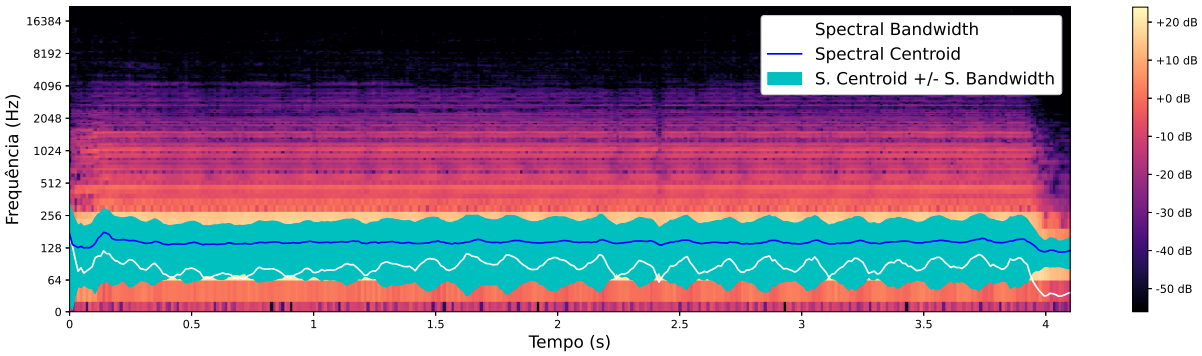
2.4.2.6 *Spectral Bandwidth (BW)*

A *Spectral Bandwidth*, ou largura de banda espectral, e também conhecida como *spectral spread*, indica o alcance espectral da energia do sinal no domínio da frequência, no entorno do centróide espectral (SC) para cada janela de análise t. Seu significado é interpretado como a variância da frequência média do sinal, como uma forma de mensurar quanto espaço o sinal ocupa no domínio da frequência, e também é utilizado no ramo da psicoacústica como uma descrição de timbre percebido. É extraída na representação do sinal no domínio da frequência e também visualizada na representação tempo-frequência na Fig. 14. A equação que define o cálculo da característica segue na Equação (7).

$$BW_t = \frac{\sum_{n=1}^{N} |n - SC_t|.m_t(n)}{\sum_{n=1}^{N} m_t(n)} \tag{7}$$

BW_t é a largura de banda em uma janela t. O parâmetro $m_t(n)$ representa a magnitude de um sinal em uma banda de frequências para uma janela t. O valor absoluto $|n - SC_t|$ representa a distância entre uma banda de frequências e o centróide espectral (SC). Essa equação é utilizada para cada nova janela t segmentada de um sinal global, e as magnitudes $m_t(n)$ são obtidas após a aplicação da STFT, também a cada nova janela.

Figura 14 – Sinal no domínio da frequência e BW de uma nota de um contrabaixo



Fonte: Autoria própria.

2.5 CLASSIFICAÇÃO DE SÉRIES TEMPORAIS

A tarefa de classificação busca atribuir rótulos a cada série de um conjunto de dados. A principal distinção em relação à tarefa de agrupamento (conhecida no inglês como *clustering*), é que as classes são conhecidas antecipadamente e o algoritmo é treinado em um conjunto de

dados de exemplo. O objetivo é primeiro aprender quais são as características distintivas que diferem as classes umas das outras. Então, quando um conjunto de dados não rotulado é inserido no algoritmo, o mesmo pode inferir automaticamente a qual classe cada série mais se aproxima pertencer (ESLING; AGON, 2012).

A classificação de séries temporais é uma forma de aprendizado de máquina em que as características do vetor de entrada são valorizadas e ordenadas no seu formato real. Esse cenário adiciona uma camada de complexidade ao problema, mas características importantes dos dados não são perdidas ou abstraídas, como podem ser nos algoritmos tradicionais não adaptados para séries. Apesar de já existirem abordagens de classificação de séries temporais univariadas, as tarefas como reconhecimento de atividade humana, diagnóstico baseado em eletrocardiograma (ECG), eletroencefalograma (EEG), magnetoencefalografia (MEG) e problemas de monitoramento de sistemas, são geralmente tarefas inerentemente compostas de séries multivariadas, e não univariadas, o que reforça a importância dessa abordagem no mundo real (BAGNALL *et al.*, 2017).

2.5.1 Categorização dos Algoritmos

Os algoritmos para classificação multivariada de séries temporais podem ser categorizados de maneira semelhante aos algoritmos para o formato univariado. Vários autores já investigaram formas de classificar séries temporais analisando suas características, dentre essas, a principal diferença está nas dependencias dos algoritmos, eles podem ser baseados em distância, intervalos, frequência, *shapelets*, *kernel*, características, aprendizado profundo, entre outros.

Essa categorização está presente de forma clara nos trabalhos de Maharaj, D'Urso e Caiado (2019), Bagnall *et al.* (2017) e Ruiz *et al.* (2021), que abordaram temas de séries temporais de maneira geral, e também está presente em constante atualização na documentação da biblioteca sktime de Löning *et al.* (2019).

2.5.2 *K-Nearest Neighbour (k-NN)*

O algoritmo *K-Nearest Neighbour (k-NN)* é o método mais básico baseado em instância. Este algoritmo assume que todas as instâncias correspondem a pontos no espaço n-dimensional. Os vizinhos mais próximos de uma instância são definidos em termos da distância Euclidiana padrão. Mais precisamente, deixa uma instância arbitrária x ser descrita pelo vetor de características

(MITCHELL, 1997, p. 231). Partindo de

$$\langle a_1(x), a_2(x), \ldots a_n(x) \rangle, \tag{8}$$

em que $a_r(x)$ denota o valor do r-ésimo atributo da instância x. Então, a distância entre duas instâncias x_i e x_j é definida como

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \tag{9}$$

De forma resumida, o algoritmo primeiro armazena os vetores de características de todos as amostras de treinamento, em seguida, para classificar uma nova instância, encontra um conjunto de k amostras de treinamento mais próximas no espaço de características, e atribui a nova amostra à classe que tem o maior número de amostras naquele conjunto. Tradicionalmente, a medida de distância Euclideana é usada para determinar a similaridade, e o número de vizinhos é determinado empiricamente (KLAPURI; DAVY, 2006, p. 185).

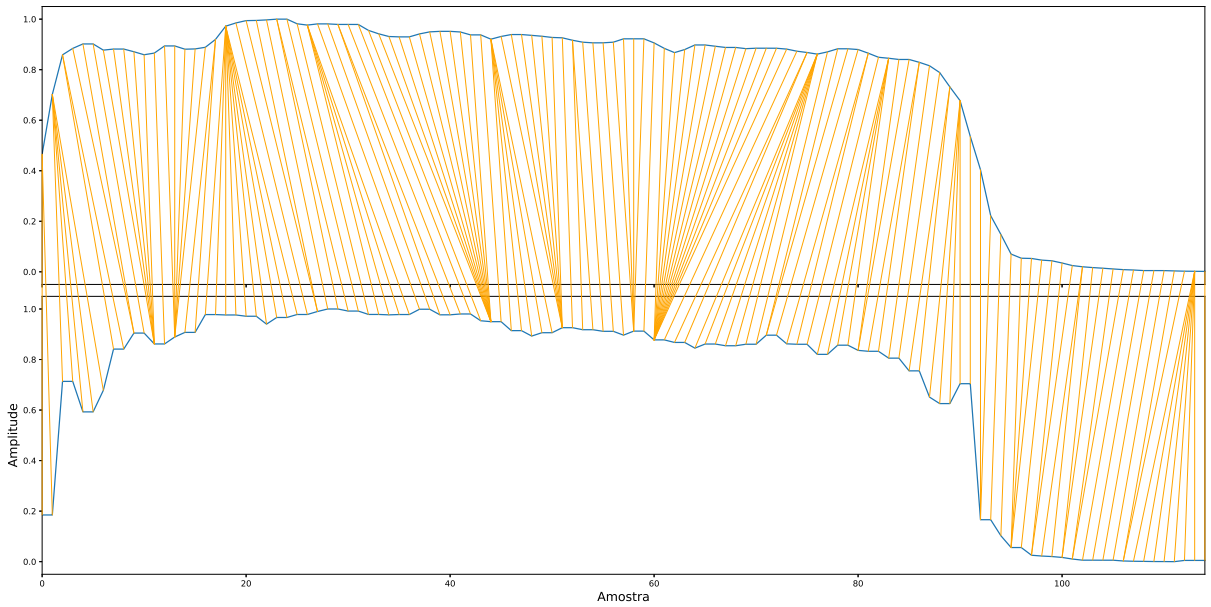
Dentre as abordagens de classificação de séries temporais baseadas em distância, mencionadas na subseção 2.5.1, a grande maioria são baseadas principalmente na medida padrão de distância *Dynamic Time Warping (DTW)*, e segundo Xi *et al.* (2006) e Ruiz *et al.* (2021), o algoritmo **1NN-DTW** tem sido uma referência popular na pesquisa de classificação de séries temporais há vários anos, e ao mesmo tempo continua sendo um padrão de referência difícil de superar nesse tipo de tarefa.

2.5.2.1 *Dynamic Time Warping (DTW)*

A medida de distância *Dynamic Time Warping (DTW)* é uma técnica bem conhecida na comunidade de reconhecimento de fala, e sua ideia base foi utilizada inicialmente por Sakoe e Chiba (1978). A função de distância DTW, assim como a maioria das funções de distância focadas em medir similaridade, utiliza das distâncias entre um elemento de uma série temporal e um elemento de outra. A especialidade dessa abordagem é que ela pode compensar uma diferença de tempo localmente flutuante entre séries temporais. Em outras palavras, a distância DTW para séries temporais "distorce" o eixo temporal, alongando ou comprimindo duas séries localmente, para no fim conseguir o melhor alinhamento possível entre suas formas. Na Figura 15 é exibido uma demonstração de duas séries temporais resultantes da característica *amplitude envelope* de duas tubas, em que a técnica DTW encontra os pontos que mais aproximam suas formas.

Na sequência está o equacionamento do cálculo de uma distância DTW para séries temporais segundo a demonstração de Maharaj, D'Urso e Caiado (2019, p 53). Em que partindo

Figura 15 – Exemplo de alinhamento com *Dynamic Time Warping (DTW)*



Fonte: Autoria própria.

de uma série temporal referência de "consulta" (ou teste) x_i, e uma série temporal de referência x'_i, com comprimentos respectivamente T,$T'(T \leqq T')$, e assumindo que $t = 1,...,T$ e $t' = 1,...,T'$ são os índices de tempo dos elementos em x_i e x'_i.

A distância total entre x_i e x'_i é então calculada utilizando do chamado *warping path*, ou *warping curve*, que garante que cada ponto de dados em x_i seja comparado ao ponto de dados mais próximo em x_i. O *warping path* é definido do seguinte modo:

$$\Phi_l = (\varphi_l, \psi_l), l = 1,...,L \tag{10}$$

Sob as seguintes restrições.

- condição de contorno: $\Phi_1 = (1,1), \Phi_L = (T,T')$;
- condição de monotonicidade: $\varphi_1 \leq ... \leq \varphi_l \leq ... \leq \varphi_L$ e $\psi_1 \leq ... \leq \psi_l \leq ... \leq \psi_L$;

O efeito de aplicar o *warping path* às duas séries temporais multivariadas é realinhar os índices de tempo de x_i e x'_i por meio das funções φ e ψ. A dissimilaridade total entre as duas séries temporais multivariadas distorcidas é:

$$\sum_{l=1}^{L} d(x_{i,\varphi_l}, x_{i',\psi_l}) m_{l,\Phi}, \tag{11}$$

em que $m_{l,\Phi}$ é um coeficiente de ponderação local e $d(.,.)$ é geralmente a distância euclidiana para séries temporais multivariadas:

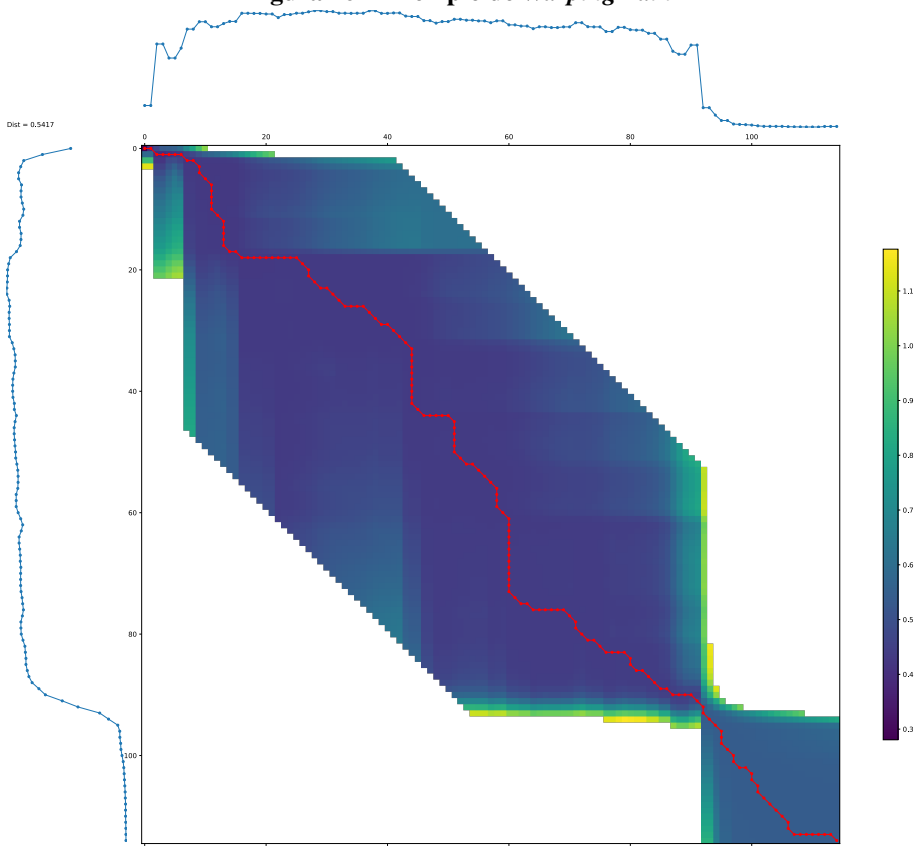
$$d(i,i') = (\|x_{it} - x_{i't'}\|)^{\frac{1}{2}}. \tag{12}$$

Como existem vários *warping paths*, a distância DTW é aquela que corresponde à *warping path* ótima, $\hat{\Phi}_l = (\hat{\varphi}_l, \hat{\psi}_l), l = 1,...,L$ que minimiza a dissimilaridade total entre x_i e x'_i:

$$d_{DTW}(x_i, x_{i'}) = \sum_{l=1}^{L} d(x_{i,\hat{\varphi}_l}, x_{i',\hat{\psi}_l}) ml,\hat{\Phi} \tag{13}$$

.

Figura 16 – Exemplo de *Warping Path*



Fonte: Autoria própria.

Portanto, a descoberta do *warping path* ótimo consiste nos seguintes passos, em que primeiro é calculada uma matriz TxT' custo local (ou distância local) que contém as distâncias entre cada par de pontos. Depois o algoritmo DTW encontra o caminho que minimiza o alinhamento entre x_i e $x_{i'}$, começando em $d(1, 1)$ e terminando em $d(T, T')$ e agregando o custo, que é a distância total. A cada passo o algoritmo encontra a direção em que a distância aumenta menos, sob as restrições dadas (MAHARAJ; D'URSO; CAIADO, 2019, p 54). Essa abordagem pode causar um problema de escalabilidade porque sua complexidade requer computação quadrática $O(n^2)$.

2.5.3 *Support Vector Machine (SVM)*

No início dos anos 2000, o modelo de classe *Support Vector Machine (SVM)* era a abordagem mais popular para o aprendizado supervisionado "pronto para uso", para quando você não tinha nenhum conhecimento prévio especializado sobre um domínio. Essa posição agora foi assumida por redes de aprendizagem profunda e florestas aleatórias. O ponto chave das SVMs é que algumas amostras são mais importantes do que outras e que identificá-las pode levar a uma melhor generalização (RUSSELL; NORVIG, 2020, p. 692). Três propriedades interessantes das SVMs se destacam:

- As SVMs constroem um separador de margem máxima, um limite de decisão com a maior distância possível aos pontos das amostras, generalizamdo bem o modelo.
- As SVMs criam o hiperplano de separação linear, mas eles têm a capacidade de incorporar os dados em um espaço de dimensão superior, usando o chamado truque do kernel. Frequentemente, os dados que não são linearmente separáveis no espaço de entrada original, são facilmente separáveis no espaço de dimensão superior.
- As SVMs não são paramétricos, o hiperplano de separação é definido por um conjunto de pontos de amostras, não por uma coleção de valores de parâmetro. Enquanto os modelos *K-Nearest Neighbour (k-NN)* precisam reter todos as amostras, um modelo SVM mantém apenas as amostras que estão mais próximas do plano de separação. Portanto, os SVMs combinam as vantagens dos modelos não paramétricos e paramétricos: eles têm a flexibilidade de representar funções complexas, mas são resistentes ao sobreajuste.

Segundo Russell e Norvig (2020, p. 693), tradicionalmente as SVMs usam a convenção de que os rótulos de classe são +1 e −1, em vez de +1 e 0. Além disso, enquanto colocamos anteriormente a interceptação no vetor de peso $\mathbf{w}$ (e um valor *dummy* 1 correspondente em $x_{j,0}$), as SVMs não fazem isso, eles mantêm a interceptação como um parâmetro b separado. Com isso em mente, o separador é definido como o conjunto de pontos $\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = 0$. Existe uma representação alternativa chamada de representação dual, na qual a solução ótima é encontrada resolvendo:

$$argmax_{\alpha} \sum_{j} \alpha_j - \frac{1}{2} \sum_{j,k} \alpha_j \alpha_k y_j y_k (\mathbf{x}_j \cdot \mathbf{x}_k) \tag{14}$$

sujeito às restrições $\alpha_j \geq 0$ e $\sum_j \alpha_j y_j = 0$. Este é um problema de otimização de

programação quadrática, para o qual existem bons pacotes de *software*. Uma vez que se encontra o vetor α, pode-se voltar a $\mathbf{w}$ com a equação $\mathbf{w} = \sum_j \alpha_j y_j \mathbf{x}_j$, ou podemos ficar na representação dual. A Equação (14) tem três propriedades importantes. Primeiro, a expressão é convexa, ela tem um único máximo global que pode ser encontrado com eficiência. Em segundo lugar, os dados entram na expressão apenas na forma de produtos escalares de pares de pontos, que também é verdadeira para a equação do próprio separador. Uma vez que os α_j foram calculados, a equação é escrita como

$$h(\mathbf{x}) = sign\left(\sum_j \alpha_j y_j (\mathbf{x} \cdot \mathbf{x_j}) - b \right). \tag{15}$$

A propriedade importante final é que os pesos α_j associados a cada ponto de dados são $zero$, exceto para os vetores de suporte, que são os pontos mais próximos do separador. Eles são chamados de vetores de "suporte" porque "sustentam" o plano de separação. Como geralmente há menos vetores de suporte do que amostras, os SVMs ganham algumas das vantagens dos modelos paramétricos (RUSSELL; NORVIG, 2020, p.694).

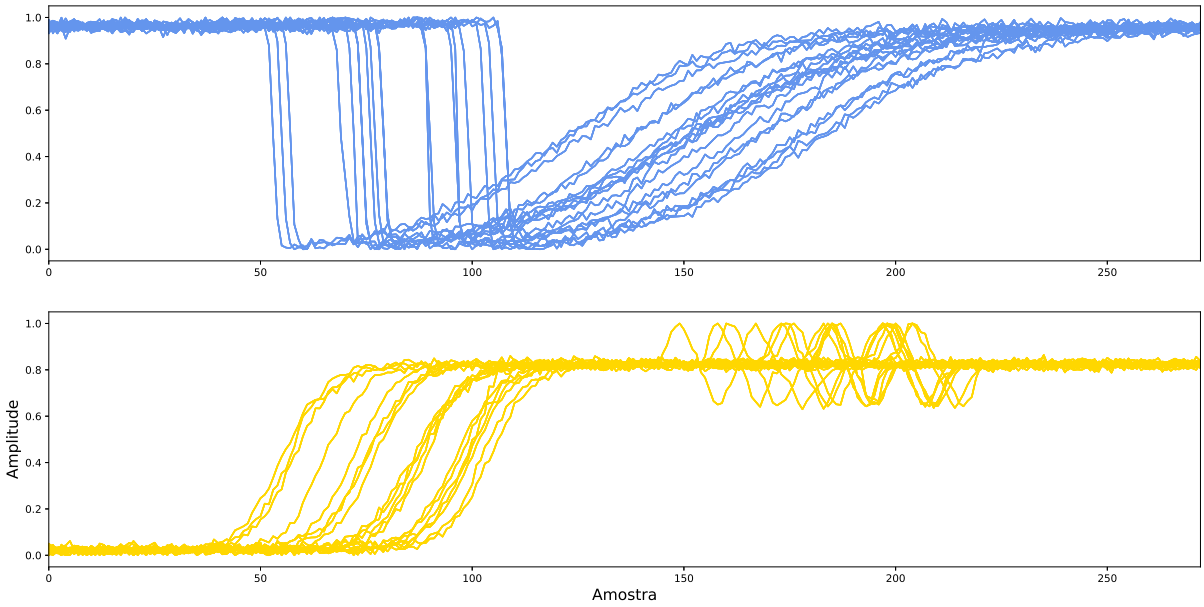
Dentre as abordagens de classificação de séries temporais baseadas em *kernel*, mencionadas em 2.5.1. Uma delas é a implementação do SVM em conjunto com o *Global Alignment Kernel (GAK)*. Esse algoritmo é possível graças ao *kernel k-means* introduzido por Dhillon, Guan e Kulis (2004), em que neste caso, diferente do *k-means* tradicional, os centros de agrupamento (*clusters*) nunca são computados explicitamente, no entanto, ainda é possível relatar as atribuições de agrupamentos de séries temporais, que são o único tipo de informação disponível após o agrupamento.

2.5.3.1 *Global Alignment Kernel (GAK)*

Cuturi *et al.* (2007) introduziram uma família de *kernels* para se lidar com séries temporais, chamada de *Global Alignment Kernels (GAK)*, dentro da estrutura de métodos de *kernel* que inclui algoritmos populares como o *SVM*. Esses *kernels* estão relacionados a família de distâncias *DTW* considerando o mesmo conjunto de operações elementares para mapear uma série em outra. Porém as similaridades DTW não são positivas semidefinidas (BOECKING *et al.*, 2014). Nos métodos do *kernel*, as similaridades grandes e pequenas são importantes, pois todas contribuem para a matriz de Gram (ou matriz kernel), em que se e somente se ela for simétrica e semidefinida positiva, tem-se a garantia de um kernel válido. Os GAK's, que são definidos

positivos, parecem fazer um trabalho melhor de quantificar todas as similaridades de forma coerente, porque consideram todos os alinhamentos possíveis. Na Figura 17 é exibido o resultado de um agrupamento de séries de duas classes diferentes, que são vetores de suporte alinhados utilizando GAK.

Figura 17 – Exemplo de alinhamento com *Global Alignment Kernel (GAK)*



Fonte: Autoria própria.

Na sequência é exposto o equacionamento de GAK como foi definido por Cuturi (2011). Escrevendo $A(n,m)$ como o conjunto de todos os alinhamentos entre duas séries temporais de comprimento n e m, e seguindo o princípio subjacente às pontuações DTW, para definir o custo de um alinhamento, uma função de pontuação $D(\pi)$ é usada:

$$D_{x,y} = \sum_{l=1}^{|\pi|} \varphi(x_{\pi_1(i)}, y_{\pi_2(i)}), \tag{16}$$

em que φ é um *kernel* arbitrário definido condicionalmente positivo, como a distância euclidiana ao quadrado $\varphi(x,y) = -\|x-y\|^2$. Considerando todos os valores de pontuação exponenciado por todas as distâncias de alinhamento, o GAK é definido como

$$k_{GA}(x,y) = \sum_{\pi \in A(m,n)} e^{-D_{x,y}(\pi)}, \tag{17}$$

que pode ser reescrita utilizando a função de similaridade local K induzida a partir da divergência φ como $K = e^{-\varphi}$, então tem-se

$$k_{GA}(x,y) = \sum_{\pi \in A(m,n)}^{|\pi|} K(x_{\pi_1(i)}, y_{\pi_2(i)}). \tag{18}$$

O k_{GA} incorpora todo o espectro de custos $\{D_{x,y}(\pi), \pi \in A(x,y)\}$ e fornece portanto uma estatística mais rica do que o mínimo desse conjunto, que é a única quantidade considerada pela distância DTW. O esforço computacional necessário para calcular escalas de k_{GA} é de $O(n^2(m + \tau))$, semelhante a da distância DTW com o adicional do m, que é o número de dimensões (características) de entrada e o τ é o número total de iterações (CUTURI, 2011).

2.6 CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentado os conceitos teóricos necessários para o entendimento deste trabalho, iniciando por algumas definições (mesmo que as vezes subjetivas) de instrumentos musicais e suas categorizações. Foram introduzidos alguns conceitos básicos de sinais e do processo de captação de um som até sua conversão para um sinal de áudio digital, sinal que pode ser analisado em alguns domínios de representação. Procurou-se apresentar brevemente esses domínios onde existem várias características que podem ser extraídas com o auxílio de técnicas como a da STFT. Essas características também são distingudas por várias categorizações e sua estrutura final é uma série temporal. Por fim, também mostrou-se que dentre os vários métodos de classificação multivariada de séries temporais, foram definidas duas abordagens distintas em que uma é baseada em distância e outra em *kernel*. A primeira utiliza do KNN em conjunto com a medida de distância DTW, a segunda e utiliza do SVM em conjunto com a medida de distância GAK, e ambas foram a base para a construção dos modelos de aprendizado de máquina supervisionados utilizados para a tarefa de classificação proposta neste trabalho.

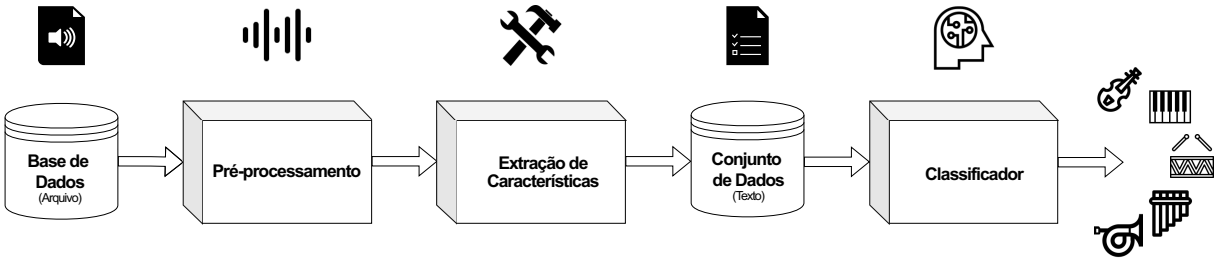
3 MATERIAIS E MÉTODO

3.1 CONSIDERAÇÕES INICIAIS

Neste capítulo foram pontuados os materiais utilizados e também é descrita a metodologia de abordagem para o desenvolvimento deste trabalho. Iniciando pela Seção 3.2 em que são exibidas as ferramentas e tenologias, assim como bilbiotecas e bases de dados a serem utilizadas. Na Seção 3.3 é apresentado o fluxograma de pré-processamento de dados, necessário antes da aplicação do modelo classificador. Na Seção 3.4 são discutidos os modelos classificadores a serem utilizados.

Na Figura 18 podemos observar o *pipeline* completo que proporciona uma dimensão geral e abstrata, do caminho do fluxo de dados desde o arquivo bruto até o processo de classificação por um modelo final. Os blocos desse fluxograma são ainda divididos em subetapas menores e explicados nas suas respectivas subseções.

Figura 18 – *Pipeline* de dados geral



Fonte: Autoria própria.

3.2 FERRAMENTAS E TECNOLOGIAS

Essa seção apresenta uma lista com as ferramentas físicas e tecnologias virtuais utilizadas no desenvolvimento deste trabalho.

- Notebook Dell Inspiron I14-3442, com Processador Intel Core i54210U 1.70GHz, NVIDIA GeForce GT420M, Memória RAM de 8GB e Windows 10 64-bit.
- Google Drive e Google Colaboratory
- Linguagem de Programação Python v3.7.10
- Biblioteca NumPy v1.20.0 (HARRIS *et al.*, 2020)
- Biblioteca Pandas v1.2.4 (MCKINNEY, 2010)
- Biblioteca Librosa v0.8.0 (MCFEE *et al.*, 2015)

- Biblioteca Sktime v0.6.0 (LÖNING *et al.*, 2019)
- Biblioteca Tslearn v0.5.2 (TAVENARD *et al.*, 2020)
- Biblioteca Scikit-learn v0.24.2 (PEDREGOSA *et al.*, 2011)
- Biblioteca Matplotlib v3.4.2 (HUNTER, 2007)
- Biblioteca Seaborn v0.11.1 (WASKOM, 2021)

3.2.1 Bibliotecas de Análise de Áudio e Processamento de Dados

Todas as etapas de pré-processamento sobre arquivos de áudio ou vetores numéricos na seção 3.3, foram implementadas com auxílio de algumas bibliotecas. A extração de características e operações com sinais de áudio foram realizadas com as bibliotecas *Librosa* e *Essentia*. As manipulações numéricas e de estrutura de dados com as bibliotecas *NumPy* e *Pandas*. As bibliotecas foram desenvolvidas na linguagem de programação *python*.

3.2.2 Bibliotecas de Séries Temporais e Aprendizado de Máquina

O tratamento de séries temporais e modelos de aprendizado de máquina na seção 3.4 foram utilizados e analisados com as bibliotecas *sktime* e *tslearn*, a biblioteca *scikit-learn* está implícita dentro das duas anteriores, ela também foi utilizada para gerar métricas de desempenho dos modelos. As bibliotecas foram desenvolvidas na linguagem de programação *python*.

3.2.3 Bibliotecas de Visualização e Análise de Dados

As visualizações geradas ao longo do trabalho e neste documento foram implementadas com as bibliotecas *Matplotlib* e *Seaborn*, ambas muito utilizadas na visualização e análise de dados. As bibliotecas foram desenvolvidas na linguagem de programação *python*.

3.2.4 Bases de Dados

Devido a grande variedade de instrumentos musicais distribuídos em diversas culturas e na disponibilidade de bases de dados, foi necessária a limitação do escopo deste trabalho quanto ao número de instrumentos. Os instrumentos comuns utilizados nas orquestras ocidentais serão os considerados pelo motivo de que as gravações desses instrumentos já foram amplamente

realizadas para a utilização em pesquisas ou produção musical.

Os dados brutos a serem utilizados são de duas bases de dados bem semelhantes com relação aos instrumentos presentes, a escolha dessas bases foi determinada também pela possibilidade de comparação justa dos resultados. As bases contém sinais de áudio digital de notas singulares gravadas em estúdio de vários instrumentos. Ambas as bases de dados são disponibilizadas na internet com licença para uso *creative commons 4.0*. Os arquivos de ambas estão em formato *wav*, armazenados com uma taxa de amostragem de 44100 kHz e resolução de 24 $bits$. Na sequência são referenciadas as bases utilizadas e na a tabela 2 listados os instrumentos e quantidades nelas contidos.

- IRCAM - TinySOL Database *(Creative Commons 4.0)* (EMANUELE *et al.*, 2020)
- The Alpine Project - A Free Orchestral Instrument Sample Library (2021 Edition) *(Creative Commons 4.0)* (TEAM, 2021)

Tabela 2 – Instrumentos que compõe as bases de dados

Instrumento	TinySOL	The Alpine Project	Total de Amostras
TROMPA (HORN)	134	42	176
TROMBONE	117	122	239
TRUMPETE	96	159	255
TUBA	108	28	136
CELLO	291	90	381
CONTRABAIXO	309	56	365
VIOLA	309	82	391
VIOLINO	284	111	395
FAGOTE (BASSOON)	126	28	154
CLARINETE	126	155	281
FLAUTA	118	145	263
OBOE	107	28	135
SAXOFONE	99	30	129
FLAUTIM (PICCOLO)	*X*	10	10
ACORDEON	689	*X*	689
TOTAL: 15 ins.	**2913**	**1086**	**3999**
UTILIZADO: 13 ins.	**2224**	**1076**	**3300**

Fonte: Autoria própria.

Foram considerados todas as amostras de instrumentos de notas normais mantidas, que são chamadas em inglês de *sustain notes*, e notas mantidas com *vibrato*, que é uma vibração causada pelo músico. Nas bases de dados haviam quatro tipos de dinâmicas que os instrumentistas aplicaram ao tocar, notas mantidas, com *vibrato*, *pizzicato* ou *staccato*. As duas últimas dinâmicas presentes na base Alpine não foram consideradas porque são movimentos específicos, restritos a execução do músico e tipo de música tocada, em que são movimentos com o objetivo de conseguir uma nota extremamente curta no tempo, tornando o som do instrumento quase percussivo, e

geralmente são executados em instumentos de cordas. As dinâmicas mencionadas resultam em séreis temporais que se distingue demais na duração do que uma nota comum mantida do mesmo instrumento em questão, essas séries foram consideradas *outliers* e desconsideradas.

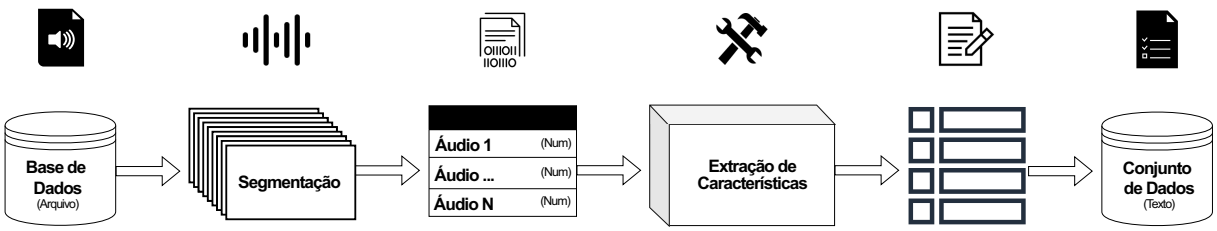
Outras amostras desconsideradas foram algumas dos instrumentos de sopro da base Alpine, onde havia algumas amostras que foram gravadas como *Straight Mute* e *Harmon Mute*, que são gravações com peças acopladas na saída de ar do instrumento. O objetivo dessas peças é a obtenção de uma característica abafada e percepção distante do som do instrumento. O motivo da remoção é porque como o som gravado se difere bastante do que é comum ao instrumento, tanto as características temporais quanto as espectrais foram afetadas, consequentemente, as resultantes são sinais que não representam um padrão comum de características do instrumento.

Os intrumentos *accordion* e *piccolo* foram desconsiderados por não estarem presentes em ambas as bases e não causarem assim uma comparação injusta se tratando das bases. Também, por terem uma quantidade muito desbalanceada de amostras, em que enquanto uma praticamente não agrega ao resultado geral por ter muito poucas amostras, a outra afetaria bastante o desempenho computacional dos modelos por ter uma quantidade muito grande de amostras.

3.3 PRÉ-PROCESSAMENTO

Nesta seção é introduzida a abordagem de todo o processo necessário para se manipular, extrair e salvar as características dos sinais de áudio digital, e na sequência a etapa de pré-processamento é dividida nas etapas da subseções 3.3.1.1, 3.3.1.2 e 3.3.1.3, para um melhor detalhamento da abordagem. Na Figura 19 há uma abstração do fluxograma de dados que representa as etapas do pré-processamento realizado, desde a importação do arquivo bruto de áudio digital na base de dados, até a preparação dos *datasets* finais. No código as etapas desse *pipeline* podem ser executadas separadamente ou de forma orquestrada pela utilização da função ***preprocessing***.

Figura 19 – *Pipeline* de pré-processamento dos dados



Fonte: Autoria própria.

3.3.1 *Pipeline* de Pré-processamento de Dados

Nas subseções em 3.3.1.1, 3.3.1.2 e 3.3.1.3 são detalhadas as etapas principais de pré-processamento representadas na Figura 19.

3.3.1.1 Etapa 1 - Segmentação de Arquivos

Inicialmente os arquivos brutos de áudio digital foram carregados das bases de dados em seus diretórios no *Google Drive* para dentro da ferramenta de programação em notebooks *Google Colaboratory*. Depois foi realizado uma etapa de segmentação de cada arquivo de áudio antes da extração de características, é realizada de forma simples na prática, mas é conceitualmente muito importante para o desempenho de todo o resto do trabalho, essa etapa é representada pelos primeiros dois objetos do fluxograma da Figura 19.

No momento prévio da extração de cada característica, deve ser aplicado uma segmentação dos arquivos em pequenas janelas móveis de análise (*windowing*), mas é importante deixar claro que esta segmentação não está relacionada com um processo de reamostragem ou quantização dos arquivos, mas sim objetivamente relacionada a "repartição" do sinal global em vários segmentos menores. Esse processo garante a análise de forma granular dos sinais para então ser possível extrair as características segmento a segmento. Portanto, a expressão "janela de análise" utilizada neste trabalho se refere à janela utilizada ao percorrer um sinal para extrair suas características em que a série resultante terá uma amostragem própria, mas os sinais de áudio digital permanecerão com amostragens originais após a análise. Todas as configurações originais dos arquivos foram mantidas para manter os dados fiéis ao sinal de áudio digital original, todos os sinais são monofônicos, com uma taxa de amostragem de 44100 Hz e com resolução de 24 $bits$. O tipo de função de janela de análise utilizada pela função da biblioteca *librosa* para analisar os sinais é por padrão a de Hanning (conhecido como *Hanning window*, ou *Hann function*). Abaixo na Tabela 3 são exibidos os parâmetros utilizados para determinar o tamanho e passo das janelas de análise.

Tabela 3 – Parâmetros utilizados nas janelas de análise

Parâmetro	Teste 01	Teste 02	Teste 03	Teste 04
FRAME SIZE	4096	2048	1024	512
HOP LENGHT	2048	1024	512	256

Fonte: Autoria própria.

O parâmetro *frame size* representa o tamanho da janela de análise e o parâmento *hop lenght* representa o tamanho do passo da janela ao longo do tempo, os dois são mensurados por uma quantidade de amostras. Tradicionalmente no campo de processamento de sinais, são utilizados valores na potência de dois pois eles impactam na eficiência da computação de algoritmos de decimação no tempo, como o algoritmo da FFT, em que o princípio da decimação no tempo é mais convenientemente computado considerando o caso especial de N como uma potência inteira de 2. O parâmetro *hop lenght* existe para garantir uma transição mais suave e menos abrupta do sinal resultante entre janelas, consequentemente suavizando o fenômeno do vazamento espectral. O passo foi definido como metade do tamanho da janela, para percorrer o mínimo de passos possíveis e realizar a menor quantidade de sobreposições de amostras.

Foram realizados quatro tipos de teste de janelas de análise por dois motivos, o primeiro fator determinante para essa escolha é a relação conhecida como troca tempo-frequência (ou *time-frequency trade-off*). Essa relação afirma que quanto maior for a janela de análise, maior será a resolução no domínio da frequência e menor resolução no domínio do tempo, e quanto menor a janela de análise, maior será a resolução do sinal no domínio do tempo e menor a resolução no domínio da frequência. Para as janelas de 4096 amostras, as características resultantes já não representam tão bem as variações temporais dos sinais enquanto para 512 as características são bem detalhadas, na Figura 24 no Capítulo 4 de resultados há um exemplo visual dessa troca no domínio do tempo para a extração da característica RMS.

Já o segundo fator determinante foi porque para janelas de análise menores que 512 amostras, por terem uma granularidade muito pequena as características de saída tiveram alta resolução no domínio do tempo e resultaram em séries temporais com uma grande quantidade de amostras, o que impactou diretamente no desempenho dos algoritmos de classificação. Na seção de resultados pode-se observar que o algoritmo SVM com a distância GAK passou a retornar erros para as tentativas de classificação com características resultantes dessa janela de análise, devido ao aumento da complexidade do algoritmo para o grande número de amostras por série.

Todo o processo de carregamento dos dados brutos e segmentação foi realizado utilizando funções das biblioteca *librosa*, em que cada arquivo de áudio é carregado como uma série temporal de valores de ponto flutuante, então cada série é armazenada individualmente em um vetor de tipo *float*. A bilioteca *numpy* permitiu o processamento e manipulação desses vetores, bem como o armazenamento em uma variável de tipo *ndarray* (vetor numérico de n dimensões). No código, a função ***get_instruments_data*** realiza o carregamento dos dados, conversão e salvamento dos sinais para vetores de séries de ponto flutuantes. Uma vez que os dados já foram salvos para

evitar a necessidade e dependência de carregar uma extensa base de dados a cada nova execução do código, foi desenvolvida a função ***load_instruments_data***, que carrega diretamente o arquivo *picke* salvo préviamente com todos os vetores numéricos de cada rótulo de instrumentos.

3.3.1.2 Etapa 2 - Extração de Características

Nesta subseção são detalhadas nas subseções 3.3.1.2.1 e 3.3.1.2.2, as características extraídas neste trabalho que são consideradas clássicas nas tarefas de análise de sinais de áudio, cada qual em seus respectivos domínios de extração. Uma vez que todos os arquivos estão representados em vetores numéricos e já foi realizada a segmentação, o processo de extração ocorre sobre cada uma das janelas segmentadas. Por fim, são aplicadas as equeções matemáticas que definem a extração de cada característica, em que as saídas resultantes são séries temporais. Essa etapa é representada pelo do terceiro e quarto objetos da Figura 19.

Os algoritmos com as equações responsáveis pela extração de cada uma das características foram implementados completamente apenas para os casos do *amplitude envelope* e *band energy ratio*. Para as demais características foram utilizadas funções da biblioteca *librosa*, as quais só dependiam da configuração correta dos parâmetros de entrada, que são os sinais, *frame size*, *hop lenght*, e para o tipo de janela foi mantido o padrão da biblioteca, janela “hann”, de Hanning. Após a extração, cada característica de cada sinal digital de áudio resultou em uma série temporal com uma quantidade de amostras que depende dos parâmetros definidos na etapa 3.3.1.1. Foram extraídas 6 características de cada sinal de áudio analisado, resultando em 6 séries temporais armazenadas que representam cada sinal e sua respectiva classe. As funções de extração no código de cada característica seguem com o nome precedidas por *extract*, como por exemplo *extract_aamplitude_envelope*, todas elas estão relacionadas com as funções *extract_all_features*, *extract_temporal_features* ou *extract_spectral_features*, que por fim são orquestradas pela função principal chamada ***feature_extraction***.

3.3.1.2.1 Domínio do Tempo

As caraterísticas de baixo nível, de escopo temporal, instantâneas, que foram extraídas no domínio do tempo (categorização feita de acordo com a subseção 2.4.1), carregam informações relacionadas as mudanças temporais da forma de onda ao longo de toda a duração dos sinais, ou seja, é possível obter quando aconteceram as variações das mesmas e entender seus comportamentos

ao longo do tempo. Para realizar a extração dessas, somente é necessário se ter posse dos sinais de áudio digital, as características que foram extraídas são respectivamente:

- *Amplitude Envelope (AE)*: Definida na subseção 2.4.2.1, representada na Fig. 9 e no código extraída com a função *extract_amplitude_envelope*.
- *Root-Mean-Square Energy (RMS)* Definida na subseção 2.4.2.2, representada na Fig. 10 e no código extraída com a função *extract_root_mean_square*.
- *Zero-Crossing Rate (ZCR)* Definida na subseção 2.4.2.3, representada na Fig. 11 e no código extraída com a função *extract_zero_crossing_rate*.

3.3.1.2.2 Domínio da Frequência

Com relação as caraterísticas de baixo nível, de escopo temporal, instantâneas, e que são extraídas no domínio da frequência (categorização feita de acordo com a subseção 2.4.1), essas características carregam informações relacionadas a mudança da energia das frequências de acordo com a tonalidade de cada instrumento. Para realizar a extração destas características é necessário além dos sinais de áudio digital, extrair a STFT (definida na subseção 2.3.2) de cada segmento das janelas de análise para cada sinal, no código a função *extract_spectrogram* realiza esse processo. A STFT também é considerada uma característica extraída, mas não é utilizada nesse trabalho com esse propósito, essa transformada se faz necessária para obter-se a representação tempo-frequência em que as respectivas características foram extraídas:

- *Band Energy Ratio (BER)*: Definida na subseção 2.4.2.4, representada na Fig. 12 e no código extraída com a função *extract_band_energy_ratio*.
- *Spectral Centroid (SC)*: Definida na subseção 2.4.2.5, representada na Fig. 13 e no código extraída com a função *extract_spectral_centroid*.
- *Bandwidth (BW)*: Definida na subseção 2.4.2.6, representada na Fig. 14 e no código extraída com a função *extract_spectral_bandwidth*.

3.3.1.3 Etapa 3 - Preparação dos *Datasets*

Essa etapa é representada no fluxograma da Figura 19 pelo quinto e sexto objetos e consistiu no armazenamento das características em uma estrutura adequada para séries temporais, também na preparação, normalização dos dados e pela divisão feita da base de dados de características em base de treino e base de testes.

3.3.1.3.1 Estrutura de Dados

Tendo posse de todas as características extraídas, para otimizar tempo e processamento foi necessário salvar essas características em alguma estrutura de dados evitando ter que realizar a segmentação e extração a cada nova execução. As 6 características de cada sinal foram salvas em um conjunto de dados (*dataset*), essa estrutura pode estar estar em vários formatos e organizações, mas geralmente são de tabelas ou de vetores em que a diferença principal está na quantidade de dimensões que os dados armazenados terão.

Como foram utilizados algoritmos de bibliotecas diferentes neste trabalho e cada biblioteca suportava um formato de *dataset* diferente, para a biblioteca *sktime*, a estrutura de dados utilizada é um *dataset* aninhado, que tem a capacidade de armazenar uma série de valores em cada célula da sua estrutura tabular. Equanto para a biblioteca *tslearn* a estrutura de dados utilizada é um *ndarray* de três dimensões, o qual indexa as informações as armazenando em uma estrutura de dados análoga a um cubo composto de informações. A função ***instrument_features_to_dataframe*** no código foi criada para a geração dos *datasets* finais de ambas as estruturas mencionadas.

3.3.1.3.2 Normalização de Dados

É uma passo importante e deve ser realizado antes de aplicar qualquer modelo de aprendizado caso as características numéricas não estejam padronizadas na mesma escala, evitando que o modelo atribua importâncias diferentes para diferentes tipos de características. A normalização pode ser feita tanto antes, quanto depois da separação da base de treino e base de teste. Neste trabalho foi feita antes da separação com todos os dados utilizando a técnica de normalização conhecida como *Min-Max Scaler*, mas neste caso adaptada por uma função da biblioteca *tslearn* para séries temporais que é chamada de *TimeSeriesScalerMinMax*. Essa técnica normalizou todos os dados numéricos para a escala $[0, 1]$, em que o valor máximo ou mínimo de uma série que representa uma característica que será respectivamente 0 ou 1. No código essa etapa está contida dentro da função ***normalize***.

Outra tarefa que se fez necessária foi uma normalização quanto a duração mínima de todos os sinais de áudio, pois isso influencia diretamente no tamanho das séries de características. Nos casos em que os sinais de áudio digital tiveram suas características extraídas e terminavam antes do comprimento mínimo definido para análise, as suas séries temporais também ficavam mais curtas comparado as outras. Apesar de ambas as bibliotecas de séries temporais utilizadas

detalharem em suas documentações que seus algoritmos tinham suporte para séries temporais de diferentes tamanhos, na utilização do algoritmo KNN da *sktime* aconteceram diversos erros recorrentes relacionados a esse problema de séries temporais de diferentes tamanhos. Portanto se fez necessário a criação de uma função que verificou o tamanho de cada série dos *datasets*, e caso ela for menor que o tamanho padrão das demais, a função completementa as amostras faltants com valor 0, garantindo assim o mesmo tamanho para todas as séries. A função que executa essa verificação e correção se chama ***get_equal_size_df_columns***, ela garante o mesmo tamanho para todas séries nas colunas de características. Apesar de o algoritmo SVM da biblioteca *tslearn* não apresentar problemas, a mesma função foi utilizada também no tratamento das bases de dados utilizadas para garantir o tratamento em todos os casos. Os parâmetros de segmentação das durações definidas para as análises foram de $2.5s$, $5s$, $10s$ e $20s$, em que dois parâmetros são mais curtos e dois mais longos para analisar a qual influência de incluir ou não momento da queda da energia final das notas musicais dos instrumentos, no âmbito da produção musical essa queda é chamada parâmetro *release* do envelope de amplitude de um sinal.

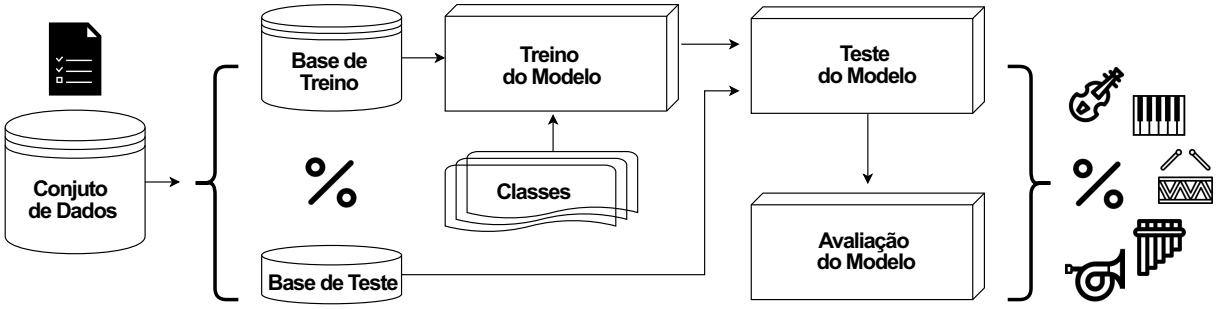
3.3.1.3.3 Divisão de Datasets

É a etapa final antes do algoritmo de classificação, nela o *dataset* final é separado em uma base de dados que será utilizada para o treino do modelo e em uma base de dados que será utilizada para avaliação do modelo já treinado. É importante ressaltar que nenhum dos dados utilizados no treinamento foram também utilizados na base de testes, as duas são completamente diferentes para evitar viéses. Como ambas as bases de dados continham uma grande quantidade de dados para todos os instrumentos musicais, a escolha da proporção para a base de testes não foi impactada e foi definido uma separação de 80% dos dados para a base de treino e 20% para a base de testes, que é uma proporção de divisão muito comum no ramo de aprendizado de máquina. Outro ponto importante é que como cada um dos instrumentos continham diferentes quantidades de dados, foi utilizado o parâmetro *stratify* da função *train_test_split* da biblioteca *sklearn*, para que apesar de diferentes quantidades fosse garantido que a divisão $80/20$ seja válida também para cada porção de dados de instrumentos, não apenas para a base como um todo, e isso garante uma base de treino e teste balanceadas em todos os rótulos de instrumentos. A função no código criada para as verificações e divisão dos *datasets* foi a ***dataset_preparation***.

3.4 MODELO CLASSIFICADOR

Nesta seção serão abordadas uma divisão de etapas essenciais para a construção do modelo classificador, iniciando pela escolha dos algoritmos de aprendizado de máquina para a realização da tarefa de classificação na subseção 3.4.1, depois abordando o processo de treinamento e testes dos modelos na subseção 3.4.2, para na sequência na subseção 3.4.3 pontuar as métricas de avaliação utilizadas. Na figura Figura 20 é exibido o processo padrão de construção de um modelo classificador genérico. No código de execução desse *pipeline* pode ocorrer separadamente ou de forma orquestrada pela função ***time_series_classifier_pipeline***.

Figura 20 – *Pipeline* genérico de um modelo classificador



Fonte: Autoria própria.

3.4.1 Etapa 1 - Algoritmos de Aprendizado

Várias classes de algoritmos tradicionais de aprendizagem podem ser usados para a tarefa de classificação, como classificadores baseados em distância, classificadores probabilísticos, classificadores com redes neurais, entre outros. Se tratando do problema da classificação de instrumentos musicais, dentre as várias abordagens e nos poucos trabalhos em que as características resultaram em séries temporais, essas séries acabaram por receber algum tipo de tratamento estatístico ou redução de dimensionalidade para depois serem utilizadas por um algoritmo tradicional definido para cada caso em específico.

Nesse trabalho ao se considerar as características como séries temporais sem abstraí-las de alguma forma, a complexidade do problema aumenta e surge a exigência de uma abordagem diferente das com algoritmos tradicionais. Para resolver esse problema, os algoritmos que foram utilizados são adaptações de algoritmos tradicionais muito famosos no campo do aprendizado de máquina, que são o algoritmo *K-Nearest Neighbours (KNN)* e as *Support Vector Machines (SVM)*, mas neste caso adaptadas para lidar com séries temporais, em que primeiro é baseado em

distância e o segundo em *kernel*. A diferença principal dessas adaptações está contida no conceito de alinhamento das séries temporais e nas medidas de distância aplicadas que foram tarefas realizadas pelos algoritmos *Dynamic Time Warping (DTW)* e o *Global Alignment Kernel (GAK)*. Essa combinação adaptada de algoritmos, resultou em uma abordagem que ajuda a identificar os padrões entre as séries temporais sem abstrair a informação das características ao longo do tempo, o que permitiu diferenciar as características também por como elas variam seu comportamento no decorrer do tempo. As duas combinações de algoritmos a seguir foram utilizados:

- *K-Nearest Neighbours (KNN)* em conjunto com o *Dynamic Time Warping (DTW)*: O KNN foi definido na subseção 2.5.2 e o DTW na subseção 2.5.2.1. Essa adaptação para séries temporais foi implementada na biblioteca *sktime* com a classe de nome *KNeighborsTimeSeriesClassifier*, o parâmetro *distance* utilizado foi o *"dtw"*.
- *Support Vector Machines (SVM)* em conjunto com o *Global Alignment Kernel (GAK)*: O SVM foi definido na subseção 2.5.3 e o GAK na subseção 2.5.3.1. Essa adaptação para séries temporais foi implementada pela biblioteca *tslearn* com classe de nome *TimeSeriesSVC*, o parâmetro *kernel* utilizado foi o *"gak"*.

Dentre as limitações, foi possível só essas duas combinações entre os algoritmos primeiro por serem implementadas em bibliotecas diferentes, e segundo por serem duas abordagens diferentes em que uma é baseada em distância e a outra em *kernel*, tornando assim as medidas de distâncias atreladas a esses algoritmos devido as próprias adaptações. No código a configuração e orquestração desses algoritmos está dentro da função ***model_train***.

3.4.2 Etapa 2 - Treino e Teste dos Modelos

O treino de modelos é a essência do aprendizado de máquina supervisionado pois é o momento em que é estabelecida uma relação matemática entre as entradas e saídas se baseando em dados de uma base de dados rotulada. Uma vez tendo os *datasets* das base de treino e das classes de algoritmos determinadas anteriormente foi possível realizar a etapa de treinamento utilizando do método *fit* de cada algoritmo. Esse método é a forma de supervisionar o aprendizado do modelo para que ele relacione iterativamente um padrão de características informadas a uma classe também informada. Na Figura 20, a atividade de treino é representada pela abstração do bloco "Treino do Modelo", o qual recebe como entradas uma base de treino composta por todas as 6 características extraídas para cada sinal e também suas respectivas classes, sua saída é o modelo de aprendizado já treinado.

Alguns algoritmos de aprendizado supervisionado exigem que o usuário especifique alguns parâmetros de ajuste que podem otimizar o desempenho do modelo, neste trabalho foram mantidos os parâmetros padrões de cada algoritmo, devido a limitações pela utilização de duas bibliotecas diferentes e da implementação dos métodos das classes desses algoritmos. Desta forma, tornou-se também impraticável a otimização de hiperparâmetros de forma automática, que poderia ser implementada com métodos de otimização, como *grid-search* em conjunto com uma classe de *pipeline* por exemplo.

O teste dos modelos de classificação foi feito a partir da comparação entre as classes preditas e as classes verdadeiras das amostras de uma base de teste. O método *predict* foi o responsável por utilizar os modelos já treinados e fazer uma predição das classes sobre as bases de dados de teste. Na Figura 20, o teste é representado no bloco "Teste do Modelo", o qual representa um modelo já treinado recebendo uma base desconhecida de testes, para então retornar na sua saída as suas inferências de rótulos preditos. No código, o treinamento do modelo ocorre na função ***model_train***, enquanto os testes na função ***model_test***, e ambas além de serem usadas separadamente, podem ser utilizadas de forma orquestrada na função ***time_series_classifier_pipeline***.

3.4.3 Etapa 3 - Avaliação dos Modelos

Nessa etapa, após a obteção das predições do modelo sobre a base de testes, foi possível gerar métricas de avaliação que refletem a assertividade e qualidade de um modelo. Com relação aos problemas de classificação, existem métricas específicas para esses casos, as utilizadas neste trabalho foram acurácia, precisão, *recall*, *F1-score*, matriz de confusão. Também foram coletados indicadores de desempenho como o tempo de predição e análise de complexidades dos modelos. Na Figura 20 a atividade de avaliação é representada no bloco "Avaliação do Modelo" e no código, a avaliação dos modelos é feita com a utilização das funções desenvolvidas ***model_evaluate*** e ***performance***.

Por limitação da implementação das diferentes bibliotecas utilizadas e pelos algoritmos serem adaptações para séries temporais dos algoritmos tradicionais, algumas métricas importantes como *cross-entropy loss*, *AUC score* e *ROC curve* não foram possíveis de serem obtidas. O problema ocorreu devido a ausência dos métodos específicos, métodos pré-requisitos, ou por incompatibilidade do modelo com outras bibliotecas de aprendizado de máquina utilizadas para o cálculos dessas métricas.

3.5 CONSIDERAÇÕES FINAIS

Neste capítulo foram elencadas as ferramentas e tecnologias utilizadas para o desenvolvimento deste trabalho, como o ambiente de desenvolvimento, linguagem de programação, bibliotecas utilizadas e suas respectivas funções. Foi detalhado também quais foram as bases de dados e quantos instrumentos musicais elas continham, também foi descrito quais e porque algumas amostras e instrumentos foram desconsiderados. Na seção de pré-processamento foi abordado o *pipeline* completo com todas as etapas necessárias, desde os tipos de segmentações de arquivos necessárias, extração de características nos domínios do tempo e frequência, a preparação dos *datasets* estruturando, normalizando e fazendo a divisão dos mesmos. Por fim, foi detalhado o processo de contrução de um modelo de classificador, partindo dos algoritmos de aprendizado supervisionado adaptados pra séries temporais utilizados, para então na sequência explicar as etapas de treino, teste, avaliação dos modelos e quais foram as métricas utilizadas.

4 RESULTADOS

Nesse capítulo estão os resultados para as diferentes análises propostas e realizadas neste trabalho. Primeiramente na Seção 4.1 estão os resultados dos modelos para diferentes durações de sinais de áudio, bem como a avaliação dos tempos de execução para esses diferentes valores. Já na Seção 4.2 estão os resultados dos modelos para diferentes tamanhos de janelas de análise e também a avaliação dos seus respectivos tempos de execução. Depois na sequência na seção 4.3 tem uma análise da classificação dos instrumentos observando os melhores resultados para cada base de dados, e por fim na seção 4.4 está uma breve análise sobre os resultados no geral das duas abordagens de algoritmos utilizadas.

Como a divisão das bases de dados de treino e teste foi balanceada pelo parâmetro *stratify* para preservar a proporção de amostras por classe, as métricas acurácia, precisão, *recall* e *f1-score* passam a resultar nos mesmos valores por causa desse balançeamento. Portanto, todos os resultados de *F1-score* exibidos nas tabelas, podem ser interpretados como qualquer uma das outras três métricas mencionadas.

O valor "total de amostras" presente na parte inferior de algumas tabelas representa a quantidade de amostras que definem o tamanho das séries de cada característica extraída, esse número varia de acordo com o comprimento mínimo dos sinais de áudio digital e os parâmetros de janela de análise definidos.

Os resultados para o teste com os parâmetros de janelas de análise com *frame size* = 512 e *hop lenght* = 256 não estão presentes nas tabelas e figuras, pois o algorimo SVM com GAK apresentou erro durante sua execução com esses parâmetros, possívelmente foi um reflexo do aumento da exigência computacional devido ao alto número de amostras.

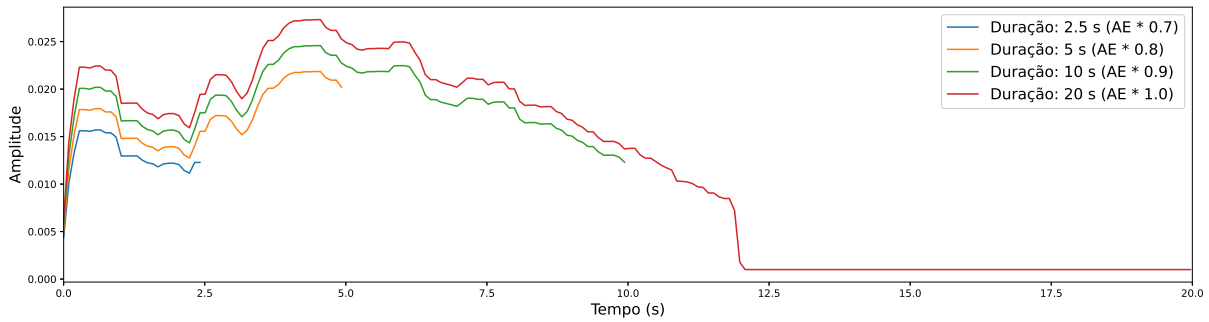
4.1 ANÁLISES PARA DIFERENTES DURAÇÕES DE SINAIS

Na presente seção estão os resultados de avaliações dos modelos para diferentes durações mínimas de sinais, a variação desse parâmetro impacta diretamente no tamanho total dos sinais de áudio, que consequentemente definem também o tamanho das características e sua quantidade de amostras no domínio do tempo ou da frequência. Na base de dados a diferença de duração dos sinais pode acontecer por vários motivos, desde a gravação do instrumento musical em um período mais curto ou mais longo de tempo, até na queda de energia abrupta ou suave do sinal que pode ser ocasionada por fatores como a execução do instrumentista ou pela reverberação

do ambiente onde ocorreu a gravação. Portanto, essa avaliação se faz útil para se analisar qual impacto diferentes durações de sinais tem sobre a resposta do modelo de classificação e também isso se reflete em diferentes bases de dados.

Para esse primeiro caso de análise foram definidos parâmetros de janela de análise fixos *frame size* = 4096 e *hop lenght* = 2048, para que o foco da análise esteja contida na variação da duração mínima dos sinais. Foram escolhidos também os menores parâmetros de janela devido a quantidade de amostras, pois cada nova duração implica em uma quantidade total de amostras maior que impacta diretamente no desempenho dos algoritmos. Na Figura 21 está sendo representado o impacto de diferentes durações de sinais sobre a extração da característica RMS, cada série foi multiplicada por uma constante para que fosse possível exibir os sinais em comparação sem sobrepô-los. Na Tabela 4 estão os resultados para diferentes comprimentos com o algoritmo KNN em conjunto com o DTW sobre as duas bases de dados, e na tabela 5 os resultados para diferentes comprimentos com o algoritmo SVM em conjunto com o GAK também sobre ambas bases de dados.

Figura 21 – Exemplo de uma característica extraída para diferentes durações de sinais



Fonte: Autoria própria.

Tabela 4 – F1-Score do KNN com DTW para diferentes durações de sinais

Duração (s)	2.5	5	10	20
Alpine (Norm.)	74,31%	83,02%	89,90%	88,53%
TinySOL (Norm.)	52,35%	57,75%	65,84%	70,78%
Total de Amostras	27	54	108	216

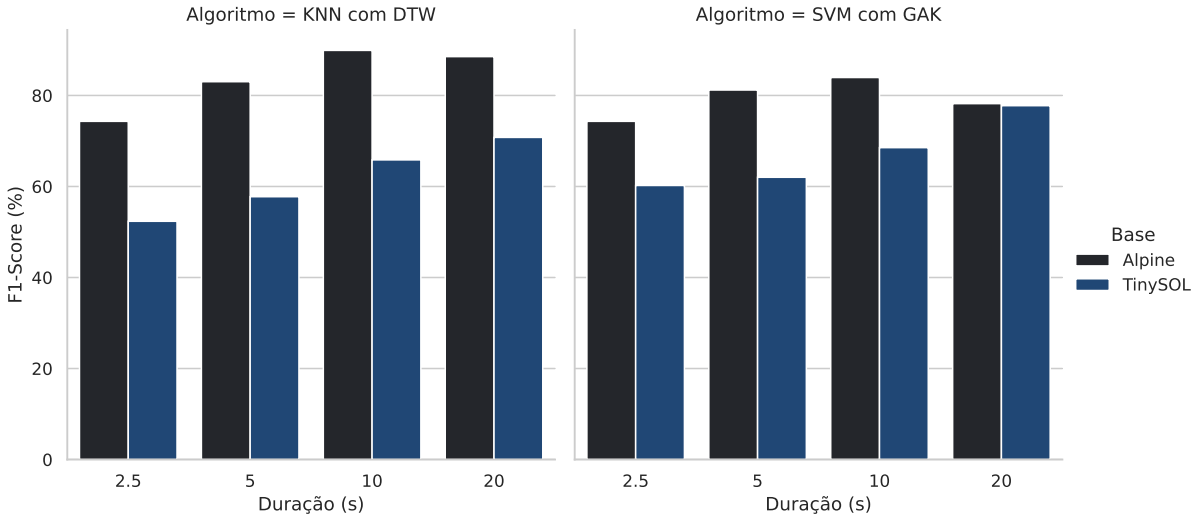
Fonte: Autoria própria.

Tabela 5 – F1-Score do SVM com GAK para diferentes comprimentos de sinais

Base	2.5 (s)	5 (s)	10 (s)	20 (s)
Alpine (Norm.)	74,31%	81,19%	83,94%	78,20%
TinySOL (Norm.)	60,22%	62,02%	68,53%	77,75%
Total de Amostras	27	54	108	216

Fonte: Autoria própria.

Figura 22 – Comparação do F1-Score para diferentes comprimentos de sinais



Fonte: Autoria própria.

Analisando os resultados de F1-score é possível perceber que o resultado da classificação de ambos os modelos foram fortemente impactados pelas diferença de duração dos sinais em ambas as bases de dados. Para a base de dados Alpine, os resultados melhoraram de aproximadamente 9% até 14% comparando o pior e melhor caso, e para a TinySOL, os resultados melhoraram de aproximadamente 17% até 18%, isso reflete em uma melhora na qualidade do modelo para durações maiores consideradas nos sinais de áudio de notas singulares.

Quanto às durações em específico, os melhores resultados para a base de dados Alpine foram com 10 segundos de duração mínima, e para a TinySOL com 20 segundos de duração mínima. Esse resultado é coerente pelo fato que na base Alpine a grande maioria dos sinais está contida e não tem duração maior que esse valor, enquanto para a TinySOL ocorre o mesmo.

Uma outra verificação feita, foi a medição do tempo total de execução no momento da predição pelos modelos de aprendizado treinados. Na Figura 23 e Tabela 6 é possível perceber que ambos algoritmos tem o comportamento exponencial crescente, mas os efeitos de cada tem proporções diferentes, o que está de acordo com o comportamento da complexidade $O(n^2)$ para o KNN com DTW, e $O(n^2(m+\tau))$ para o SVM com GAK com uma curva mais acentuada.

Tabela 6 – Tempo de predição do KNN com DTW para diferentes durações de sinais

Duração (s)	**2.5**	**5**	**10**	**20**
Alpine (Norm.)	14,68s	16,35s	76,18s	295,60s
TinySOL (Norm.)	32,75s	65,27s	321,41s	1114,10s

Fonte: Autoria própria.

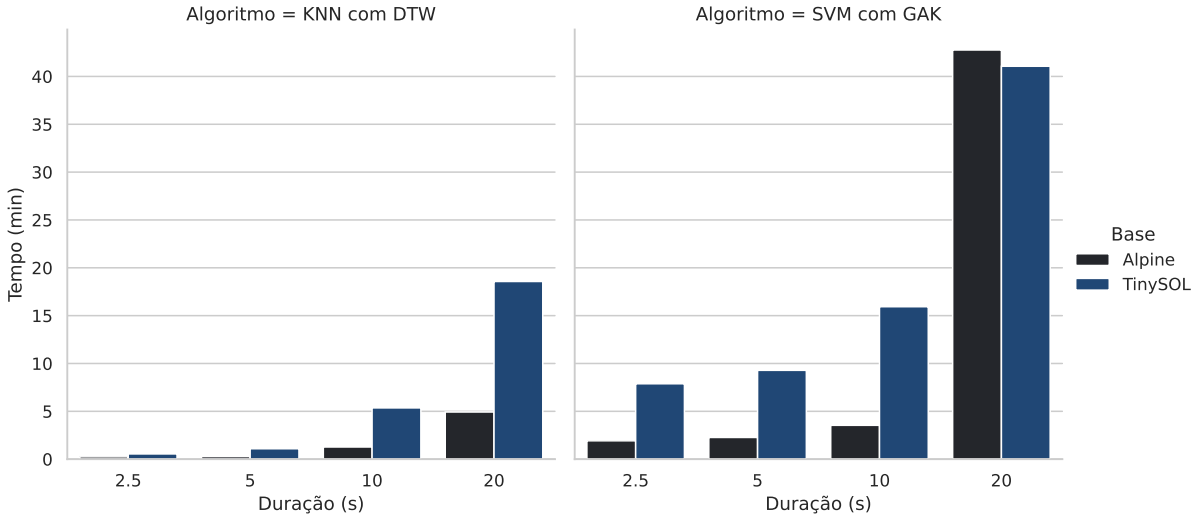
Observando os resultados de tempo de execução fica fácil compreender que quanto maior forem as séries utilizadas, maior vai ser o tempo necessário para a computação das predições

Tabela 7 – Tempo de predição do SVM com GAK para diferentes durações de sinais

Duração (s)	2.5	5	10	20
Alpine (Norm.)	115,10s	136,05s	212,21s	2565,71s
TinySOL (Norm.)	472,74s	557,01s	955,91s	2463,80s

Fonte: Autoria própria.

Figura 23 – Comparação dos tempos de predição para diferentes comprimentos de sinais



Fonte: Autoria própria.

pelo modelo. Para o algoritmo KNN com DTW, entre a menor e a maior duração houve uma diferença de 4 e até 18 minutos de execução dependendo da base de dados, enquanto para o SVM com GAK, entre a menor e a maior duração houve diferenças de 30 e até 40 minutos de execução dependendo da base de dados. Comparando os dois algoritmos sobre essas bases de dados, em todos os casos testados o algoritmo KNN com DTW foi mais rápido do que o SVM com GAK ao realizar a tarefa de predição.

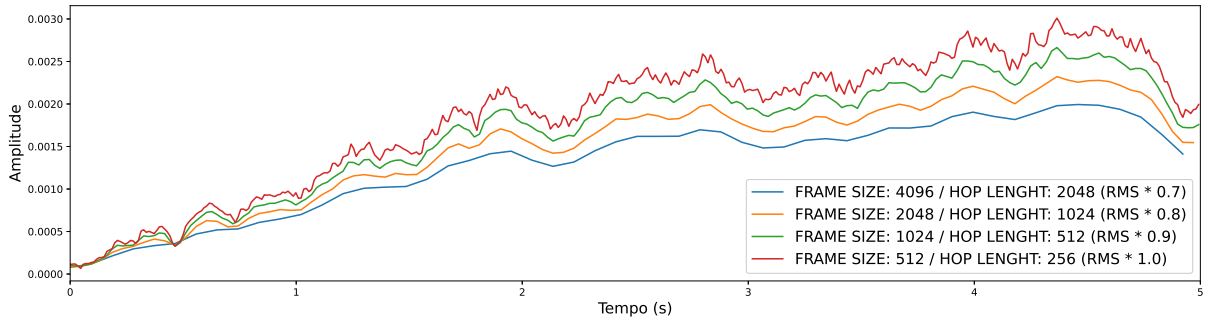
4.2 ANÁLISES PARA DIFERENTES PARÂMETROS DE JANELAS

Nesta seção estão os resultados de avaliações dos modelos para diferentes janelas de análise de extração de características, em que a variação desses parâmetros impacta diretamente na resolução da características no domínio do tempo ou no domínio da frequência, então ela se faz útil para analisar qual impacto diferentes tamanhos de janela de análise utilizadas tem sobre a resposta do modelo de classificação, e se isso também se reflete em diferentes bases de dados.

Para este caso de análise foi definido um limite máximo de 5 segundos como comprimento mínimo dos arquivos de áudio digital de notas musicais, porque apesar de “cortar” os sinais ignorando seu momento de queda, em ambas as bases eles eram na sua maioria maiores que essa

duração. Foram realizados vários testes variando proporcionalmente os parâmetros *frame size* e *hop lenght* da janela de análise, na Figura 24 está representado um exemplo visual do impacto dos diferentes parâmetros utilizados na extração da característica AE. Cada uma das séries foi multiplicada por uma constante para que fosse possível exibir os sinais em comparação sem sobrepô-los. Já na Tabela 8 estão os resultados de diferentes janelas de análise com o algoritmo KNN com o DTW sobre as duas bases de dados, e na tabela 9 os resultados de diferentes janelas de análise com o algoritmo SVM com o GAK sobre ambas bases de dados.

Figura 24 – Exemplo de uma característica extraída para diferentes parâmetros de janelas



Fonte: Autoria própria.

Tabela 8 – F1-Score do KNN com DTW para diferentes janelas

FRAME SIZE / HOP LENGHT	**4096/ 2048**	**2048/ 1024**	**1024/ 512**	**512/ 256**
Alpine (Norm.)	83,02%	81,65%	83,48%	83,48%
TinySOL (Norm.)	57,75%	62,47%	65,84%	64,71%
Total de Amostras	54	108	216	432

Fonte: Autoria própria.

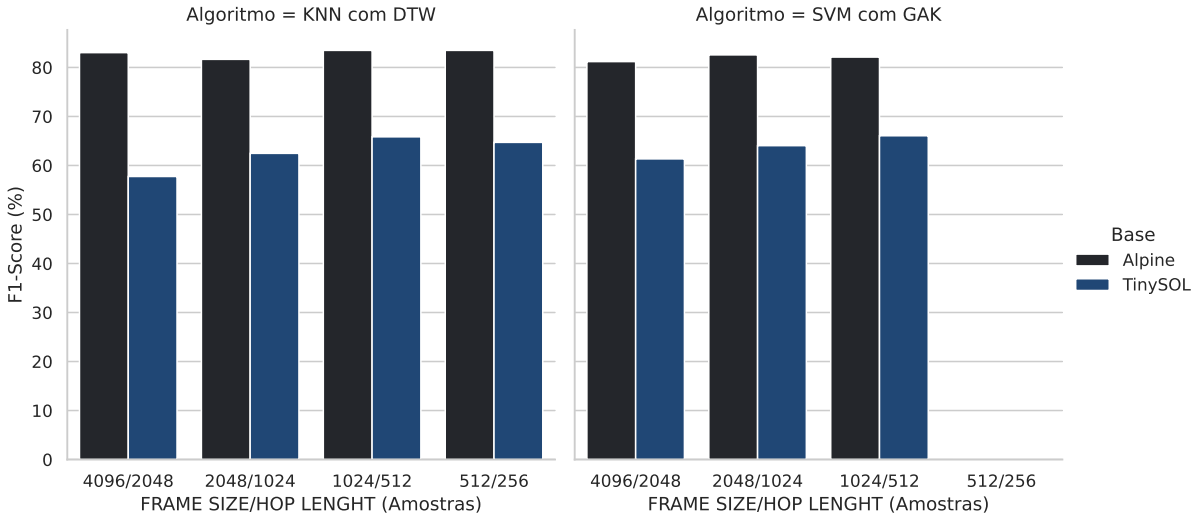
Tabela 9 – F1-Score do SVM com GAK para diferentes janelas

FRAME SIZE / HOP LENGHT	**4096/ 2048**	**2048/ 1024**	**1024/ 512**	**512/ 256**
Alpine (Norm.)	81,19%	82,56%	82,11%	*ERRO*
TinySOL (Norm.)	61,34%	64,04%	66,06%	*ERRO*
Total de Amostras	54	108	216	432

Fonte: Autoria própria.

Analisando os resultados de F1-score é possível perceber que os resultados da classificação de ambos os modelos foram levemente impactados pelas diferença de janelas de análise em ambas as bases de dados, mas para a base de dados Alpine os resultados ficaram variando de aproximadamente 1% até 2% comparando o pior e melhor caso, enquanto para a TinySOL os resultados melhoraram de aproximadamente 4% até 8% comparando os piores com os melhores resultados. Isso reflete que a utilização de diferentes janelas pode melhorar ou apenas manter os

Figura 25 – Comparação do F1-Score para diferentes janelas de análise



Fonte: Autoria própria.

resultados no mesmo intervalo, o que pode se afirmar é que mesmo com algoritmos diferentes a melhora de resultados foi pouco significativa na base de dados TinySOL, portanto a proporção de melhora dos resultados com diferentes janelas parece depender mais da base de dados do que do algoritmo utilizado.

Mais uma verificação feita foi a medição do tempo total de execução no momento da predição pelos modelos de aprendizado treinados para as variadas janelas de análise. Na Figura 23 é possível verificar que ambos algoritmos tem o comportamento exponencial crescente, o que está de acordo com a característica sua complexidade $O(n^2)$ para o KNN com DTW, e $O(n^2(m+\tau))$ para o SVM com GAK com uma curva mais acentuada.

Tabela 10 – Tempo de predição do KNN com DTW para diferentes janelas

Duração (s)	**2.5**	**5**	**10**	**20**
Alpine (Norm.)	16,64s	78,15s	307,56s	1456,63s
TinySOL (Norm.)	71,55s	322,82s	1264,19s	9157,92s

Fonte: Autoria própria.

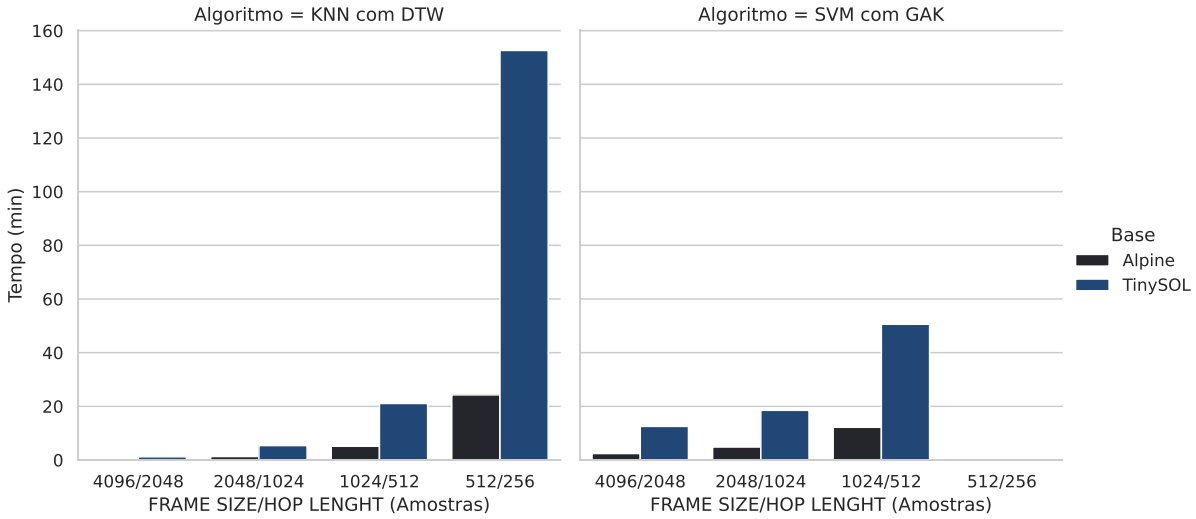
Tabela 11 – Tempo de predição do SVM com GAK para diferentes janelas

Duração (s)	**2.5**	**5**	**10**	**20**
Alpine (Norm.)	145,24s	290,33s	732,09s	*ERRO*
TinySOL (Norm.)	751,74s	1111,06s	3035,34s	*ERRO*

Fonte: Autoria própria.

Observando os resultados de tempo de execução, compreende-se que quanto menor forem as janelas de análise utilizadas maior será a resolução das séries e mais amostras elas terão, consequentemente exigindo maior tempo necessário para a computação das predições.

Figura 26 – Comparação dos tempos de predição para diferentes janelas de análise



Fonte: Autoria própria.

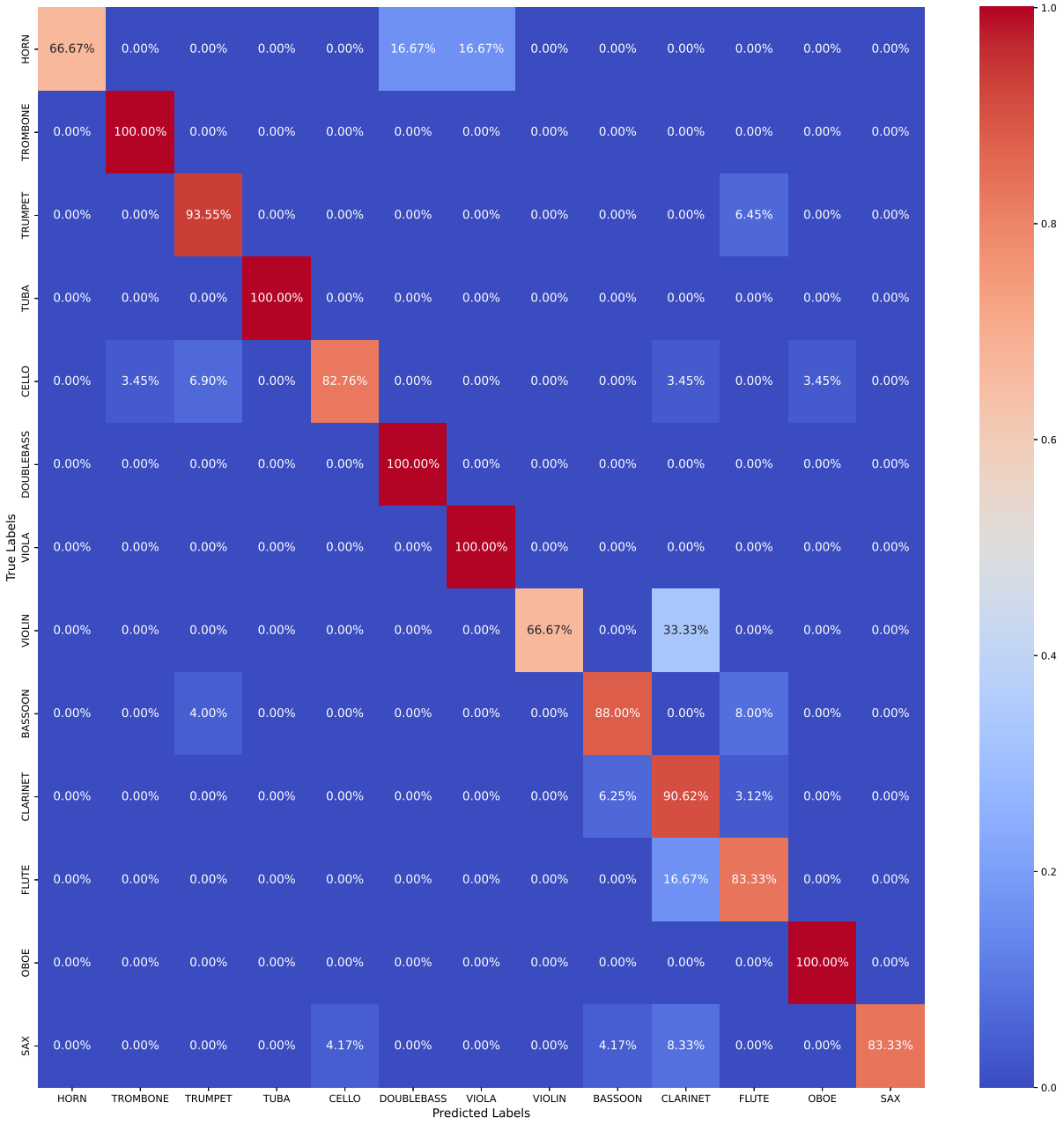
Para o algoritmo KNN com DTW, entre a menor e a maior duração houve uma diferença aproximadamente de 5 até 20 minutos de execução para os três primeiros casos dependendo da base de dados, enquanto para o SVM com GAK, entre a menor e a maior duração houve diferenças de aproximadamente 9 e até 38 minutos de execução para os três primeiros casos dependendo da base de dados. Analisando os tempos é possível notar que comparando os dois algoritmos sobre essas bases de dados, o algoritmo KNN com DTW foi mais rápido ao realizar o processo de predição para diferentes parâmetros de janelas.

4.3 ANÁLISES PARA DIFERENTES INSTRUMENTOS MUSICAIS

Nesta seção são interpretados os dois melhores resultados tanto dos algoritmos quanto para as bases de dados, em que no primeiro caso na Figura 27 está o melhor resultado obtido para a classificação da base de dados Alpine em que o algoritmo KNN com DTW atingiu *F1-Score* de 89.90%, enquanto a Figura 28 exibe o melhor resultado obtido sobre a base de dados TinySOL com o algoritmo SVM com GAK atingindo um resultado de 77.75%. Para interpretar a matriz deve-se olhar nas linhas a classe verdadeira e então cruzar com a informação das colunas que é a classe predita, e então identificar a porcentagem de acerto ou erro por classes.

Analisando os resultados da matriz de confusão pode-se perceber que os instrumentos musicais melhor classificados foram o trombone, tuba, contrabaixo, viola e oboe, todos com 100% das predições corretas. Já destacando os instrumentos que obtiveram uma maior porcentagem de erros na predição de pelo menos 10%, fica evidente que as caracetísticas de uma boa porção

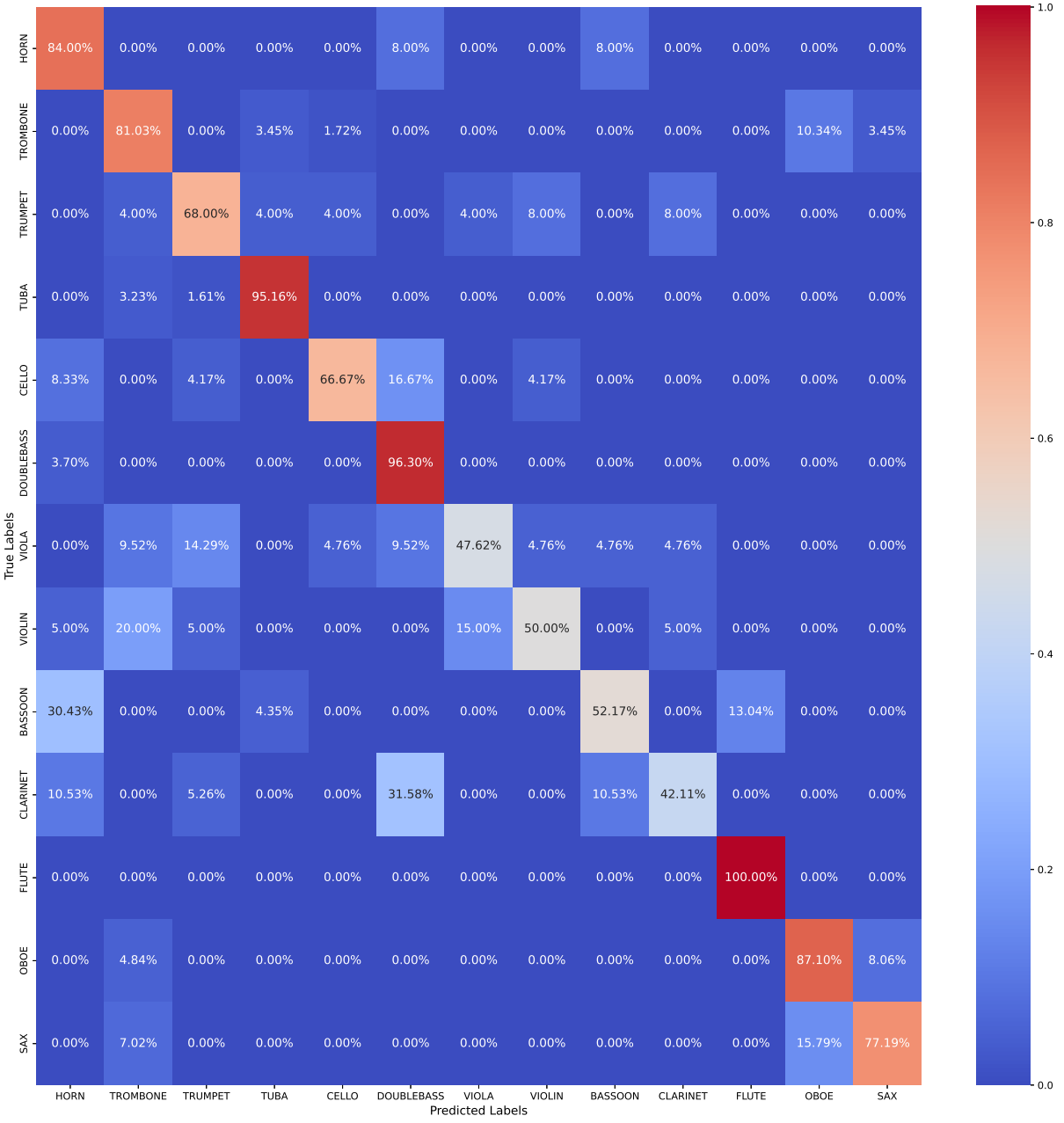
Figura 27 – Matriz de confusão do melhor resultado com a base de dados Alpine



Fonte: Autoria própria.

dos violinos foram confundidas com as do clarinete, as do instrumento trompa tiveram uma boa porção confundidas com as da viola e do contrabaixo, e também uma porcentagem das amostras de flautas foram confundidas com clarinete.

Figura 28 – Matriz de confusão do melhor resultado com a base de dados TinySOL



Fonte: Autoria própria.

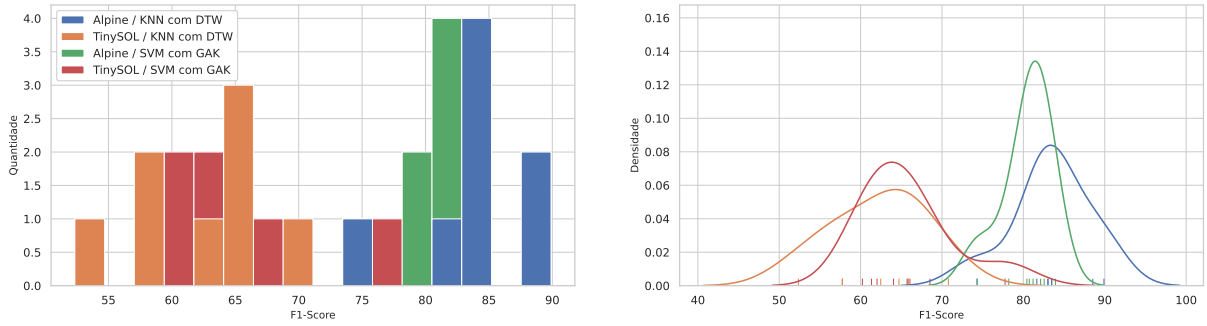
Ao analisar a matriz de confusão do melhor resultado sobre a base de dados TinySOL, foi possível interpretar que os instrumentos musicais melhor classificados foram a flauta, tuba e tumpete, trombone e trompa. Também percebe-se que nesta base de dados houve muito mais sinais com características semelhantes devido a certa quantidade de erros de predição. Dentre os exemplos observados com menor acurácia de acertos e pelo menos 10% das amostras

confundidas com outras classes, o clarinete teve uma boa porção das suas amostras confundidas com o contrabaixo, já o fagote teve várias classes confundidas com o instrumento trompa e algumas com a flauta. As características do violino foram confundidas em alguns casos com a viola e trombone, e a viola foi confundida em alguns casos com o trumpete.

4.4 ANÁLISES PARA DIFERENTES ALGORITMOS

Nesta última seção deste capítulo é comentado sobre as análises de todos os resultados com relação aos algoritmos das abordagens de classificação multivariada de séries temporais utilizadas. Na Figura 29 está sendo representado um histograma e as distribuições normais de todos os resultados de *F1-score* para as duas bases de dados juntando as análises feitas nas seções 4.1 e 4.2, com o objetivo apenas de destacar como as diferenças de resultados foram muito mais afetadas pelas bases de dados do que pelos algoritmos utilizados.

Figura 29 – Histograma e distribuições normais de todos os resultados obtidos



Fonte: Autoria própria.

Analisando ambos os algoritmos de uma forma geral, deve ficar claro que os resultados e desempenho de ambos analisados aqui estão restritos a essas bases de dados específicas. Quando se trata de *F1-score* ou outra das métricas tradicionais coletadas, nos resultados para diferentes com durações o algoritmo KNN com DTW se saiu um pouco melhor que o SVM com GAK na base Alpine, enquanto na TinySOL ocorreu o contrário e o SVM com GAK se sobressaiu. Já nos resultados para diferentes janelas de análise, tanto com a base Alpine quanto com a TinySOL, os resultados ficaram em intervalos muito parecidos e tiveram pouca variabilidade.

Uma afirmação trivial que pode ser abstraída é que para ambos os algoritmos os resultados foram melhores na base Alpine do que na TinySOL, na Figura 29 fica eviente visualmente que os resultados em azul e verde da Alpine foram superiores ao vermelho e laranjado da TinySOL.

Ao observar os tempos de execução dos algoritmos, fica evidente que há uma grande diferença entre os dois quanto aos seus tempos de execução, apesar do algoritmo SVM ser mais rápido que o KNN nos classificadores tradicionais dependendo do seu *kernel*, nesse trabalho existem dois fatores a mais a serem considerados. Primeiro que os algoritmos utilizados nesse trabalho são adaptações para séries temporais, e segundo que a medida de similaridade utilizada no cálculo das distâncias tem um grande impacto nos algoritmos. Pode-se observar que o SVM com GAK acabou tornando-se muito mais lento que o KNN com DTW, reflexo das suas diferenças de complexidades também, que acontece provavelmente devido a exigência de um número maior de computações na utilização da técnica de alinhamento GAK.

4.5 IMPLEMENTAÇÃO E CÓDIGO

O código com a implementação de todas as etapas deste trabalho foi desenvolvido inteiramente em python em um jupyter notebook, ele está documentado e disponível em dois ambientes nos respectivos links: *Google Colaboratory* e *GitHub Repository.*

5 CONCLUSÃO

Além da tarefa de classificação das notas singulares dos instrumentos musicais, foram realizadas também análises para entender os impactos de diferentes parâmetros nas características e nos resultados dos modelos. O objetivo principal dessas análises esteve focado na garantia da qualidade dos dados das características além dos modelos de aprendizado de máquina em si. Como neste trabalho foi proposto uma abordagem que utilizava características instantâneas que resultavam em séries temporais, alguns fatores foram relevados para garantir que a qualidade e comportamento dos sinais fossem representadas com precisão, tanto nos dados dos sinais de áudio digital quanto nas suas características extraídas nos domínios do tempo e frequência.

A primeira análise proposta foi compreender os efeitos de diferentes durações de sinais digitais das notas dos instrumentos musicais e foi discutida na Seção 4.1. Ao avaliar os resultados é possível entender que o fator duração escolhida para a padroniação dos tamanhos dos sinais impacta fortemente nos resultados. Foi interpretado que a melhor escolha para essa duração mínima está relacionada com o tamanho mínimo padrão dos sinais de cada bases de dados específica, pois os resultados em geral melhoraram e até estabilizaram no momento que o tamanho mínimo definido já comportava a grande maioria dos sinais das bases de dados. Então é compreendido que ao escolher o parâmetro de duração mínima padrão, deve-se optar por uma duração que comporte praticamente toda a totalidade e duração dos sinais da base em questão.

Ao tratarmos dos resultados para diferentes janelas de análise utilizadas no momento da extração de características, foi possível perceber nos resultados da Seção 4.2 que os efeitos dessas variações de parâmetros tiveram uma influência positiva nos resultados com a base de dados TinySOL, enquanto para a Alpine os resultados se mantiveram praticamente constantes e variando pouco no mesmo intervalo. Percebeu-se que para essas janelas de análise em específico, a maior ou menor resolução das características extraídas afeta diretamente a amostragem e o nível de detalhe das séries em questão, mas não afeta seu formato global ao ponto de impactar significativamente o resultado dos modelos. Portanto, observando os resultados no geral para os parâmetros avaliados, pode-se afirmar que a escolha dos parâmetros de janela de análise no momento da extração pode manter estável ou trazer melhoras aos resultados do modelo.

Fazendo uma abstração geral da análise dos melhores resultados de classificação com relação a matriz de confusão dos acertos ou erros de certos instrumentos apresentados na Seção 4.3, deve-se ficar claro que não se pode afirmar que tais instrumentos sempre terão tal contribuição independente da base de dados. Mas sim compreender que apesar de algumas dessas

relações se fazerem presentes em todo tipo de sinais de alguns instrumentos, pois fazem parte da característica do seu som, a proporção da influência dessas relações vai variar sempre sobre os resultados e será maior ou menor dependendo de vários fatores como a qualidade dos dados, de como os sinais foram gravados, da técnica do instrumentista, se houve algum tratamento nos sinais, entre outros. Ou seja, como são muitos os influenciadores ao longo do caminho percorrido desde a vibração do som, até o sinal digital final na base de dados, todos vão contribuir posivita ou negativamente para as características dos sinais.

Com relação aos dois algoritmos utilizados nos modelos, tendo como base os resultados da Seção 4.4, não se pode inferir uma comparação relativamente justa entre os dois ao se olhar apenas para as métricas de avaliação obtidas, pois ambos tiveram resultados em intervalos muito próximos tanto alterando duração quanto parâmetros de janela, e o fator que causou a maior variabilidade nos resultados ainda foram as diferentes bases de dados, que já era de se esperar. Ao tratarmos do tempo de execução na predição pelos modelos, em todos os casos os modelos do KNN com DTW foi pelo menos duas vezes mais rápido na predição do que do SVM com GAK em ambas as bases de dados. Apesar de nos algoritmos tradicionais essa relação geralmente apresentar resultados contrários, nessas adaptações utilizadas também há o impacto direto dos algoritmos das medidas de similaridade a serem considerados, impacto que pode causar problemas de escalabilidade. Enquanto o DTW calcula a menor distância analisando uma janela local de amostras, o GAK calcula as distâncias analisando uma janela global de amostras, realizando assim muito mais operações matemáticas para a construção do seu *kernel*, causando a diferença de complexidade e eficiência dos algoritmos.

As técnicas de classificação multivariada de séries temporais aplicadas foram eficientes na resolução da tarefa de classificação automática de instrumentos musicais. Com relação aos resultados, se comparados com os das técnicas as quais utilizam de características como MFCC, imagens de espectrogramas e algoritmos de aprendizado profundo, este trabalho apresentou resultados de classificação inferiores a esses métodos, mas ainda sim muito bons considerando que se trata de uma abordagem nova focada nas características como séries temporais, diferentemente das abordagens tradicionais. Também fica evidente que existem limitações de escalabilidade devido a complexidade dos algoritmos dessa abordagem, e dependem do tamanho e quantidade de séries consideradas.

5.1 LIMITAÇÕES

Por se tratarem de adaptações de algoritmos tradicionais para lidar com problemas de séries temporais, houve nesse trabalho várias limitações atreladas as bibliotecas utilizadas. Dentre as limitações pode-se citar a ausência de alguns métodos que são utilizadas para gerar métricas de avaliação e visualizações. Em alguns casos ao tentar contornar esse problema, os objetos resultantes desses métodos de classes utilizadas eram incompatíveis com outras bibliotecas, a solução para esse problema seria a implementação desses métodos, porém não foi algo necessário para obtenção dos resultados desse trabalho e exigiria um tempo exclusivo dedicado. Outra limitação esteve com relação ao tratamento de séries temporais de diferentes tamanhos por ambas as bibliotecas, em alguns casos os algoritmos apresentaram erro durante a execução por esse motivo. A solução foi criar funções de verificação dos tamanhos das séries temporais de características e quando alguma não atendia o tamanho mínimo padrão, foram acrescentados pontos com o valor zero até o tamanho mínimo ser obtido para resolver esse problema.

5.2 TRABALHOS FUTUROS

Essa abordagem de clasificação de instrumentos musicais utilizando características como séries temporais gera várias novas possibilidades de resolução dessa tarefa. Novas pesquisas podem fazer a avaliação do impacto de diferentes características que em outras pesquisas foram utilizadas como valor global, mas agora como valores instantâneos que variam ao longo do tempo. Dentro do mesmo escopo, pode-se avaliar também o efeito das características no domínio do tempo e domínio da frequência. Outra possibilidade interessante seria extrair conjuntos de coeficientes como MFCC e LPC, mas de forma instantânea, resultando em conjuntos de séries temporais que representam cada coeficiente para cada intervalo de frequências. Então se torna possível compreender o comportamento dos diferentes coeficientes ao longo do tempo e utilizar essa abordagem de classificação multivariada de séries temporais. Por último, como classificação multivariada de séries temporais é um campo de pesquisa relativamente novo e novos algoritmos estão surgindo a cada ano, novas pesquisas podem serem desenvolvidas avaliando o desempenho dos outros algoritmos existentes para a tarefa de classificação de instrumentos musicais.

REFERÊNCIAS

AGOSTINI, G.; LONGARI, M.; POLLASTRI, E. Musical instrument timbres classification with spectral features. In: 2001 IEEE Fourth Workshop on Multimedia Signal Processing (Cat. No.01TH8564). [*S. l.: s. n.*], 2001. P. 97–102. DOI: 10.1109/MMSP.2001.962718.

ALESSIO, M. **Industry Trends: Machine Learning for Commercial Audio Production**. [*S. l.: s. n.*], 2019. Disponível em: https://signalprocessingsociety.org/newsletter/2019/11/industry-trends-machine-learning-commercial-audio-production. Acesso em: 28 fev. 2021.

ALGONAUT. **Algonaut Audio ATLAS 2**. [*S. l.: s. n.*]. Disponível em: https://algonaut.audio/. Acesso em: 11 mai. 2022.

BAGNALL, A. *et al.* The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. **Data Mining and Knowledge Discovery**, v. 31, mai. 2017. DOI: 10.1007/s10618-016-0483-9.

BHALKE, D. G.; RAMA RAO, C. B.; BORMANE, D. S. Dynamic time warping technique for musical instrument recognition for isolated notes. In: 2011 International Conference on Emerging Trends in Electrical and Computer Technology. [*S. l.: s. n.*], 2011. P. 768–771. DOI: 10.1109/ICETECT.2011.5760221.

BOECKING, B. *et al.* Support vector clustering of time series data with alignment kernels. **Pattern Recognition Letters**, v. 45, p. 129–135, 2014. ISSN 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2014.03.015.

CHAKRABORTY, S.; PAREKH, R. Improved Musical Instrument Classification Using Cepstral Coefficients and Neural Networks. In: p. 123–138. ISBN 978-981-13-2344-7. DOI: 10.1007/978-981-13-2345-4_10.

CHANDWADKAR, D. M.; SUTAONE, M. S. Role of features and classifiers on accuracy of identification of musical instruments. In: 2012 2nd National Conference on Computational Intelligence and Signal Processing (CISP). [*S. l.: s. n.*], 2012. P. 66–70. DOI: 10.1109/NCCISP.2012.6189710.

CHETRY, N.; SANDLER, M. Linear Predictive Models for Musical Instrument Identification. In: 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings. [*S. l.: s. n.*], 2006. v. 5, p. v–v. DOI: 10.1109/ICASSP.2006.1661253.

CUTURI, M. Fast global alignment kernels. In: PROCEEDINGS of the 28th International Conference on International Conference on Machine Learning (ICML-11). [*S. l.: s. n.*], 2011. P. 929–936. Disponível em: https://dl.acm.org/doi/10.5555/3104482.3104599.

CUTURI, M. *et al.* A Kernel for Time Series Based on Global Alignments. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07. [*S. l.: s. n.*], 2007. v. 2, p. ii-413-ii–416. DOI: 10.1109/ICASSP.2007.366260.

DE MAN, B.; STABLES, R.; REISS, J. D. **Intelligent Music Production**. [*S. l.*]: Routledge, 2019. P. 218. (Audio Engineering Society Presents). ISBN 1138055190. DOI: https://doi.org/10.4324/9781315166100.

DEAN, R. T. *et al.* **The Oxford Handbook of Computer Music**. [*S. l.*]: OUP USA, 2011. P. 624. (Oxford Handbooks in Music). ISBN 9780199792030. DOI: http://doi.org/10.1093/oxfordhb/9780199792030.001.0001.

DENG, J. D.; SIMMERMACHER, C.; CRANEFIELD, S. A Study on Feature Analysis for Musical Instrument Classification. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, v. 38, n. 2, p. 429–438, 2008. DOI: 10.1109/TSMCB.2007.913394.

DHILLON, I. S.; GUAN, Y.; KULIS, B. Kernel K-Means: Spectral Clustering and Normalized Cuts. In: PROCEEDINGS of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. [*S. l.*]: Association for Computing Machinery, 2004. P. 551–556. DOI: 10.1145/1014052.1014118.

EINSTEIN, A. **William Miller Interview**. Princeton, New Jersey, USA, mai. 1955.

EMANUELE, C. *et al.* **TinySOL: an audio dataset of isolated musical notes**. [*S. l.*]: Zenodo, jan. 2020. DOI: 10.5281/zenodo.3632287.

ERONEN, A.; KLAPURI, A. Musical instrument recognition using cepstral coefficients and temporal features. In: 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100). [*S. l.: s. n.*], 2000. v. 2, ii753–ii756 vol.2. DOI: 10.1109/ICASSP.2000.859069.

ESLING, P.; AGON, C. Multiobjective Time Series Matching for Audio Classification and Retrieval. **IEEE Transactions on Audio, Speech, and Language Processing**, v. 21, n. 10, p. 2057–2072, 2013. DOI: 10.1109/TASL.2013.2265086.

ESLING, P.; AGON, C. Time-Series Data Mining. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 45, n. 1, 2012. ISSN 0360-0300. DOI: 10.1145/2379776.2379788.

FREDERICK, S. O. **THE INSTRUMENTS OF THE ORCHESTRA I: FAMILIES**. [*S. l.: s. n.*], 2019. Disponível em: https://fredericksymphony.org/the-instruments-of-the-orchestra-i-families/. Acesso em: 24 abr. 2022.

HARRIS, C. R. *et al.* Array programming with NumPy. **Nature**, Springer Science e Business Media LLC, v. 585, n. 7825, p. 357–362, 2020. DOI: 10.1038/s41586-020-2649-2.

HERRERA-BOYER, P.; PEETERS, G.; DUBNOV, S. Automatic classification of musical instrument sounds. **Journal of New Music Research**, Taylor & Francis, v. 32, n. 1, p. 3–21, 2003. DOI: 10.1076/jnmr.32.1.3.16798.

HUNTER, J. D. Matplotlib: A 2D graphics environment. **Computing in Science & Engineering**, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.

IZHAKI, R. **Mixing audio: concepts, practices and tools**. Second Edition. Boston: Focal Press, 2012. ISBN 978-0-240-52222-7. DOI: https://doi.org/10.1016/B978-0-240-52222-7.00034-5.

IZOTOPE. **iZotope Neutron 3**. [*S. l.: s. n.*]. Disponível em: https://www.izotope.com/en/products/neutron.html. Acesso em: 11 mai. 2022.

IZOTOPE. **iZotope RX 9**. [*S. l.: s. n.*]. Disponível em: https://www.izotope.com/en/products/rx.html. Acesso em: 11 mai. 2022.

KAMINSKY, I.; MATERKA, A. Automatic source identification of monophonic musical instrument sounds. In: PROCEEDINGS of ICNN'95 - International Conference on Neural Networks. [*S. l.: s. n.*], 1995. v. 1, 189–194 vol.1. DOI: 10.1109/ICNN.1995.488091.

KARTOMI, M. **On Concepts and Classifications of Musical Instruments**. 1st. [*S. l.*]: The University of Chicago Press, 1990. ISBN 978-0226425498.

KLAPURI, A.; DAVY, M. **Signal Processing Methods for Music Transcription**. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387306676. DOI: 10.5555/1200896.

KNEES, P.; SCHEDL, M. **Music similarity and retrieval: an introduction to audio-and web-based strategies**. [*S. l.*]: Springer, 2016. v. 36. ISBN 3662497204. DOI: https://doi.org/10.1007/978-3-662-49722-7.

LEAPWING. **Leapwing Audio Al Schmitt Signature Plugin**. [*S. l.: s. n.*]. Disponível em: https://www.leapwingaudio.com/product/alschmitt/. Acesso em: 11 mai. 2022.

LERCH, A. **An introduction to audio content analysis: Applications in signal processing and music informatics**. [*S. l.*]: Wiley-IEEE Press, 2012. ISBN 978-1118266823. DOI: 10.1002/9781118393550.

LÖNING, M. *et al.* sktime: A Unified Interface for Machine Learning with Time Series. In. DOI: https://doi.org/10.48550/arXiv.1909.07872. eprint: 1909.07872.

MAHARAJ, E. A.; D'URSO, P.; CAIADO, J. **Time series clustering and classification**. [*S. l.*]: Chapman e Hall/CRC, 2019. ISBN 9780429058264. DOI: 10.1201/9780429058264.

MARTIN, K. D.; KIM, Y. E. Musical instrument identification: A pattern-recognition approach. **The Journal of the Acoustical Society of America**, Acoustical Society of America, v. 104, n. 3, p. 1768–1768, 1998. DOI: 10.1121/1.424083.

MCFEE, B. *et al.* librosa: Audio and Music Signal Analysis in Python. In: PROCEEDINGS of the 14th python in science conference. [*S. l.: s. n.*], jan. 2015. v. 8, p. 18–24. DOI: 10.25080/Majora-7b98e3ed-003.

MCKINNEY, W. Data Structures for Statistical Computing in Python. In: WALT, S. van der; MILLMAN, J. (Ed.). **Proceedings of the 9th Python in Science Conference**. [*S. l.: s. n.*], 2010. P. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

MITCHELL, T. M. **Machine Learning**. 1. ed. [*S. l.*]: McGraw-Hill, Inc., 1997. P. 432. ISBN 0070428077.

MONTAGU, J. **Origins and development of musical instruments**. [*S. l.*]: Scarecrow Press, 2007. ISBN 0810856573.

MUDA, L.; BEGAM, M.; ELAMVAZUTHI, I. Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques. **CoRR**, abs/1003.4083, 2010. DOI: https://doi.org/10.48550/arXiv.1003.4083.

MÜLLER, M. **Fundamentals of music processing: Audio, analysis, algorithms, applications**. [*S. l.*]: Springer, 2015. ISBN 3319357654. DOI: 10.1007/978-3-319-21945-5.

NICK, M. **Cleaner, Easier Stem Isolation with Music Rebalance in RX**. [*S. l.: s. n.*], 2020. Disponível em: https://www.izotope.com/en/learn/stem-isolation-music-rebalance.html. Acesso em: 11 mai. 2022.

OPPENHEIM, A. V.; WILLSKY, A. S.; NAWAB, S. H. **Signals & Systems (2nd Ed.)** [*S. l.*]: Prentice-Hall, Inc., 1996. ISBN 0138147574.

PEDREGOSA, F. *et al.* Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PIKRAKIS, A.; THEODORIDIS, S.; KAMAROTOS, D. Recognition of isolated musical patterns using context dependent dynamic time warping. **IEEE Transactions on Speech and Audio Processing**, v. 11, n. 3, p. 175–183, 2003. DOI: 10.1109/TSA.2003.811533.

RUIZ, A. P. *et al.* The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. **Data Mining and Knowledge Discovery**, v. 35, p. 401–449, 2021. DOI: 10.1007/s10618-020-00727-3.

RUSSELL, S.; NORVIG, P. **Artificial intelligence: a modern approach**. 4. ed. [*S. l.*]: Pearson Education, Inc., 2020. P. 1136. ISBN 0134610997.

SAKOE, H.; CHIBA, S. Dynamic programming algorithm optimization for spoken word recognition. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 26, p. 43–49, 1978. DOI: 10.1109/TASSP.1978.1163055.

SHAHAN, N. **iZotope and Assistive Audio Technology**. [*S. l.: s. n.*], 2018. Disponível em: https://www.izotope.com/en/learn/izotope-and-assistive-audio-technology.html. Acesso em: 3 mar. 2021.

SMITH, S. **Digital signal processing: a practical guide for engineers and scientists**. 1. ed. [*S. l.*]: Elsevier, 2003. ISBN 9780080477329. DOI: https://doi.org/10.1016/B978-0-7506-7444-7.X5036-5.

TAVENARD, R. *et al.* Tslearn, A Machine Learning Toolkit for Time Series Data. **Journal of Machine Learning Research**, v. 21, n. 118, p. 1–6, 2020. Disponível em: http://jmlr.org/papers/v21/20-091.html.

TEAM, A. P. **The Alphine Project: A Free Orchestral Instrument Sample Library**. [*S. l.: s. n.*], 2021. Disponível em: https://alpineproject.wixsite.com/main.

VON HORNBOSTEL, E. M.; SACHS, C. Classification of Musical Instruments: Translated from the Original German by Anthony Baines and Klaus P. Wachsmann. **The Galpin Society Journal**, Galpin Society, v. 14, p. 3–29, 1961. DOI: http://doi.org/10.2307/842168.

WASKOM, M. L. seaborn: statistical data visualization. **Journal of Open Source Software**, The Open Journal, v. 6, n. 60, p. 3021, 2021. DOI: 10.21105/joss.03021.

WATKINSON, J. **The art of digital audio**. [*S. l.*]: Taylor & Francis, 2001. ISBN 0240515870. DOI: https://doi.org/10.4324/9780080499369.

WEIHS, C. *et al.* **Music Data Analysis: Foundations and Applications**. 1st. [*S. l.*]: Chapman e Hall/CRC, 2016. P. 694. ISBN 9780367872816. Disponível em: https://doi.org/10.1201/9781315370996.

XI, X. *et al.* Fast Time Series Classification Using Numerosity Reduction. In: PROCEEDINGS of the 23rd International Conference on Machine Learning (ICML-06). [*S. l.: s. n.*], 2006. P. 1033–1040. DOI: 10.1145/1143844.1143974.

XLN. **XLN Audio XO**. [*S. l.: s. n.*]. Disponível em: https://www.xlnaudio.com/products/xo. Acesso em: 11 mai. 2022.

ANEXO A — LEI N.º 9.610, DE 19 DE FEVEREIRO DE 1998: DIREITOS AUTORAIS / DISPOSIÇÕES PRELIMINARES

Presidência da República
Casa Civil
Subchefia para Assuntos Jurídicos

LEI Nº 9.610, DE 19 DE FEVEREIRO DE 1998.

Mensagem de veto

Regulamento

Altera, atualiza e consolida a legislação sobre direitos autorais e dá outras providências.

O PRESIDENTE DA REPÚBLICA Faço saber que o Congresso Nacional decreta e eu sanciono a seguinte Lei:

Título I

Disposições Preliminares

Art. 1º Esta Lei regula os direitos autorais, entendendo-se sob esta denominação os direitos de autor e os que lhes são conexos.

Art. 2º Os estrangeiros domiciliados no exterior gozarão da proteção assegurada nos acordos, convenções e tratados em vigor no Brasil.

Parágrafo único. Aplica-se o disposto nesta Lei aos nacionais ou pessoas domiciliadas em país que assegure aos brasileiros ou pessoas domiciliadas no Brasil a reciprocidade na proteção aos direitos autorais ou equivalentes.

Art. 3º Os direitos autorais reputam-se, para os efeitos legais, bens móveis.

Art. 4º Interpretam-se restritivamente os negócios jurídicos sobre os direitos autorais.

Art. 5º Para os efeitos desta Lei, considera-se:

I - publicação - o oferecimento de obra literária, artística ou científica ao conhecimento do público, com o consentimento do autor, ou de qualquer outro titular de direito de autor, por qualquer forma ou processo;

II - transmissão ou emissão - a difusão de sons ou de sons e imagens, por meio de ondas radioelétricas; sinais de satélite; fio, cabo ou outro condutor; meios óticos ou qualquer outro processo eletromagnético;

III - retransmissão - a emissão simultânea da transmissão de uma empresa por outra;

IV - distribuição - a colocação à disposição do público do original ou cópia de obras literárias, artísticas ou científicas, interpretações ou execuções fixadas e fonogramas, mediante a venda, locação ou qualquer outra forma de transferência de propriedade ou posse;

V - comunicação ao público - ato mediante o qual a obra é colocada ao alcance do público, por qualquer meio ou procedimento e que não consista na distribuição de exemplares;

VI - reprodução - a cópia de um ou vários exemplares de uma obra literária, artística ou científica ou de um fonograma, de qualquer forma tangível, incluindo qualquer armazenamento permanente ou temporário por meios eletrônicos ou qualquer outro meio de fixação que venha a ser desenvolvido;

VII - contrafação - a reprodução não autorizada;

VIII - obra:

a) em co-autoria - quando é criada em comum, por dois ou mais autores;

b) anônima - quando não se indica o nome do autor, por sua vontade ou por ser desconhecido;

c) pseudônima - quando o autor se oculta sob nome suposto;

d) inédita - a que não haja sido objeto de publicação;

e) póstuma - a que se publique após a morte do autor;

f) originária - a criação primígena;

g) derivada - a que, constituindo criação intelectual nova, resulta da transformação de obra originária;

h) coletiva - a criada por iniciativa, organização e responsabilidade de uma pessoa física ou jurídica, que a publica sob seu nome ou marca e que é constituída pela participação de diferentes autores, cujas contribuições se fundem numa criação autônoma;

i) audiovisual - a que resulta da fixação de imagens com ou sem som, que tenha a finalidade de criar, por meio de sua reprodução, a impressão de movimento, independentemente dos processos de sua captação, do suporte usado inicial ou posteriormente para fixá-lo, bem como dos meios utilizados para sua veiculação;

IX - fonograma - toda fixação de sons de uma execução ou interpretação ou de outros sons, ou de uma representação de sons que não seja uma fixação incluída em uma obra audiovisual;

X - editor - a pessoa física ou jurídica à qual se atribui o direito exclusivo de reprodução da obra e o dever de divulgá-la, nos limites previstos no contrato de edição;

XI - produtor - a pessoa física ou jurídica que toma a iniciativa e tem a responsabilidade econômica da primeira fixação do fonograma ou da obra audiovisual, qualquer que seja a natureza do suporte utilizado*;*

XII - radiodifusão - a transmissão sem fio, inclusive por satélites, de sons ou imagens e sons ou das representações desses, para recepção ao público e a transmissão de sinais codificados, quando os meios de decodificação sejam oferecidos ao público pelo organismo de radiodifusão ou com seu consentimento;

XIII - artistas intérpretes ou executantes - todos os atores, cantores, músicos, bailarinos ou outras pessoas que representem um papel, cantem, recitem, declamem, interpretem ou executem em qualquer forma obras literárias ou artísticas ou expressões do folclore.

XIV - titular originário - o autor de obra intelectual, o intérprete, o executante, o produtor fonográfico e as empresas de radiodifusão. (Incluído pela Lei nº 12.853, de 2013)

Art. 6º Não serão de domínio da União, dos Estados, do Distrito Federal ou dos Municípios as obras por eles simplesmente subvencionadas.

...

Texto completo da lei:

ÍNDICE REMISSIVO