UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Mecânica

Cassiano Sergio Noventa Corrêa Bueno

Modelo de aprendizado profundo para identificação de sinais por meio de sensoriamento remoto de vibrações

CAMPINAS
2024

Cassiano Sergio Noventa Corrêa Bueno

Modelo de aprendizado profundo para identificação de sinais por meio de sensoriamento remoto de vibrações

Dissertação apresentada à Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Mecânica, na Área de Projetos Mecânicos.

Orientador: Prof. Dr. Josué Labaki Silva

ESTE TRABALHO CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELO ALUNO CASSIANO SERGIO NOVENTA CORRÊA BUENO, E ORIENTADO PELO PROF. DR JOSUÉ LABAKI SILVA.

CAMPINAS
2024

Ficha catalográfica
Universidade Estadual de Campinas (UNICAMP)
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

B862m

Bueno, Cassiano Sergio Noventa Correa, 1995-
Modelo de aprendizado profundo para identificação de sinais por meio de sensoriamento remoto de vibrações / Cassiano Sergio Noventa Correa Bueno. – Campinas, SP : [s.n.], 2024.

Orientador: Josué Labaki Silva.
Dissertação (mestrado) – Universidade Estadual de Campinas (UNICAMP), Faculdade de Engenharia Mecânica.

1. Porcessamento de sinais. 2. Aprendizado profundo. 3. Vibração. 4. Redes neurais profundas. I. Labaki, Josué, 1982-. II. Universidade Estadual de Campinas (UNICAMP). Faculdade de Engenharia Mecânica. III. Título.

Informações Complementares

Título em outro idioma: Deep learning model for signal identification through remote vibration sensing
Palavras-chave em inglês:
Signal processing
Deep learning
Vibration
Deep neural network
Área de concentração: Mecânica dos Sólidos e Projeto Mecânico
Titulação: Mestre em Engenharia Mecânica
Banca examinadora:
Josué Labaki Silva [Orientador]
Éric Fujiwara
Renan Ramalho Geraldes
Data de defesa: 14-06-2024
Programa de Pós-Graduação: Engenharia Mecânica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0009-0006-0033-5419
- Currículo Lattes do autor: http://lattes.cnpq.br/2931452392402446

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

DISSERTAÇÃO DE MESTRADO ACADÊMICO

Modelo de aprendizado profundo para identificação de sinais por meio de sensoriamento remoto de vibrações

Autor: Cassiano Sergio Noventa Corrêa Bueno
Orientador: Prof. Dr. Josué Labaki Silva

A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:

Prof. Dr. Josué Labaki Silva
DSI/FEM/ UNICAMP

Prof. Dr. Eric Fujiwara
DSI/FEM/UNICAMP

Dr. Renan Ramalho Geraldes
LNLS/CNPEM

A Ata de Defesa com as respectivas assinaturas dos membros encontra-se no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 14 de junho de 2024.

Agradecimentos

Primeiramente, gostaria de expressar minha profunda gratidão ao meu professor orientador Prof. Dr. Josué Labaki, por toda a paciência e dedicação ao longo deste trabalho. Sua orientação cuidadosa e experiente foi fundamental para o sucesso desta dissertação de mestrado.

Além disso, gostaria de agradecer minha família e amigos pelo apoio constante durante todo o meu período de estudos. Suas palavras de encorajamento foram de imensa ajuda para mim, permitindo-me focar totalmente na minha pesquisa. Também gostaria de agradecer aos meus colegas de trabalho, que sempre me incentivaram e me ajudaram a enfrentar os desafios da vida acadêmica e profissional. Seus conselhos e compartilhamento de experiências enriqueceram meu trabalho e me deram forças para continuar avançando mesmo diante das dificuldades. Em suma, agradeço a todos que me apoiaram nesta jornada acadêmica.

Resumo

Redes neurais são cada vez mais utilizadas para identificação de sinais e imagens, e o uso de tais ferramentas tem atingido níveis cada vez mais complexos nos campos da engenharia, economia, medicina e ciências naturais. O princípio matemático das redes neurais possibilita a esta técnica solução de vários problemas no campo de tratamento, segmentação e classificação de imagens, sinais e padrões, sendo hoje em dia, um dos campos mais estudados no âmbito do aprendizado de máquina. Diferentes tipos de rede já se mostraram eficazes para identificação de atividades humanas, falhas em dispositivos mecânico, correção semântica, otimização de processos, entre outros. Neste trabalho, vamos explorar diferentes modelos de redes neurais para identificação de sinais de aceleração resultante da digitação de caracteres em um teclado de computador. As ondas geradas pela digitação se propagarão pela mesa e serão medidas por um acelerômetro instalado próximo ao teclado e, a posteriori, os sinais coletados são tratados e inseridos em redes neurais para classificação nas classes (teclas) disponíveis. Este trabalho busca analisar quais os melhores tratamentos que esse sinal pode receber, bem como avaliar diferentes tipos de rede (convolucionais, LSTM, totalmente conectadas) e métodos de treinamento, a fim de se obter uma maior taxa de classificações corretas e a possibilidade de se decodificar textos digitados.

Palavras-Chave: keylogger, processamento de sinais, deep learning, redes neurais, vibração.

Abstract

Neural networks are increasingly being used for signal and image identification, and the use of such tools has reached increasingly complex levels in the fields of engineering, economics, medicine, and natural sciences. The mathematical principle of neural networks enables this technique to solve various problems in the field of image processing, image segmentation, and image and signal classification, making it one of the most studied fields in machine learning today. Several types of networks have already proven effective in identifying human activities, and failures in mechanical devices, among others. In this work, we will explore different models of neural networks to identify acceleration signals from character typing on a computer keyboard. The waves will propagate by the table and will be measured by an accelerometer installed near the keyboard, and then the collected signals will be processed and inserted into neural networks for classification into available classes (keys). This work aims to analyze the best treatments this signal can receive, as well as evaluate distinct types of networks (convolutional, LSTM, fully connected) and training methods, to achieve a higher rate of correct classifications and the possibility of decoding typed texts.

Key Word: keylogger, signal processing, deep learning, neural networks, vibration.

Lista de Ilustrações

Figura 1 - Imagem da reflexão sísmica resultado de prospecção de solo. O eixo vertical se refere ao tempo viajado pelas ondas, e o eixo horizontal, CMP (Common Midpoint), se refere as diferentes posições de medida do experimento. (Fonte: COELHO 2019) 16
Figura 2 - Fluxograma de dados para o ataque e decodificação de senhas. 19
Figura 3 - Compração entre a porcentagem de recuperação de palavras em MAITI 2016, MARQUARDT 2011 e BERGER 2006. (Fonte: MAITI 2022) 25
Figura 4 - Comparação entre espectrogramas normais e ruidosos. (Fonte: LEE 2018) 27
Figura 5 - Configuração para medida inicial com acelerômetro na posição I. 30
Figura 6 - Conjuntos de teclas utilizadas para construção dos dados de treinamento das redes neurais artificiais. Conjunto de teclas distantes “1”, “P”, “Z” e “:” (destacados em azul). Conjunto de teclas próximas “A”, “S”, “Z” e “X” (destacados em verde). 34
Figura 7 - Esquemático das posições I, II, III, IV medidas com o acelerômetro durante a digitação das teclas de interesse. 35
Figura 8 – Malha 3x3 utilizada para medida modal da mesa de escritório utilizada nos experimentos e localização do ponto de impacto. 37
Figura 9 - Excitação e resposta do acelerômetro para o impacto na posição 1 da medida. 38
Figura 10 - Funções de resposta em frequência para o ponto de medida 1 nas três direções. . 40
Figura 11 - Exemplo de um sinal de aceração medido para a tecla “1”, com acelerômetro posicionado no ponto I. 42
Figura 12 - Espectrograma dos 10s iniciais da medida para a tecla “1” na direção Y. 42
Figura 13 - Série temporal de aceleração dos dados de um único impacto da tecla [Z]. 45
Figura 14 - Transformada de Fourier para os dados temporais da tecla [Z] mostrados na Figura 13. 46
Figura 15 - Espectrograma da aceleração na direção Y da tecla “Z” mostrada na Figura 13. . 46
Figura 16 – Escalograma da transformada de Wavelet contínua do sinal das teclas mostrada na Figura 13. 47
Figura 17 - Decomposição utilizando DWT em 8 camadas, com a janela Symlet no sinal de exemplo do tipo chrip. 49
Figura 18 - Decomposição utilizando DWT em 8 camadas, com a janela Symlet no sinal temporal da Figura 13. 50
Figura 19 - Esquemático de um neurônio de uma camada escondida mostrando o fluxo de informação através dos pesos, do bias e da função de ativação. 52

Figura 20 - Esquema de uma rede neural totalmente conectada com 2 camadas escondidas. . 55
Figura 21 - Função de ativação sigmoid avaliada com entradas de -5 a 5. 56
Figura 22 - Exemplo de sinal temporal para cada uma das teclas do conjunto de teclas distantes. 59
Figura 23 - Matriz de confusão para o experimento com rede totalmente conectada, entrada como sinal temporal de 1s e número de camadas escondidas de 16. À esquerda a matriz de confusão para as teclas próximas e à direita a matriz de confusão para as teclas distantes. 60
Figura 24 – Lado positivo da Transformada de Fourier para impactos do conjunto de teclas distantes. 62
Figura 25 - Matriz de confusão para o experimento com rede totalmente conectada, entrada como sinal no domínio da frequência de 1 a 5000Hz e número de camadas escondidas de 16. À esquerda, a matriz de confusão para as teclas próximas e, à direita, a matriz de confusão para as teclas distantes. 63
Figura 26 - Comparação da acurácia de redes neurais completamente conectadas, com diferente número de camadas, para entrada com dados temporais e em frequência, para o conjunto de teclas próximas. 64
Figura 27 - Comparação da acurácia de redes neurais completamente conectadas, com diferente número de camadas, para entrada com dados temporais e em frequência, para o conjunto de teclas distantes. 65
Figura 28 – Exemplo de espectrograma normalizado em escalas de cinza. 66
Figura 29 - Curvas de perda para os dados de treinamento e validação correspondentes à rede neural treinada com dados em Y para o conjunto de teclas distantes. 67
Figura 30 - Matriz de confusão para o conjunto de teclas distantes quando o acelerômetro estava posicionado na configuração IV. 72
Figura 31 – Célula unitária de uma rede neural do tipo LSTM. 74
Figura 32 - Exemplo de uma camada convolucional, onde um filtro (centro) percorre a figura de entrada (esquerda) resultando na convolução mostrada à direita. 76
Figura 33 - Exemplo da camada de pooling, onde a janela azul varre a imagem (esquerda) e seleciona o maior valor de cada posição. 78
Figura 34 - Exemplo da arquitetura completa de uma rede neural convolucional simples. 78
Figura 35 - Espectrograma normalizado passado como estrada para a rede neural convolucional, com tamanho de 128 x 128 pixels. 79
Figura 36 - Curva de aprendizado para o caso de teclas próximas da rede neural LeNet-5 modificada. 81

Figura 37 - Espectrograma das 3 direções disponíveis, X, Y, e Z, onde cada cor da escala RGB foi alimentada com um espectrograma de uma direção única, formando a imagem colorida utilizada como entrada para a rede neural. 82
Figura 38 – Escalograma da CWT normalizada entre 0 e 256, apresentada em escalas de cinza. 83
Figura 39 - Filtros de uma camada convolucional, mostrando o aprendizado de detalhes de diferentes formas e texturas (MATLAB CNN 2023). 85
Figura 40 - Evolução do melhor individuo por geração ao longo da otimização. 90
Figura 41 - Exemplos das diferentes superposições que serão utilizadas na Seção 5. 96
Figura 42 - Exemplo dos cortes em sinais de 1s para treinamento das redes com sinais menores. 98
Figura 43 – Sinal completo para as teclas distantes com superposição de 0,4s. 102
Figura 44 - Histograma de acurácia de saída da rede convolucional de 0.2s quando avaliado com dados superposto mostrados na Figura 43. As linhas tracejadas verticais demarcam os intervalos de tempo que foram alimentados a rede. 103
Figura 45 - Diagrama de resumo dos resultados encontrados até o presente momento. 105

Lista de Tabelas

Tabela 1 - Probabilidade de decodificação de sequências de caracteres com 4 e 8 dígitos, com diferentes acurácias de redes neurais para teclas individuais. 18
Tabela 2- Especificações do acelerômetro Kistler triaxial 8762A5. 32
Tabela 3 - Especificações do conversor analógico digital NI USB-4431. 33
Tabela 4 - Conjunto de dados de teclas próximas e distantes e número de impactos coletados para treinamento das redes neurais. 36
Tabela 5 - Tabela de parâmetros do martelo de impacto Kistler 9722A500. 37
Tabela 6 - Frequências naturais do sistema até 100Hz e descrição dos modos de vibrar. 40
Tabela 7 - Detalhes extraídos de cada uma das sub-bandas de frequência utilizando a DWT. 50
Tabela 8 – Resultados de acurácia obtidos no treinamento das redes neurais totalmente conectadas com diferentes números de camadas escondidas para os dados no tempo. 59
Tabela 9 - Resultados de acurácia das redes neurais totalmente conectadas para os dados de entrada da Transformada de Fourier. 63
Tabela 10 – Resultados de acurácia para as redes totalmente conectas treinadas com dados de espectrogramas em diferentes direções. 67
Tabela 11 - Resultados de acurácia das redes neurais totalmente conectadas para os dados de entrada da extração de detalhes utilizando janela Symlet com 5 coeficientes. 70
Tabela 12 - Resultados de acurácia das redes neurais totalmente conectadas para os dados de entrada da extração de detalhes utilizando janela Daubechies com 2 coeficientes. 70
Tabela 13 - Resultado de acurácia da rede totalmente conectada quando avaliada utilizando a técnica de extração de detalhes, para os diferentes pontos de medida do acelerômetro. 72
Tabela 14 - Arquitetura da rede LeNet-5 modificada utilizada. 80
Tabela 15 - Acurácia encontrada utilizando a rede LeNet-5 modificada para ambos os conjuntos de dados. 81
Tabela 16 - Comparação entre os resultados de acurácia das redes treinadas com sinais em 1 direção e 3 direções. 82
Tabela 17 - Comparação de resultados de alguns casos apresentados até o momento. 84
Tabela 18 - Comparação de acurácia para treinamento utilizando transfer learning e entradas de espectrograma e escalograma (transformada de wavelet). 86
Tabela 19 - Tipos de redes neurais utilizadas como gene no algoritmo genético. 89
Tabela 20 Parâmetros utilizados para formação dos diferentes genes. 89

Tabela 21 - Parâmetros utilizados na configuração do algoritmo genético. 89
Tabela 22 - Parâmetros dos melhores indivíduos encontrados na otimização genética para ambos os conjuntos de dados 90
Tabela 23 - Parâmetros utilizados para a otimização utilizando a biblioteca Optuna. 91
Tabela 24 - Resultado de acurácia da rede convolucional quando avaliada utilizando a técnica de transfer learning, com entrada como os escalogramas, para os diferentes pontos de medida do acelerômetro. 93
Tabela 25 – Acurácia das redes para entrada de sinais superpostos na condição A e B com duração de 1 segundo mostrados na Figura 41. 97
Tabela 26 - Acurácia encontrada com as redes convolucionais com treinamento de sinais menores. 99
Tabela 27 - Acurácia encontrada com as redes totalmente conectadas com treinamento de sinais menores. 99
Tabela 28 - Porcentagem de acertos da rede convolucional quando avaliadas com sinais superpostos. 100
Tabela 29 - Porcentagem de acertos da rede totalmente conectada quando avaliadas com sinais superpostos. 100

Sumário

1 INTRODUÇÃO 15

1.1 Contextualização 19

1.2 Objetivo e Visão Geral da Tese 20

2 REVISÃO DA LITERATURA 22

2.1 Identificação e classificação de sinais 22

2.2 Keyloggers e identificação de texto 23

2.3 Identificação de sinais e redes neurais 26

3 METODOLOGIA DE MEDIÇÃO 30

3.1 Setup e procedimento de medição 30

3.2 Caracterização do setup de aquisição 36

3.3 Dados no tempo e análises em frequência 41

3.4 Extração de detalhes dos sinais 47

4 PROCESSAMENTO DE SINAIS USANDO REDES NEURAIS 52

4.1 Redes Neurais Totalmente Conectadas 54

4.1.1 Dados no tempo 54

4.1.2 Dados no Domínio da Frequência 61

4.1.3 Espectrogramas 65

4.1.4 Extração de detalhes 68

4.2 Redes Neurais LSTM 73

4.3 Redes Neurais Convolucionais 76

4.3.1 Redes convolucionais e imagens de espectrogramas 79

4.3.2 Redes convolucionais e imagens da transformada de wavelet 83

4.3.3 Transferência de aprendizado 84

4.3.4 Otimização dos parâmetros de treinamento com algoritmo genético 87

4.3.5 Otimização dos parâmetros da rede com Optuna 91
4.3.6 Avaliação da influência da posição de medida 92
5 SUPERPOSIÇÃO DE SINAIS 94
5.1 Superposição de sinais simples 94
5.2 Limitação do tamanho temporal de dados para treinamento 97
5.3 Identificação de senhas 101
5.4 Resumo de Resultados 104
6 CONCLUSÃO 106
6.1 Trabalhos futuros 107

1 INTRODUÇÃO

A identificação de sinais de vibração é uma tarefa essencial em muitas aplicações industriais, acadêmicas e cotidianas. Vários trabalhos são desenvolvidos anualmente visando melhorar a coleta e análise destes sinais, para otimização de processos e produtos e para monitorar e controlar fontes potencialmente danosas de vibração (ZHU & HUANG 2019) (LAARADJ 2023). É possível citar a indústria de petróleo e gás, que utiliza sensores de vibração em equipamentos críticos, como turbinas e compressores, para monitorar seu desempenho e integridade (DJAIDIR 2017). A falha desses equipamentos pode resultar em acidentes graves, além de causar grandes prejuízos financeiros para estas indústrias. Além disso, essa mesma indústria também utiliza a análise deste tipo de sinais para prospectar novos poços e zonas de perfuração (COELHO 2019), o que, se feito de forma errônea, pode gerar custos altos e desnecessários de tempo e dinheiro.

Outra aplicação importante de identificação de sinais de vibração é na indústria automotiva, onde a análise de vibrações é utilizada para avaliar o desempenho e a durabilidade, e para realização de manutenção preventiva em componentes, como motores e suspensões (CHIN 2022). A identificação de vibrações anormais pode indicar a necessidade de manutenção preventiva ou a substituição de peças defeituosas. Já no nosso cotidiano, nossos celulares são munidos de acelerômetros, microfones e outros sensores, que monitoram sinais de vibração do corpo para inferir informações sobre nossas atividades diariamente e nos dar importantes informações sobre nossos hábitos de atividades físicas, sono e saúde em geral (MAJUMDER & DEEN 2019).

No entanto, a análise de sinais de vibração pode ser uma tarefa desafiadora, uma vez que os sinais podem ser afetados por ruídos e interferências, tornando difícil a identificação de padrões relevantes. Como mencionado, a indústria petrolífera utiliza a análise de sinais de vibração para prospecção de possíveis novos poços, usando por exemplo a análise de gráficos da resposta de reflexão sísmica do solo, como mostrado na Figura 1 (COELHO 2019), onde as ondulações presentes contêm informações sobre o solo abaixo, discretizando tempo e as posições de medidas (CMP, *Common Midpoint*). Como pode-se perceber, essa análise é muito difícil, dadas a natureza e a baixa relação sinal-ruido do sinal. Para essa análise de viabilidade de perfuração, especialistas com dezenas de anos de experiencia são necessários para examinar

os resultados e dar o veredito sobre a possibilidade de sucesso da empreitada. Claramente, essa classificação pode ser imprecisa e subjetiva, uma vez que não é possível quantificar com clareza as possibilidades de sucesso. Um método matemático para análise desses dados pode tornar a reposta mais robusta e assertiva, e diminuir o tempo requerido para se tomar uma decisão.

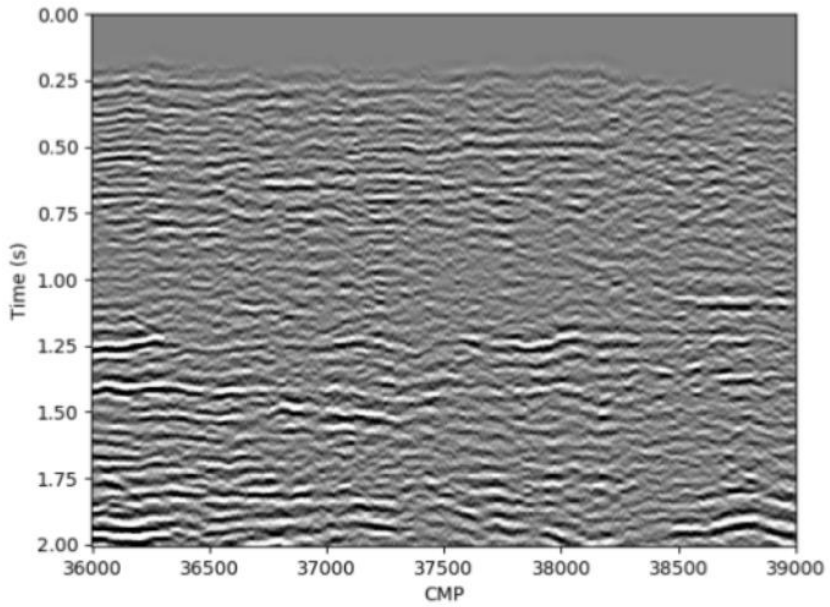

Figura 1 - Imagem da reflexão sísmica resultado de prospecção de solo. O eixo vertical se refere ao tempo viajado pelas ondas, e o eixo horizontal, CMP (*Common Midpoint*), se refere as diferentes posições de medida do experimento. (Fonte: COELHO 2019)

Nesse sentido, as redes neurais têm se mostrado uma ferramenta promissora para a identificação de sinais de vibração, devido à sua capacidade de aprender a partir de exemplos e reconhecer padrões complexos em dados (NIELSEN 2019). As redes neurais são modelos matemáticos inspirados no funcionamento do cérebro humano, compostos por camadas de neurônios interconectados. Esses modelos são capazes de aprender a partir de exemplos, ajustando os pesos das conexões entre os neurônios para minimizar o erro na predição de saídas esperadas. A partir do treinamento desses modelos, com uma base de dados conhecidos, eles são capazes de identificar os mesmos padrões em novos dados e classificá-los como o treinamento. No contexto da identificação de sinais de vibração e sonoros, as redes neurais podem ser treinadas a partir de exemplos de sinais normais e anormais (LEE 2018), para que possam reconhecer padrões relevantes, por exemplo.

O objetivo desse trabalho é entender como redes neurais podem ser usadas para melhorar a compreensão de sinais ruidosos. Utilizaremos como caso de estudo a classificação dos sinais de vibrações advindos da digitação de teclas em um teclado de computador, que serão medidos na mesa em que o teclado está fixado. Assim como os exemplos mostrados acima, neste o sinal medido também possui baixa relação sinal-ruido, uma vez que a energia gerada pela digitação deve se propagar em forma de onda mecânica pela estrutura do teclado e então chegar à mesa, onde será medida pelo acelerômetro. Esse problema foi escolhido por sua complexidade e grau de relevância para o campo de segurança de dados, que está cada dia mais em evidência. Além disso, este problema representa um desafio na identificação de sinais vibratórios com baixa relação sinal-ruído.

Além das redes neurais, o pré-processamento dos dados é um aspecto fundamental para a identificação de sinais de vibração. Diversas técnicas de pré-processamento podem ser aplicadas para remover ruídos, reduzir a dimensionalidade dos dados e melhorar a qualidade dos sinais, além da análise em frequência e as diferentes transformadas que podemos aplicar nos sinais temporais para estudá-los de outros pontos de vista (ALTES 1998). Neste trabalho, utilizaremos diferentes técnicas de análise de dados como aplicação de filtros, transformada de Fourier, espectrogramas, transformada contínua wavelet e extração de detalhes utilizando transformada discreta wavelet, a fim de se avaliar qual desses métodos, combinado com as redes neurais, nos resulta em maior acurácia na identificação do sinal adquirido. Essa transformada, em sua forma contínua e discreta, será mais bem descrita nas seções 3.3 e 3.4.

Vale ressaltar, que, se bem-sucedida, a identificação de teclas digitadas utilizando os sinais de vibração como mostrado em MARQUARDT 2011 ou sonoro como mostrado em HARRISON 2023, podem gerar quebras de segurança graves, uma vez que senhas e outras informações sensíveis podem ser roubadas através desse método, e de forma mais discreta, uma vez que acelerômetros já estão presentes nos aparelhos eletrônicos que usamos diariamente. Além disso, sistemas de medida de aceleração são compactos e podem ser facilmente instalados em dispositivos públicos como computadores e caixas eletrônicos, e, uma vez que não é necessário acesso ao sistema em questão, como no caso da instalação de *softwares* de *keyloggers*, esta ameaça pode estar presente sem uma ação direta do usuário do equipamento que está sendo monitorado (BERGER 2006 e KELLY 2010).

Como esperado, quanto maior a acurácia de uma rede neural, maior o número de sinais de teclas ela classificará corretamente, e maior a probabilidade de que uma sequência de teclas

digitadas seja decodificada. O caso de sequências é dificultado pelo fato de que vários sinais devem ser classificados corretamente para a sequência seja descoberta. Isso faz com que cada ganho de acurácia na rede seja importante, principalmente em sequências cada vez mais longas, onde a probabilidade do acerto é elevada ao número de dígitos da sequência.

Como exercício, podemos aproximar a probabilidade de decodificar uma senha, como a multiplicação da probabilidade de identificação de cada um dos caracteres que a compõe. Sendo assim, a dada uma senha com n caracteres, e uma rede neural que identifica a tecla digitada corretamente com acurácia A, a probabilidade de identificarmos todos os caracteres corretamente é dada por A^n. A Tabela 1, mostra alguns resultados para esse exercício, considerando como acurácia da rede neural valores entre 50 e 95% e senhas de 4 ou 8 caracteres.

Tabela 1 - Probabilidade de decodificação de sequências de caracteres com 4 e 8 dígitos, com diferentes acurácias de redes neurais para teclas individuais.

Acurácia da rede	**Sequencias de:**	
	$\boldsymbol{n = 4}$ **caracteres**	$\boldsymbol{n = 8}$ **caracteres**
50%	6,2%	0,3%
60%	12,9%	1,6%
70%	24,0%	5,7%
80%	40.9%	16.7%
90%	65,6%	43.0%
95%	81,4%	66,3%

Fica claro pela Tabela 1, que um pequeno aumento de acurácia, por exemplo de 90 para 95%, já traz mudanças significativas de probabilidade de decifrar sequências, passando-se de 43 para 66%, e isso corrobora nossa busca pela rede com maior acurácia possível. Vale ressaltar que sequencias de 4 e 8 dígitos são geralmente utilizadas para senhas de bancos e celulares. Por ser uma senha, não é possível garantir que os dígitos pressionados formem uma palavra ou sequência lógica, impossibilitando o uso de avaliadores semânticos MARQUARDT 2011 e outras técnicas parecidas, e fazendo com que a rede seja a única responsável pela saída correta da decodificação.

1.1 Contextualização

Este trabalho faz parte de uma iniciativa para se estudar a viabilidade da quebra de senhas utilizando sensoriamento não invasivo remoto. O objetivo desta iniciativa é estudar e desenvolver diferentes aspectos relacionados a quebra de senhas e identificação de sinais, sendo eles: aquisição remota de dados e transmissão para processamento na nuvem, mostrados por RIBEIRO 2022; identificação do instante de digitação (identificação de picos), mostrado em MESSIAS 2023; e identificação de teclas digitadas, mostrada neste trabalho e em MESSIAS 2023, utilizando redes neurais, e em GAIOTTO 2022, utilizando a estatística da latência entre digitações.

Todos os trabalhos mencionados vão ao encontro do diagrama da Figura 2, que ilustra a proposta de passo-a-passo de um algoritmo para a quebra de senhas. O sinal adquirido continuamente é subdivido em amostras quando há a identificação de uma digitação pelo algoritmo. Após essa identificação, o sinal é detalhado utilizando técnicas de processamento de sinais, gerando os dados necessários para a entrada da rede neural. A rede neural então recebe os dados e faz a classificação da resposta mais provável para aquele sinal. A classificação da rede neural pode conter também, por exemplo, as duas respostas mais prováveis, ou seja, as duas letras com maior probabilidade no vetor de saída da rede, para que outros algoritmos e análises, como o mostrado em GAIOTTO 2022, façam a classificação final.

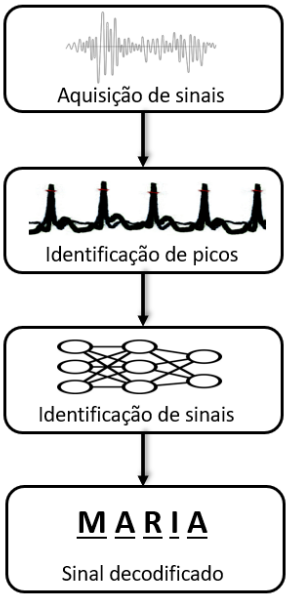

Figura 2 - Fluxograma de dados para o ataque e decodificação de senhas.

1.2 Objetivo e Visão Geral da Tese

Apesar da importância deste trabalho para a segurança de dispositivos eletrônicos, os métodos e estratégias aqui empregados podem ser expandidos para outros casos com o mesmo problema intrínseco de baixa relação sinal-ruido. Como exemplo podemos citar WIDODO 2016, onde as análises das imagens de um radar de penetração do solo são utilizadas para identificar corpos enterrados, o que poderia ajudar as equipes de busca e resgate em uma situação de soterramento de vítimas. Novamente esse problema lida com sinais de baixa relação sinal-ruido e com imagens difíceis de serem interpretadas, e a utilização de redes neurais e métodos matemáticos pode resultar em mais rapidez e acurácia na análise dos dados e ganhar minutos preciosos nessas situações.

Outro exemplo de empregabilidade deste tema é a identificação de distúrbios mecânicos em grandes infraestruturas de pesquisa. O Sirius (SIRIUS 2022) é a maior e mais complexa infraestrutura de pesquisa do Brasil. Nele são estudados materiais, medicamentos, e tecnologias fundamentais para o desenvolvimento de novos produtos e processos. Sendo um grande microscópio de raio-X, desde a construção do prédio, até o projeto dos componentes de pesquisa é feito para analisar detalhes nanométricos das amostras. Sendo assim, vibrações e distúrbios são vilões diários da operação desses experimentos e, um método automatizado, eficaz e veloz para identificação da fonte desses distúrbios poderia ajudar no sentido de aumentar a qualidade dos dados adquiridos (BUENO 2020).

Em suma, este trabalho tem como objetivo explorar diferentes abordagens de pré-processamento e redes neurais para a identificação de sinais de vibração. Utilizou-se como estudo de caso, teclas digitadas em um teclado, usando para isso somente a interpretação dos sinais vibratórios gerados por elas. A expectativa é que os resultados obtidos possam contribuir para o desenvolvimento de técnicas mais eficientes e precisas de identificação de sinais, resultando em melhorias significativas na manutenção preditiva, no controle de qualidade, segurança da informação, salvamento de vítimas, otimização de experimentos físicos e na detecção de falhas em aplicações industriais.

Na próxima seção, faremos uma revisão de literatura sobre os temas aqui abordados, mostrando com mais detalhes trabalhos semelhantes e pontos positivos e negativos de cada análise. Referências sobre estudos de redes neurais, processamento e análise de sinais, problemas de baixa relação sinal-ruido e identificação de textos serão oferecidas para o leitor

se aprofundar nos temas. Na Seção 3, o sistema de medida será detalhadamente descrito e os conjuntos de dados apresentado. Também serão discutidos os métodos matemáticos e os parâmetros para cálculo da transformada de Fourier, espectrogramas e transformada wavelet continua e discreta. Na Seção 4 serão explicados os tipos de redes neurais utilizados e mostrados os resultados para uma série de experimentos visando a identificação dos sinais com mais acurácia. Para os melhores resultados, otimizações utilizando algoritmo genético e a biblioteca Optuna são apresentadas. A Seção 5 tratará sobre o caso com maior dificuldade de identificação, quando os sinais de duas ou mais teclas estão superpostos no sinal adquirido pelo sensor. Ao final, é feito um resumo sobre os resultados encontrados e uma breve discussão sobre o tema geral. Na Seção 6, faremos um resumo sobre os dados obtidos e as principais informações encontradas durante o trabalho. Também serão apresentadas algumas possíveis continuações de estudo sobre este tema.

2 REVISÃO DA LITERATURA

2.1 Identificação e classificação de sinais

A classificação de sinais com baixo relação sinal-ruido é um desafio encontrado em muitas áreas de pesquisa. Em WIDODO 2016, encontra-se o estudo de caso da reflexão de ondas de um radar de penetração do solo, quando estas ondas são refletidas por corpos enterrados, visando criar um método para resposta mais rápida a vítimas de soterramento. O estudo analisa a viabilidade da detecção destes corpos enterrados através da análise do perfil das imagens geradas pelo radar e os resultados mostram uma correlação entre anormalidades no perfil das imagens de radar e a presença de corpos e possibilidade de distinguir entre corpos enterrados recentemente e enterros mais antigos.

Já em MOHAMED 2018, o autor tenta fazer a classificação de sinais eletromagnéticos de radares de comunicação para melhorar o alcance de um rádio de comunicação. Dadas a distância e a quantidade de sinais presentes no ambiente, a magnitude do sinal de interesse pode rapidamente se tornar pequena em relação a todos os outros distúrbios. A classificação do sinal correto ajuda o equipamento a se adaptar às interferências de rádiofrequência do ambiente e melhorar a comunicação entre equipamentos. O autor simula diferentes casos de comunicação e distúrbios em Wi-Fi (Wireless Fidelity) e LTE (Long Term Evolution), e utiliza redes neurais para fazer a distinção entre sinais, chegando a acurácias de mais de 90% na maioria dos casos.

Sobre a identificação de sinais utilizando técnicas de análise em frequência, podemos citar ALTES 1998, que faz uma análise sobre a identificação de sinais utilizando correlação de espectrogramas. No trabalho o autor propõe que a correlação entre espectrogramas de sinais medidos e de sinais de referência pode ser usada para definir parâmetros de similaridade do sinal, e então, fazer a classificação e/ou detecção do sinal medido. O autor ainda pontua que a deficiência deste método é a adição de ruído ao sinal medido, que pode fazer com que o espectrograma seja dependente das janelas e funções utilizadas para seu cálculo.

Outro caso de classificação utilizando imagens de espectrogramas é apresentado em LEVY 2022, onde imagens de espectrogramas de sinais 1D de áudio de vozes são classificadas com relação ao sotaque dos narradores nos áudios do conjunto de dados. Os autores propõem um método utilizando a triangulação de Delaunay para planificar e medir a distorção dos

espectrogramas e então fazer a classificação dos sotaques através dos parâmetros de medida da distorção. A triangulação de Delaunay é uma técnica de triangulação que conecta um conjunto de pontos em um plano de tal forma que nenhum ponto está dentro do círculo circunscrito de qualquer triângulo da triangulação. Isso resulta em uma malha de triângulos que maximiza o menor ângulo de todos os ângulos dos triângulos na triangulação.

2.2 Keyloggers e identificação de texto

Já na área de identificação de teclas digitadas, vários trabalhos foram desenvolvidos ao longo dos últimos anos mostrando diferentes possibilidades de roubo de dados através de diferentes sinais emitidos pelos aparelhos. O interesse nessa área se iniciou há mais de 50 anos, quando pesquisadores do Laboratório Bell, do exército dos Estados Unidos, perceberam que seu discador ultrassecreto de encriptação, o Bell 131-B2 emitia picos de sinais eletromagnéticos quando as teclas eram pressionadas. Os pesquisadores também descobriram que esses picos de sinais poderiam ser utilizados pelo inimigo para recuperar até 75% das mensagens criptografadas no campo de batalha (MONACO 2018).

Ao longo dos anos esses ataques maliciosos evoluíram, tornando-se mais sofisticados e começaram a usar outros sinais provenientes de digitação. BERGER 2006 e KELLY 2010 utilizam o sinal acústico dos dígitos das diferentes teclas para fazer a identificação de texto. No primeiro, são utilizados principalmente o tamanho da palavra e correlação do sinal das diferentes letras dentro de uma palavra. O trabalho mostra que, utilizando este método, que não requer nenhum treinamento, consegue-se reconstruir boa parte das palavras digitas a partir do sinal acústico, principalmente utilizando-se da decodificação de palavras longas e palavras com letras repetidas, conseguindo uma acurácia de 90% para as palavras longas e com letras repetidas e 60% para as palavras curtas sem letras repetidas. Já KELLY 2010 utiliza a análise em frequência dos sinais acústicos para extrair padrões, e utiliza um método de agrupamento para fazer uma pré-classificação dos dados. Após essa clusterização, os dados então são submetidos a uma rotina onde eles repetidamente sofrem uma conferência gramatical, e a um algoritmo treinado para fazer a classificação. Com esse método, KELLY 2010 consegue recuperar mais de 80% do texto em análise.

BACKES 2010 também utiliza os dados acústicos provenientes do processo de uma impressora para identificar o conteúdo sendo impresso. Primeiramente, dados conhecidos são coletados para treinamento de um algoritmo baseado no reconhecimento de detalhes da análise em frequência e conhecimento linguístico. Os dados desconhecidos são então adquiridos e tratados na frequência com o mesmo método dos dados de treinamento, e então classificados em palavras pelo algoritmo treinado. A posteriori, uma revisão linguística é aplicada para verificar os dados classificados e correções são feitas baseadas na hipótese de um texto lógico. Os dados então corrigidos são utilizados para um retreinamento do algoritmo para uma melhor performance nos casos seguintes. Segundo o autor, esta metodologia atingiu uma acurácia de 72% para um caso genérico e até 95% se é feita uma hipótese sobre o assunto do texto.

XIMING 2019 utiliza os sinais acústicos no espectro do ultrassom para fazer um ataque malicioso a senhas PIN de smartphones. O trabalho propõe dois métodos, o primeiro baseado na identificação dos tempos entre as digitações e o segundo baseado no espectro de ultrassom proveniente da digitação. O primeiro método apresenta uma quebra mais rápida da senha do que o método de tentativas aleatórias, e segundo a autora o segundo método apresenta uma acurácia de 75% para decodificar senhas. A autora também marca que o advento da validação em dois passos (2FA) ajuda a mitigar o perigo de drivers maliciosos, desde que ao menos um dos dois passos não envolva digitação (biometria, reconhecimento facial e de íris).

Com a popularização de aparelhos eletrônicos como *smartphones* e *smartwatches*, os sensores presentes nesses aparelhos a cada versão têm se tornado mais potentes e precisos. MARQUARDT 2011, MAITI 2016 e CAI 2011 se aproveitaram dos sensores já embutidos nesses aparelhos para coletar dados de vibração e movimento e realizar a decodificação de textos. No caso de MARQUARDT 2011, ele utiliza o acelerômetro embutido em um iPhone como sensor para decodificar textos digitados em um teclado próximo. O autor propõe um método de decodificação onde as teclas digitadas são separadas em conjunto de duas e são tratadas por três classificadores: classificador que determina se a primeira tecla foi digitada do lado esquerdo (E) ou direito (D) do teclado, outro que classifica o lado da segunda tecla e outro que classifica se as teclas estão próximas (P) ou distantes (L) umas das outras. Sendo assim cada conjunto de suas teclas é classificado por três letras (lado da primeira tecla, lado da segunda tecla e distância entre teclas) que correspondem às propriedades acima. O autor mostra que esta classificação filtra as possíveis palavras digitadas e utilizando uma probabilidade linguística consegue decifrar uma parte do texto. O autor reporta a acurácia em número das palavras mais

possíveis para aquela codificação, sendo que em 40% dos casos a palavra digitada está entre as dez primeiras palavras sugeridas pelo modelo, e até 80% no está entre as duzentas primeiras palavras sugeridas pelo modelo.

MAITI 2016 utiliza o sensor presente em um smartwatch para decodificar o texto digitado pelo usuário em um teclado QWERTY. O autor faz a aquisição do sensor de movimento do smartwatch e utiliza padrões de movimento do pulso e compara com um dicionário de movimentos similar ao proposto por MARQUARDT 2011. A partir do dicionário, as palavras mais prováveis são escolhidas e o texto é decodificado, chegando a uma acurácia próxima do apresentado por MARQUARDT 2011 e BERGER 2006, como comparado pelo autor na Figura 3.

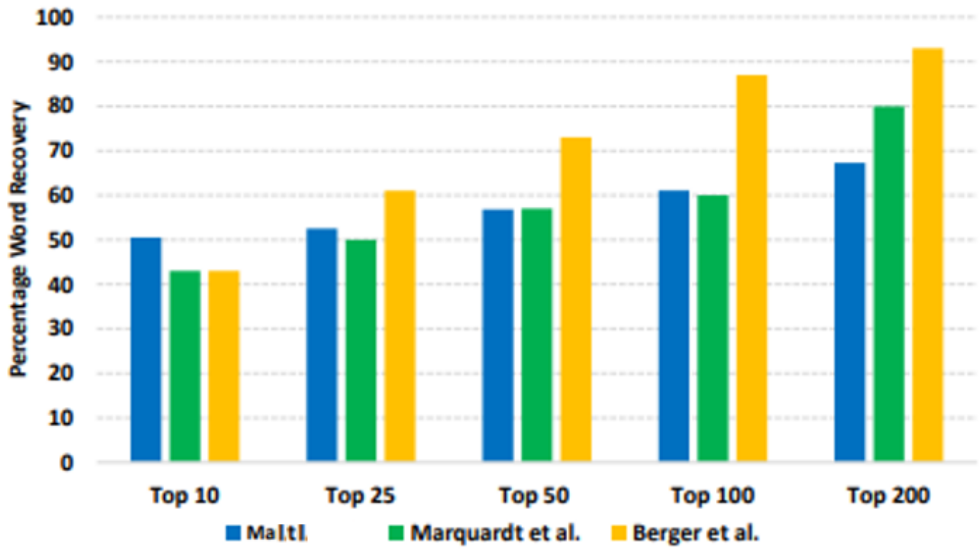

Figura 3 - Compração entre a porcentagem de recuperação de palavras em MAITI 2016, MARQUARDT 2011 e BERGER 2006. (Fonte: MAITI 2022)

CAI 2011 utilizam os sensores embutidos em *smartphones* para adquirir dados de vibração e giroscópio e fazer a classificação do toque. A autora defende que a inserção da tecnologia de *touchscreen* nos *smartphones* dificultou o uso dos dados de aceleração e movimento para reconhecimento de dígitos. Mesmo assim, essa tecnologia não é imune a ataques maliciosos. O algoritmo proposto utiliza os diferentes padrões de ângulos gerados quando diferentes partes de uma tela touchscreen são tocadas e foca na identificação dos

números do teclado numérico. Segundo reportado pela autora, essa metodologia consegue recuperar com até 70% de exatidão as teclas utilizadas.

2.3 Identificação de sinais e redes neurais

A ideia básica do funcionamento de redes neurais está proposta desde a década de 50, quando o psicólogo Frank Rosenblatt (ROSENBLATT 1962) propôs e construiu um enorme equipamento chamado *Perceptron*. Dada a limitada tecnologia de processamento da época, o *Perceptron* era uma réplica física das ligações que existem no cérebro humano, sendo formado por inúmeros pequenos motores que ajustavam os ganhos de pequenos resistores variáveis. Ao aprender, o *Perceptron* ajustava a posição dos motores de forma que a saída analógica era a correta dada a entrada proposta. O *Perceptron* foi capaz de aprender a reconhecer diferentes letas e a diferenciar homens e mulheres em pequenas fotos da época. (SCHUCHMANN 2019)

Com o avanço da tecnologia de processadores e computadores, nos últimos anos o tema de redes neurais voltou a receber muita atenção, já que as capacidades de processamento atuais são apts de realizar os inúmeros cálculos necessários para o treinamento das redes neurais. Sua utilização em elementos de texto, imagem, vídeo e sinais vem crescendo e se popularizando a cada ano, ao mesmo tempo em que aumenta a necessidade de cada vez mais potentes sistemas de processamento e a necessidade de cada vez mais extensas bases de dados. Desde o cotidiano, como reconhecimento de atividades físicas em celulares até mais complexas tarefas de processamento de imagem, as redes neurais nunca foram tão empregadas como hoje em dia. A seguir, estão mostrados alguns trabalhos que, assim como esse, tentam utilizar essa ferramenta para reconhecimento e classificação de sinais temporais.

LEE 2018 utiliza uma rede neural convolucional (que será explicada em detalhes na seção 4.3), para classificar espectrogramas dos sinais de potência de sistemas embarcados em três principais tipos: normal, ruido AC e ruido impulsivo, como mostrado na Figura 4. O autor se vale do bom detalhamento que a análise em frequência de espectrogramas resulta e da boa performance das redes neurais convolucionais em tratar imagens para desenvolver um método com acurácia de 99.52%.

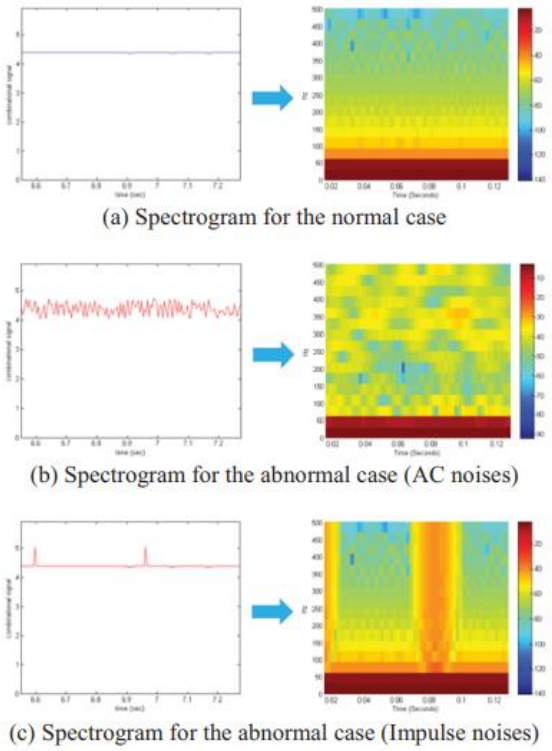

Figura 4 - Comparação entre espectrogramas normais e ruidosos. (Fonte: LEE 2018)

Já em OZTURK 2020, o autor utiliza imagens de espectrograma em combinação também com redes neurais convolucionais para identificar diferentes tipos de veículos aéreos não tripulados (UAVs) através do sinal de rádiofrequência deles. Estes sinais, dada as interferências do ambiente, são de baixa relação sinal-ruido, trazendo um desafio para esse problema, e fazendo com que o autor tenha que focar suas análises em uma banda específica do espectro. O trabalho mostra que a utilização das redes neurais convolucionais para esse destino foi um sucesso, uma vez que a acurácia de reconhecimento destes sinais ficou acima de 90% para todos os casos, e chegando até a 100% em alguns casos.

A indústria do petróleo também tem se utilizado das técnicas de aprendizado de máquinas como redes neurais para otimizar os diferentes processos na cadeia de extração e produção. Em COELHO 2019, a autora mostra a utilização de redes neurais convolucionais para auxiliar no processo de prospecção de novos poços. Os espectrogramas gerados pela prospecção do solo em locais de possíveis novos poços são geralmente analisados por especialistas com décadas de experiência, elevando os custos e o tempo para prosseguimento do trabalho. A autora propõe e treina uma rede neural convolucional baseada na rede *LeNet5* com as imagens analisadas por esses especialistas e os pontos de importâncias destas para que a rede, por si só, seja capaz de

encontrar os detalhes buscados. A autora mostra sucesso nos testes, com acurácias superiores a 90% para os dados de treinamento e superior a 80% para os dados de teste.

Sendo uma poderosa ferramenta para classificação de sinais, redes neurais já têm sido utilizadas para fazer a classificação de sinais de teclas digitadas. Como exemplo, podemos citar NARAIN 2014, onde o sinal acústico e de giroscópio de *smartphones* são utilizados em combinação em uma série de algoritmos de aprendizado de máquinas para fazer a identificação da tecla digitada. Segundo os testes apresentado no próprio trabalho, o método proposto tem acurácia de mais de 90%, e, como está identificando teclas separadas, também é independente de algoritmos semânticos de identificação e correção de textos, sendo um grande avanço para a área. O autor também pontua que a utilização dos dados de microfone é mais eficiente que a utilização dos dados do giroscópio. Entretanto, a utilização de ambos em conjunto potencializa o aprendizado do algoritmo, chegando a acurácias de até 94%.

LOY 2007 utiliza um tipo especial de rede neural, a ARTMAP-FD, para fazer a classificação de dígitos utilizando a pressão exercida pelo operador em um teclado e a latência entre a digitação das teclas. As redes ARTMAP-FD (*Fuzzy ARTMAP with Distributed Output Code*) são uma extensão das redes Fuzzy ARTMAP, projetadas para resolver problemas de classificação em ambientes de aprendizagem supervisionada. Elas utilizam códigos de saída distribuídos para melhorar a capacidade de generalização e reduzir a sensibilidade a ruídos. A principal característica do ARTMAP-FD é a sua capacidade de lidar eficientemente com dados de entrada ambíguos ou ruidosos, mantendo a precisão da classificação e a estabilidade da rede durante o processo de aprendizado incremental.

O autor compara a acurácia deste tipo de rede com outros tipos de rede largamente utilizados, mostrando a paridade entre essas ferramentas. O autor também demonstra a utilização dos padrões de pressão para reconhecimento do usuário que está operando o teclado, sendo possível distinguir entre diferentes pessoas digitando. Utilizando os dados de pressão e latência de digitação, o autor reporta uma acurácia de 80%.

Estudos recentes, incluindo UPRETEE 2021, PATTNAIK 2016, DI NARDO 2022 e AZIZ 2021, têm demonstrado avanços significativos na classificação de sinais biológicos, empregando técnicas sofisticadas de extração de características. UPRETEE 2021 destaca o uso de sinais acústicos cardíacos para a detecção de patologias cardiovasculares, evidenciando que a precisão do método depende significativamente da experiência dos operadores. O autor sugere o emprego de técnicas de extração de características espectrais para treinar classificadores

capazes de diagnosticar condições cardíacas, apresentando resultados promissores com a extração de uma única característica espectral.

PATTNAIK 2016 aborda a identificação de padrões em sinais de eletroencefalograma (EEG), visando a interação cérebro-computador. Utilizando a transformada wavelet discreta (seção 3.4) para extrair informações do EEG, os autores desenvolveram um modelo baseado em rede neural artificial, alcançando uma precisão de cerca de 80% na diferenciação entre imaginação de movimentos da mão direita ou esquerda, com base em métricas como média, desvio padrão e potência de pico.

No caso de DI NARDO 2022, o autor investiga a análise de sinais de eletromiografia de superfície através da transformada wavelet contínua, destacando a ausência de um padrão de referência para a detecção precisa de ativação e desativação muscular. O estudo propõe uma rede neural de camada única totalmente conectada, comparando sua eficácia com outras metodologias e demonstrando uma acurácia superior a 90%.

Já AZIZ 2021 introduz uma metodologia inovadora para a análise de sinais de eletrocardiograma (ECG), combinando médias móveis e a transformada fracionária de Fourier para aprimorar a detecção de características cardíacas. O uso de um amplo banco de dados hospitalar com mais de 10.000 pacientes possibilitou treinar um modelo de aprendizado de máquina eficaz na classificação de doenças cardíacas, superando o desempenho de algoritmos preexistentes.

Por fim, ASIF 2023 explora o reconhecimento biométrico por ECG, propondo uma metodologia que integra a transformada discreta wavelet e uma rede neural convolucional recorrente unidimensional para melhorar a precisão do reconhecimento. Esta abordagem demonstrou acurácias significativas em diversos conjuntos de dados, ressaltando seu potencial para aplicações em segurança e saúde, particularmente na identificação biométrica.

3 METODOLOGIA DE MEDIÇÃO

A presente seção tem por objetivo mostrar a metodologia de medição utilizada ao longo dos estudos de redes neurais do projeto. Os conjuntos de medidas aqui adquiridos servirão para avaliação e treinamento dos diferentes tipos de redes neurais. Além disso, esta seção mostra uma caracterização dinâmica do setup de medida, através da análise modal da mesa utilizada para as medidas. Também serão mostrados os conceitos por trás dos processamentos de sinais empregados aos dados medidos. Esses processamentos, como transformadas e análises em frequência, serão posteriormente utilizados como dados de entradas das redes neurais, como a transformada de Fourier, espectrograma e transformada wavelet continua e discreta.

3.1 Setup e procedimento de medição

Para medidas e redes estudadas, buscou-se a implementação de uma configuração de medida simples e enxuta, de forma a minimizar todas as possíveis entradas de distúrbios no sistema. A configuração de medida utilizada inicialmente é mostrada na Figura 5, e conta com uma mesa de escritório, um teclado Dell e um acelerômetro Kistler, além dos elementos de aquisição não mostrados na imagem, sendo eles: um conversor analógico-digital e um computador. Mais informações sobre esses itens serão apresentadas a seguir.

Figura 5 - Configuração para medida inicial com acelerômetro na posição I.

Visto que nesse ponto do estudo ainda não era possível avaliar a dificuldade do problema, optou-se por retirar alguns fatores de complexidade do sistema para estas medidas. A mesa utilizada não possuía outros equipamentos ou componentes sobre ela, uma vez que as ondas mecânicas se propagando pela mesa, devido aos impactos nas teclas, poderiam refletir em interfaces ou excitar essas massas adicionais, propagando de volta outras ondas específicas para aquela configuração. Além disso, a movimentação ou modificação de algum dos componentes sobre a mesa tornaria impossível a repetibilidade do teste e aquisição de dados.

Outro ponto de atenção: a mesa utilizada foi isolada das mesas adjacentes e cuidadosamente nivelada com o piso. A etapa de isolamento da mesa com as mesas adjacentes visa novamente a fácil repetibilidade dos testes, já que a aquisição de novas teclas, e possivelmente novos pontos de aquisição, poderia acontecer ao longo do estudo, dado que o número de dados necessários para cada tecla ainda não era conhecido nesse momento. Já o nivelamento da mesa com o piso é importante para que os dados de vibração não sejam afetados por uma oscilação da mesa em si, além de também garantir a fácil repetibilidade dos testes.

Escolheu-se utilizar o teclado DELL sem fio modelo KM636, com medidas de 441 mm x 128 mm x 29 mm e 549g. O modelo sem fio foi escolhido para que as ondas propagadas e medidas pelo acelerômetro fossem apenas transmitidas pelos apoios do teclado. A presença do cabo poderia dissipar energia das ondas das teclas, além de a posição do cabo poder modificar as ondas geradas, o que também dificultaria a reprodução dos resultados. Esse modelo de teclado possui 4 apoios, sendo um próximo a cada vértice, de material emborrachado. Cogitou-se a remoção dos apoios emborrachados para maximizar a energia das ondas transmitidas entre o teclado e a mesa, mas a remoção deles acarretaria instabilidade do equipamento, visto que os pontos de apoios não ficariam bem definidos pela geometria restante, adicionando um elemento não repetitivo e não controlável nas medições. Assim, mantiveram-se os suportes emborrachados, mesmo sabendo-se do maior amortecimento a que o sistema estaria sujeito.

Para a eletrônica de medida, utilizou-se um acelerômetro triaxial Kistler modelo 8762A5 com sensibilidade de 102mV/(m/s²), e banda de medição de 0.5Hz a 6kHz. Mais especificações desse acelerômetro estão mostradas na Tabela 2. Esse sensor foi escolhido devido à boa relação sinal-ruido (ruido menor que $0.0003(m/s^2)^2/Hz$ para a banda de frequência de 1Hz a 2.5kHz) de sua medida e à pouca massa que ele adiciona ao sistema, minimizando-se a influência dele na resposta mecânica do sistema. Outros fatores para escolha desse modelo foram sua banda de frequência de medida e amplitude máxima de medida, que são compatíveis com as frequências

esperadas do conjunto. Além das características mostradas acima, o acelerômetro escolhido é de medida triaxial, ou seja, ele mede a aceleração em 3 eixos ortogonais. Isso é importante, uma vez que as respostas de vibração em diferentes direções podem resultar em dados com qualidades e informações diferentes para a classificação pela rede neural artificial. Assim, com essa medida em 3 direções, é possível avaliar qual a direção que maximiza a taxa de aprendizagem e acurácia da rede. O sistema de coordenadas utilizado está mostrado na Figura 6 e possui os seguintes eixos:

-Eixo X: alinhado com a direção comprimento do teclado e apontando para a direita;

-Eixo Y: ortogonal ao plano da mesa e apontando para baixo;

-Eixo Z: Alinhado com a largura do teclado;

Tabela 2- Especificações do acelerômetro Kistler triaxial 8762A5.

Parâmetro	**Valor**
Amplitude de medida	±49.05m/s²
Sensibilidade	102mV/(m/s²)
Resolução	0.0029m/s²
Banda de frequência	0.5 a 6000Hz
Massa	23g
Ressonância interna	30kHz
Corrente de excitação	>2mA

O conversor analógico-digital (AD) utilizado foi da marca *NI*, modelo NI USB-4431, com quatro entradas analógicas de 24bits e amplitude de ±10V. Mais informações sobre esse sistema estão mostradas na Tabela 3. Esse conversor foi escolhido por ter um ruído interno consideravelmente baixo (menor do que 75µVrms) comparado aos outros modelos disponíveis, pela boa discretização de amplitude dos sinais medidos, dado sua conversão com 24btis, e pela fácil comunicação e programação com os softwares disponíveis para o autor. O software utilizado para a aquisição de dados foi o *Signal Express* 2015, também da empresa *NI*. Esse software permite a configuração (taxa de aquisição, tempo de aquisição, filtros digitais) rápida de diferentes tipos de medida, bem como a visualização e salvamento dos dados. Os canais X, Y e Z do acelerômetro foram ligados respectivamente nos canais AI1, AI2 e AI3 do conversor analógico-digital.

Tabela 3 - Especificações do conversor analógico digital NI USB-4431.

Parâmetro	**Valor**
AD Resolução	24bits (1.2µV)
Taxa máxima	102.4kHz
Taxa mínima	1kHz
Amplitude	±10V
Ruido a 20kHz Max.	75µVrms
Ruido a 20kHz Tip.	55µVrms

Para as medidas iniciais foram escolhidos parâmetros de aquisição que proporcionassem flexibilidade para o tratamento de dados, não limitando frequências amostradas e o tamanho temporal do dado. Assim, foi usada uma taxa de aquisição de 10kHz (F_s), com tempo de medida de 2 minutos e sem aplicação de filtros analógicos ou digitais. Com esses parâmetros e utilizando a fórmula da frequência de Nyquist mostrada na Equação 1, a frequência máxima observável (F_{obs}) é de 5kHz.

Em todo conjunto de medição, outro ponto importante a ser analisado é o problema de *aliasing.* O problema de *aliasing* ocorre quando um sinal contínuo é amostrado a uma taxa insuficiente, resultando na sobreposição de frequências na representação digital. Isso acontece porque, de acordo com o teorema de Nyquist-Shannon, a frequência de amostragem deve ser pelo menos duas vezes a maior frequência presente no sinal. Se essa condição não for atendida, as diferentes frequências do sinal original se misturam, produzindo distorções e perda de informações na reconstrução do sinal, conhecido como *aliasing*. O conversor AD utilizado funciona de forma que os sinais são adquiridos na taxa de 102.4kHz para cada canal e posteriormente esses sinais são decrementados para a taxa requerida pelo usuário. Assim, o problema de *aliasing* será apenas para frequências acima de 102.4kHz, que podem ser desprezadas para esse sistema mecânico.

$$F_{obs} = F_s/2 \qquad \text{(eq. 1)}$$

Durante os 2 minutos de aquisição continua, as teclas de interesse eram repetidamente pressionadas pelo usuário, sendo que a cada medida de 2 minutos, apenas uma tecla era digitada.

Para garantir a integridade do sinal e minimizar distúrbios, adotou-se o cuidado de manter o ambiente de aquisição de dados livre de presenças de outras pessoas, além do próprio usuário, e de assegurar que não ocorressem outros processos que gerassem ruído, tanto vibratório quanto sonoro, nas salas adjacentes. Seguindo o protocolo estabelecido, a digitação foi realizada de maneira que o único contato do usuário com o sistema (composto por mesa e teclado) se limitasse às teclas acionadas. Quaisquer medições que registrassem interações acidentais do usuário com outras partes do sistema, como o tampo ou a base da mesa, foram integralmente descartadas e os dados readquiridos.

O usuário controlou a força de digitação de forma que essa se assemelhasse a um processo de digitação cotidiano. O espaçamento temporal entre impactos era tal que o sinal da tecla tinha tempo suficiente para decair a valores próximos ao ruido de fundo da medição. Esse espaçamento temporal foi empiricamente estimado em aproximadamente 2s. Assim, cada medida de 2 minutos possui aproximadamente 60 sinais de impactos nas teclas de interesse.

Para os testes que serão mostrados, nas seções a seguir, escolheu-se adquirir dois conjuntos distintos de dados: um composto por teclas próximas e outro composto por teclas distantes, destacados na Figura 6. A utilização do conjunto de dados de teclas distantes tem objetivo de avaliar o desempenho da rede em casos são considerados menos complexos, uma vez que a propagação das ondas tem um caminho bem diferente para cada uma das teclas. Já o conjunto de dados próximos, tem por objetivo avaliar a acurácia da rede para casos em que a diferença de sinal é mínima, levando ao limite a necessidade de uma boa aprendizagem de padrões.

Figura 6 - Conjuntos de teclas utilizadas para construção dos dados de treinamento das redes neurais artificiais. Conjunto de teclas distantes “1”, “P”, “Z” e “:” (destacados em azul). Conjunto de teclas próximas “A”, “S”, “Z” e “X” (destacados em verde).

A posição de fixação do acelerômetro foi variada entre as configurações I, II, III, IV, mostradas na Figura 7. A posição I foi escolhida de forma que não houvesse teclas com posições espelhadas com relação aos eixos, ou seja, nenhuma das teclas está a uma mesma distância em direção oposta de outra tecla em referência aos eixos de coordenadas. Isso foi pensado para que houvesse maiores diferenças entre as assinaturas dos sinais de cada tecla, sendo que os sinais de teclas espelhadas poderiam confundir o algoritmo de classificação. A posição II segue o mesmo racional da posição I, entretanto se encontra mais próxima ao conjunto de próximas (‘A’, ‘S’, ‘Z’ e ‘X’). A posição III é simétrica à posição I com relação ao eixo de simetria lateral do teclado, e a configuração IV está posicionada de forma que as teclas “1” e “P” têm a mesma distância do acelerômetro.

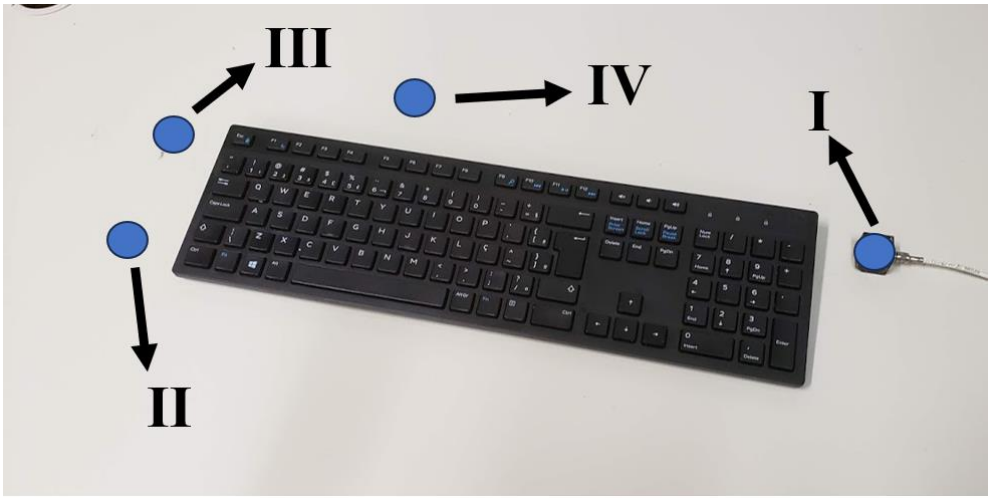

Figura 7 - Esquemático das posições I, II, III, IV medidas com o acelerômetro durante a digitação das teclas de interesse.

Para todas as configurações de posição do acelerômetro foram mantidas a mesma orientação do acelerômetro e os parâmetros de aquisição já mencionados. Além disso, para todas as configurações o acelerômetro foi posicionado próximo ao teclado para minimizar perdas por dissipação de energia na mesa e assim maximizar a relação sinal ruido. O sensor foi fixado com cera de abelha para uma melhor aderência na mesa até uma frequência de acima de 8kHz (OLOFSSON 1995) e o cabo foi fixado em dois pontos na mesa, para minimizar os distúrbios externos que poderiam se propagar de fora da mesa até o acelerômetro, e para diminuir possíveis efeitos de mudanças no acoplamento do acelerômetro com a mesa. A Tabela 4 relaciona cada tecla com o número de digitações que foi medido durante o processo de

aquisição de dados. Para cada uma das configurações de posição do acelerômetro ambos os conjuntos de dados foram medidos para cada uma das teclas de interesse.

Tabela 4 - Conjunto de dados de teclas próximas e distantes e número de impactos coletados para treinamento das redes neurais.

	Conjunto de teclas próximas				**Conjunto de teclas distantes**			
Teclas	“A”	“S”	“Z”	“X”	“1”	“P”	“Z”	“:”
Configuração I	62	67	68	62	59	60	68	63
Configuração II	59	63	61	65	53	61	59	63
Configuração III	63	60	62	62	64	66	62	63
Configuração IV	55	67	57	59	64	60	61	68

3.2 Caracterização do setup de aquisição

Na continuação dos experimentos descritos na subseção anterior, o mesmo acelerômetro foi empregado para a caracterização modal da mesa onde o teclado foi posicionado. A caracterização modal é importante para entender melhor o sistema da mesa onde o teclado foi posicionado. Com a caracterização modal, podemos estabelecer com precisão as frequências naturais do sistema e seus respectivos modos de vibrar, o que balizará algumas decisões nas próximas seções.

O acelerômetro foi estrategicamente posicionado seguindo um esquema de malha 3 x 3, o que resultou na divisão equitativa da superfície da mesa em áreas distintas. A disposição foi concebida para capturar uma representação abrangente dos modos de vibração da mesa, permitindo uma identificação precisa dos mesmos. A distribuição das posições utilizadas para o acelerômetro é ilustrada esquematicamente na Figura 8.

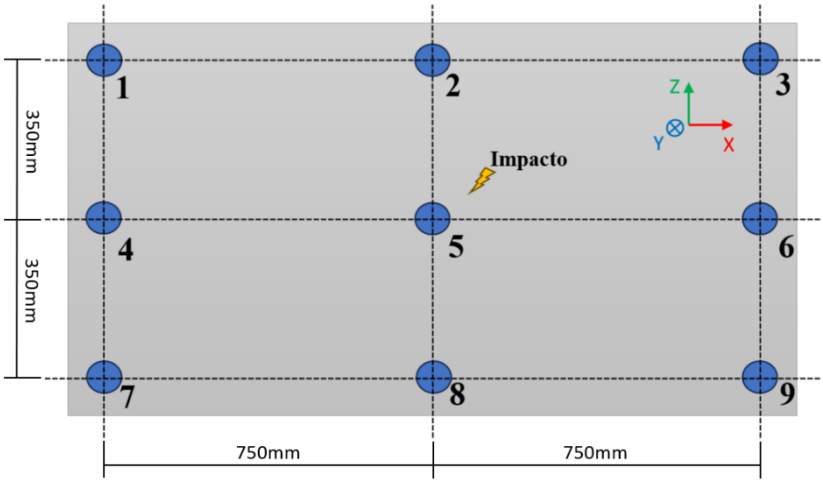

Figura 8 – Malha 3x3 utilizada para medida modal da mesa de escritório utilizada nos experimentos e localização do ponto de impacto.

Adicionalmente, foi utilizado um martelo de impacto Kistler 9722A500 para gerar e mensurar a força de excitação sobre o sistema. Este instrumento pode ser acoplado a diferentes tipos de pontas, adaptando-se assim às diversas gamas de frequência desejadas para a excitação. No teste específico mencionado, optou-se por uma ponta de nylon, que possibilita a excitação do sistema em altas frequências, alcançando excitações até por volta de 2kHz. Esta escolha foi baseada na análise dos sinais adquiridos, que revelou componentes significativos até pelo menos a frequência de 700 Hz. As características técnicas do martelo de impacto são sumarizadas na Tabela 5.

Tabela 5 - Tabela de parâmetros do martelo de impacto Kistler 9722A500.

Parâmetro	**Valor**
Sensibilidade	10 mV/N
Força	0 - 500N
Frequência máxima	8.3kHz
Frequência de ressonância	27kHz
Peso	100g

O procedimento experimental consistiu em posicionar o acelerômetro em cada um dos nove pontos previamente estabelecidos, fixando-o com cera de abelha para garantir estabilidade durante a coleta de dados. Realizaram-se quatro impactos com o martelo em um ponto fixo, próximo ao centro da mesa (ponto 5), orientados verticalmente ao longo do eixo Y do sistema de coordenadas. A escolha deste ponto visou replicar as condições de digitação dos testes anteriores, simulando a resposta da mesa a tais estímulos.

Utilizou-se o mesmo sistema de aquisição dos experimentos precedentes, configurado para uma taxa de amostragem de 10 kHz. Monitorou-se a força aplicada em cada impacto para evitar a ultrapassagem do limite de 500N do equipamento, o que poderia ocasionar perda de dados, comportamento não linear do sistema ou danos aos instrumentos. Todas as aquisições em que, porventura, a força de excitação passou o limite de 500N, foram descartadas e readquiridas.

Após a aquisição dos dados relativos aos quatro impactos em cada ponto, procedeu-se ao tratamento dos sinais para obter as funções de resposta em frequência (FRF) correspondentes. Através das FRFs, podemos identificar as frequências naturais do sistema e o modo de vibrar para cada uma delas. Inicialmente, os sinais foram seccionados em intervalos de 2 segundos baseando-se na observação de que as oscilações induzidas decrescem para níveis inferiores ao ruído de fundo após esse período. Foi assegurada a inclusão de 1500 (0.15s) amostras antes de cada impacto, para evitar problemas no janelamento dos picos, e evitar que parte do pico não seja levada em conta pelo corte do sinal no pico mais alto.

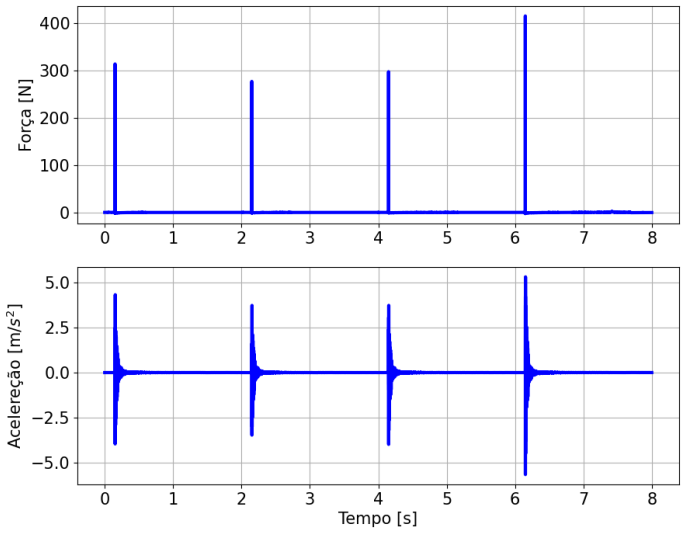

Figura 9 - Excitação e resposta do acelerômetro para o impacto na posição 1 da medida.

Posteriormente, os sinais de impacto e resposta para um mesmo ponto foram concatenados, formando um registro único de 8 segundos, conforme ilustrado na Figura 9. Sendo a função de transferência H_{n_a} entre o ponto de impacto i_n e o ponto medido m_{n_a}, onde o subíndice n denota o ponto de medida, como nomeado na Figura 8, e o subíndice a denota uma das três direções de medida do acelerômetro (X, Y ou Z), podemos estimar H_{n_a} como sendo:

$$H_{n_a} = \frac{P_{m_{n_a} i_n}}{P_{i_n}} \quad \text{(eq. 2)}$$

onde P_{i_n} representa a densidade espectral de potência (em inglês *PSD*, ou *power spectrum density*) do sinal de impacto do martelo quando o acelerômetro estava posicionado no ponto n. Já o termo $P_{m_{n_a} i_n}$ denota a densidade espectral de potência cruzada (em inglês C*SD*, ou cross-*power spectrum density*) entre o sinal de entrada do martelo e de resposta do acelerômetro para a direção a, quando este está fixado na posição n. Este método possui vantagem com relação a simples divisão da transformada de Fourier dos dois sinais, como mostrado em ALLEMANG 2001, pois assume que o ruído presente no sinal, não está correlacionado com o sinal de entrada.

O cálculo da densidade espectral de potência e da densidade espectral de potência cruzada foi feito utilizando respectivamente as funções em *python welch* e *csd*, da biblioteca *python scipy*. As funções mencionadas foram executadas com janelas de 2s, para que se fosse feita a média dos quatro impactos medidos. A Figura 10 mostra o módulo das funções de transferência H_{1_X}, H_{1_Y} e H_{1_Z}:

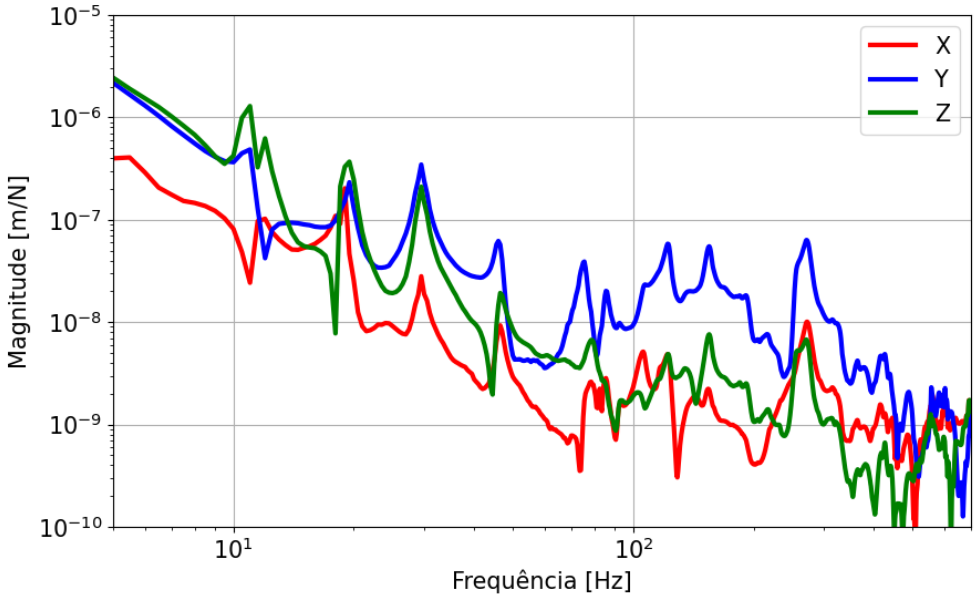

Figura 10 - Funções de resposta em frequência para o ponto de medida 1 nas três direções.

Após o cálculo de todas as funções H_{n_a}, podemos analisar os gráficos gerados e identificar frequências naturais do sistema. Após identificadas as frequências naturais, podemos fazer uma animação do modo de vibrar de cada frequência de interesse utilizando as informações de amplitude e frequência presentes na função de resposta em frequência. Infelizmente, as animações geradas, não podem ser mostradas neste texto estático, entretanto seus resultados são compilados na Tabela 6 abaixo, junto com as frequências naturais.

Tabela 6 - Frequências naturais do sistema até 100Hz e descrição dos modos de vibrar.

Frequência Natural	**Modo de vibrar**
12Hz	Rotação em Rx
20Hz	Translação em Z
29Hz	Translação em Y
45Hz	Rotação em Rz
74Hz	Flexão ao longo de X
83Hz	Rotação em Rz

Analisando os resultados, vemos conteúdo relevante até aproximadamente 300Hz e após isso há uma queda acentuada da amplitude da oscilação. Estes resultados contribuíram significativamente para o nosso entendimento das frequências que são excitadas por diferentes ações de digitação. Além disso, o discernimento acerca dos modos de vibração auxiliou na orientação de decisões na condução de testes e processamentos, com o objetivo de aprimorar a distinção entre os sinais de digitação registrados. Uma vantagem notável desse resultado é a sua aplicabilidade na validação de futuras coletas de dados. Isso deixa aberto, caso necessário não apenas a expansão, mas também o enriquecimento do banco de dados existente, assegurando uma base de dados mais robusta e representativa para análises subsequentes.

3.3 Dados no tempo e análises em frequência

O sinal de aceleração no tempo, aquirido da medida de uma das teclas está apresentado na Figura 11 a seguir, na forma de um gráfico com amplitude em [m/s²]. É possível claramente observar os impactos (picos de aceleração) consecutivos na tecla e a formato exponencial do sinal sendo dissipado na mesa para a direção Y. Já para a direção X e Z essas diferenças são mais sutis. Mas ainda assim podem ser observadas. Também é possível observar uma variação da magnitude do pico de aceleração entre impactos diferentes. Isso se deve a variações de pressão exercida pelo usuário na digitação de cada tecla. Vale ressaltar que, durante as medidas, apenas a tecla entra em contato com o operador. Medidas em que houve contato com a mesa ou outros distúrbios (pessoas na sala, máquinas) foram descartadas.

A Figura 12 mostra o espectrograma dos 5 primeiros impactos (10s) na tecla “1”, na direção Y, com o acelerômetro posicionado na configuração I, para os valores de frequência entre 1 e 100Hz. É possível notar algumas frequências que são excitadas logo após o impacto e que decaem até que haja outro impacto, principalmente nas regiões entre 20 e 100Hz. O espectrograma foi calculado com tempo de janelamento de 100ms e sobreposição de 90%, apenas para visualização, os parâmetros utilizados para as análises serão comentados mais adiante.

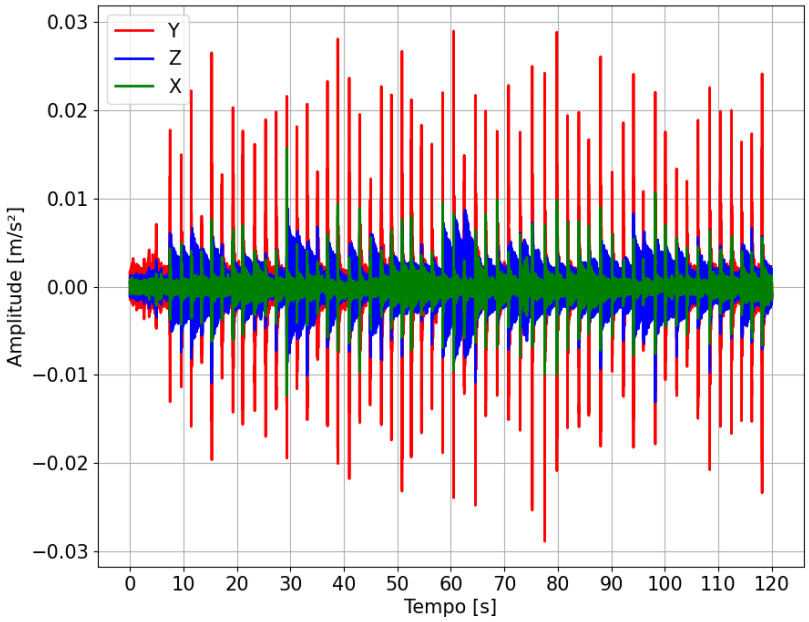

Figura 11 - Exemplo de um sinal de aceração medido para a tecla “1”, com acelerômetro posicionado no ponto I.

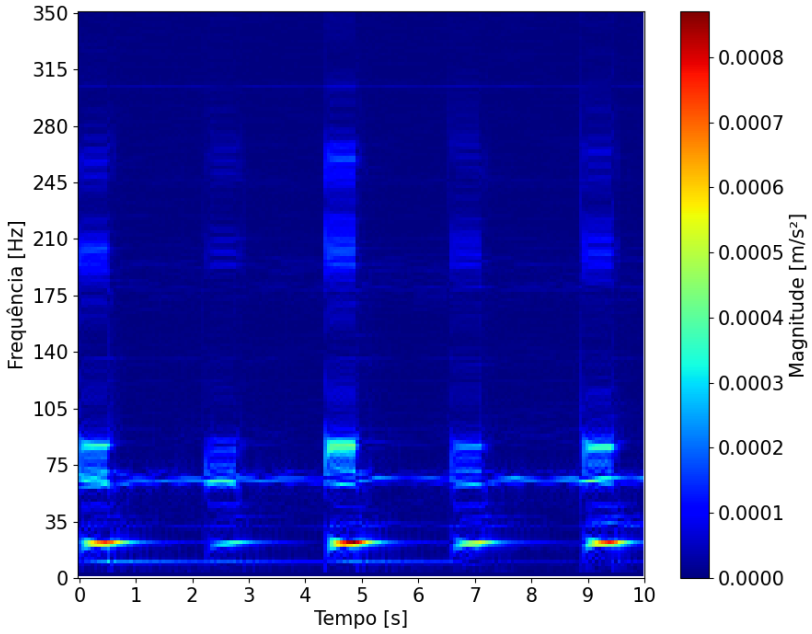

Figura 12 - Espectrograma dos 10s iniciais da medida para a tecla “1” na direção Y.

Para padronizar os dados para a entrada das redes neurais artificiais, um código de pós-processamento foi desenvolvido para identificar os impactos e salvar vetores de 1s após os impactos. Com isso, para cada tecla, em cada configuração do acelerômetro, teremos uma matriz de dados com aproximadamente 60 linhas (vide Tabela 4) e 10000 colunas, sendo que cada linha representa um impacto e cada coluna representa um instante de tempo após o impacto até a marca de 1s. Note que para os dados mostrados acima esse tempo de 1s não é suficiente para que mais de um impacto esteja no mesmo vetor de dados. Essa é uma simplificação, já que as superposições de teclas e impactos serão trados em medidas mais à frente, na Seção 5.

Outra simplificação, foi a normalização dos dados temporais com respeito à maior amplitude presente em cada corte de 1s de dados. Essa normalização visa remover a influência da força de digitação sobre os algoritmos de classificação, uma vez que, mesmo em uma digitação real, a força de pressionamento de tecla pode variar devido a vários fatores não controláveis. As Figuras 13, 14, 15, e 16 mostram respectivamente o sinal no tempo (antes da normalização), a transformada de Fourier, o espectrograma e a transformada continua wavelet (CWT) de um único impacto, com tempo de análise de 1s. Esses dados serão a entrada das redes neurais artificiais, que serão apresentadas na Seção 4.

A Transformada de Fourier (OPPENHEIM 1975) é uma técnica matemática utilizada para decompor um sinal, que pode ser uma função de tempo contínuo ou discreto, em suas componentes de frequência. Introduzida em 1821 por Joseph Fourier, e difundida a partir do século XX, a transformada de Fourier, hoje em dia, é amplamente utilizada em diversas áreas da ciência, como engenharia, física, matemática, processamento de sinais, entre outras, sendo crucial na análise de sinais oscilatórios, como os utilizados aqui. A ideia básica por trás da transformada de Fourier é que qualquer sinal pode ser representado como uma soma (ou integral) de ondas senoidais de diferentes frequências e com diferentes fases e amplitudes. Cada uma dessas ondas senoidais é chamada de "componente de frequência", e a transformada de Fourier permite encontrar a amplitude e a fase de cada uma delas. A transformada de Fourier de um sinal contínuo para um componente de frequência pode ser calculada com a Equação 3, onde $f(t)$ é o sinal temporal, ω é a frequência a ser analisada e i é a unidade imaginaria ($i^2 = \sqrt{-1}$). Já para o caso de sinais discretos, como o que é adquirido pelo nosso sistema, utilizamos a função mostrada na Equação 4, onde $f(n)$ é o valor do sinal adquirido na posição n, k é o índice de frequência relativo ao comprimento N do sinal.

$$F(\omega) = \int f(t)\ e^{-i\omega t} dt \qquad \text{(eq. 3)}$$

$$F(k) = \sum f(n)\, e^{-2i\pi nk/N} \qquad \text{(eq. 4)}$$

Para os dados da Figura 14, foi utilizada a biblioteca *python scipy* (SCIPY 2022), e a função *fft* (FFT SCIPY 2022). O resultado da transformada de Fourier são as componentes de frequência do sinal desde 0Hz até a frequência de Nyquist, mostrada na Equação 1, em nosso caso 5kHz. A resolução também é dada pelo tamanho do sinal (t_{sinal}) utilizado, e, sendo assim, como utilizamos 1s, nossa resolução em frequência (f_{res}) é: $f_{res} = 1/t_{sinal} = 1Hz$.

Similarmente, o espectrograma é composto por uma série de transformadas de Fourier performadas ao longo do sinal de interesse, fazendo com que seja possível detalhar a intensidade de cada frequência no sinal. Como desvantagem, a resolução em frequência do espectrograma é reduzida, uma vez que a transformada de Fourier tem que ser feita em um sinal de comprimento menor. Para computar os valores mostrados na Figura 15, foi utilizada a função *spectrogram* (SPECTROGRAM SCIPY 2022), também da biblioteca *scipy*. Os parâmetros utilizados na função *spectrogram* foram escolhidos como: janela retangular (*boxcar*) de 2000 amostras e sobreposição de 1953 amostras, para que a matriz resultante tenha tamanho de 128 x 128 valores, também resultando em uma imagem de mesmo tamanho em pixels. A janela retangular foi escolhida para que seja válida a suposição de sinal estocástico (sendo que o primeiro e último ponto serão forçados a 0) mas também para que a intensidade relativa entre as componentes de frequência do sinal, seja mantida.

A transformada de wavelet (MALLAT 2008) é uma ferramenta para avaliar sinais em que há diferentes detalhes variando de tamanho em escalas de tempo. A transformada de wavelet possui duas formas: discreta e continua. Nesse trabalho serão analisadas ambas as formas da transformada, em momentos distintos. Para calcular a Transformada Continua Wavelet (CWT), funções de base, ou filtros, são multiplicados ao sinal temporal, e o resultado é a intensidade de semelhança entre estes sinais, fazendo com que a CWT seja muito dependente da janela utilizada. A grande vantagem do uso da CWT é sua característica de multiresolução, que realça detalhe de alta frequência quando a função base é utilizada com um comprimento pequeno, e de baixa frequência quando a função base é utilizada com um comprimento (temporal neste caso) maior. Estas diferentes escalas possibilitam que diferentes aspectos dos sinais sejam realçados no resultado e dão nome ao gráfico resultante da aplicação da CWT, sendo ele o

“escalograma”. Vários tipos de funções base (ou janelas) estão disponíveis nas diversas bibliotecas para serem utilizados.

A biblioteca *python* utilizada para o cálculo das CWT foi a *pyWavelets* (PYWAVELETS 2022). Utilizando a função *cwt* (CWT PYWAVELETS 2022), com a janela *mexh* e com discretização com 128 escalas, foi gerado o escalograma mostrado na Figura 16. Esta janela foi escolhida dado ao seu uso comum em aplicações de identificação de imagem e extração de detalhes de imagens, como em SARVAYA 2010. Apesar de nossos dados não serem imagens, essa janela servira como ponto de partida para nossa análise. O número de escalas foi escolhido para que o tamanho da transformada na direção vertical fosse de 128 valores, para que a imagem final tenha o tamanho de 128 pixels, assim como a imagem do espectrograma. As razões para estas escolhas ficarão claras na Seção 4.3, quando utilizaremos estas imagens para treinar uma rede neural convolucional.

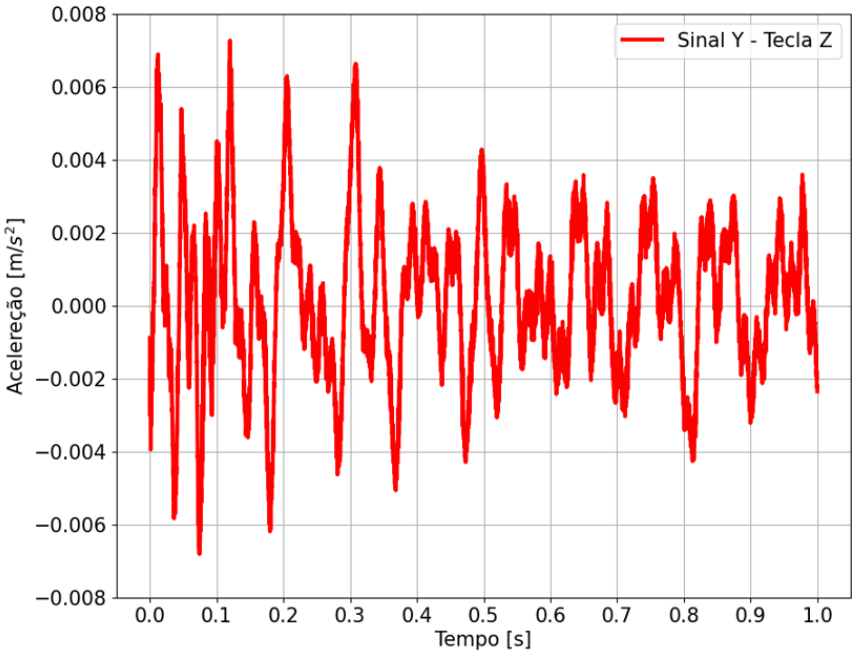

Figura 13 - Série temporal de aceleração dos dados de um único impacto da tecla [Z].

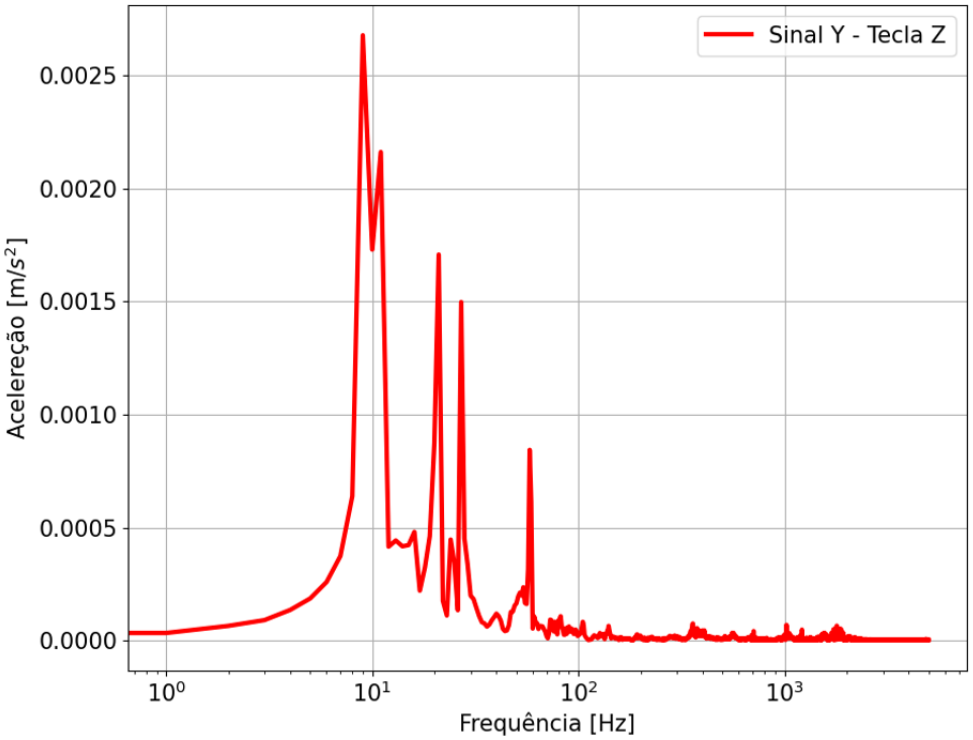

Figura 14 - Transformada de Fourier para os dados temporais da tecla [Z] mostrados na Figura 13.

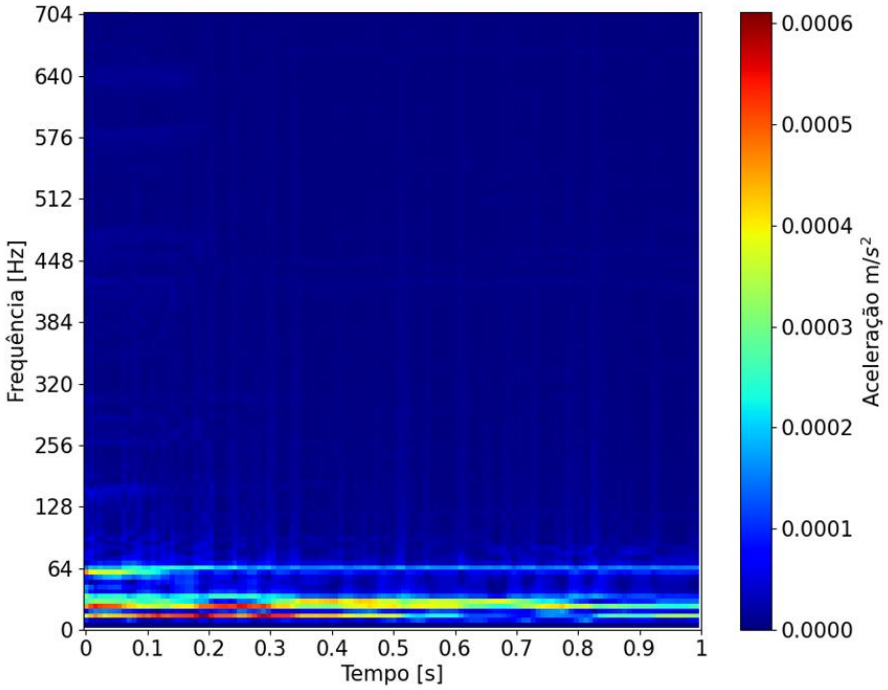

Figura 15 - Espectrograma da aceleração na direção Y da tecla “Z” mostrada na Figura 13.

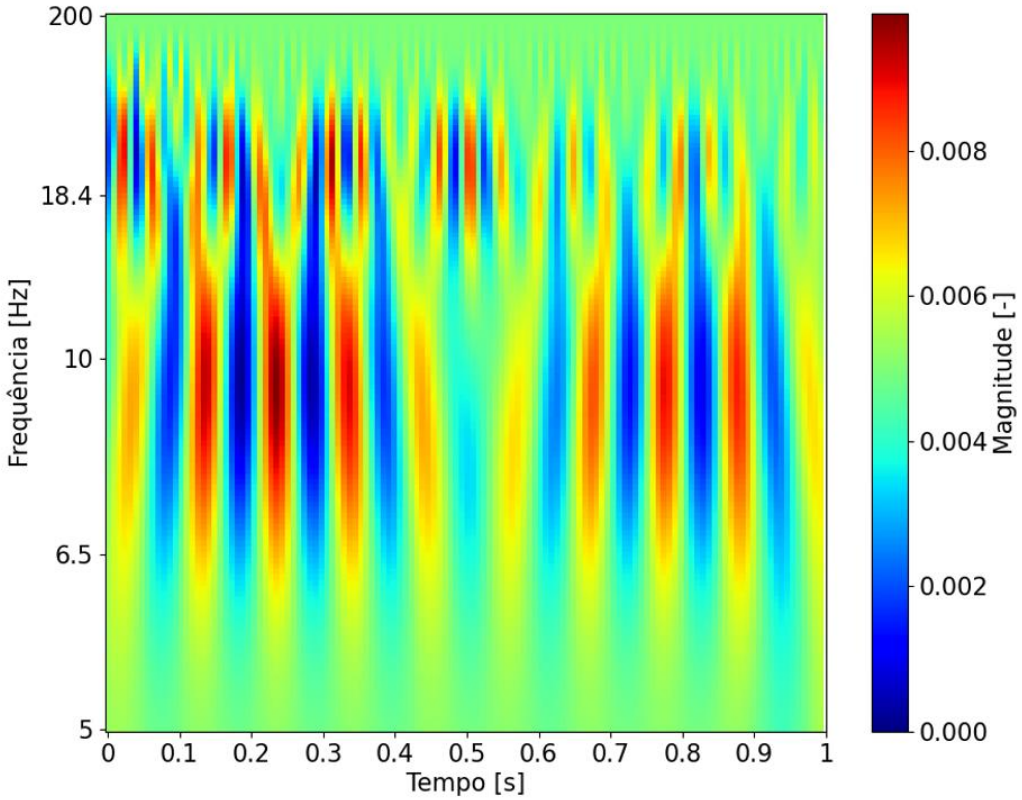

Figura 16 – Escalograma da transformada de Wavelet contínua do sinal das teclas mostrada na Figura 13.

3.4 Extração de detalhes dos sinais

A extração de detalhes é um método de análise de sinais utilizado em muitas aplicações de classificação de sinais e aprendizado de máquina, como mostrado em algumas referências da Seção 2 (PATTNAIK 2016) (DI NARDO 2022) (AZIZ 2021). Ao contrário dos métodos apresentado anteriormente, a extração de características relevantes dos sinais capturados, visa a otimizar a performance da classificação por redes neurais artificiais, diminuindo a dimensionalidade do dado de entrada para a rede neural. Consequentemente, esse método resulta em uma entrada mais simples, mas que foca no que se percebe como o mais importante para a classificação dos sinais.

A transformada discreta wavelet (DWT) apresenta-se como uma ferramenta poderosa para a decomposição de sinais em sub-bandas de frequência, permitindo a análise multiresolução dos sinais, e permitindo que seja utilizada para a extração de features em sinais.

A aplicação da DWT nos sinais adquiridos resulta em uma representação que captura tanto informações temporais quanto de frequência, o que é particularmente útil para sinais não-estacionários como os estudados nesta pesquisa.

A DWT funciona através da aplicação de filtros passa-baixa e passa-alta ao sinal original, seguida da subamostragem, o que resulta em dois conjuntos de coeficientes: aproximações (detalhes de baixa frequência) e detalhes (componentes de alta frequência). Esse processo pode ser iterado, aplicando-se a mesma lógica aos coeficientes de aproximação para obter um novo conjunto de detalhes de alta frequência (mais baixa que o anterior) e um novo conjunto de aproximações, resultando em uma análise multiresolução. Com esta técnica cada nível de decomposição fornece uma visão mais granular do sinal, permitindo isolar características específicas em diferentes escalas de frequência.

Para cada sub-banda de frequência obtida pela DWT, diferentes features estatísticas foram extraídas. Estas incluem medidas de tendência central, como a média e a mediana, medidas de dispersão, como o desvio padrão, além de medidas de forma, como a assimetria. Essas características estatísticas fornecem um resumo informativo das propriedades dos sinais em cada sub-banda, capturando aspectos essenciais do sinal, que serão relevantes para a tarefa de classificação, e servirão de entrada para o treinamento da rede neural.

A decomposição utilizando DWT foi calculada utilizando a função *dwt* (DWT PYWAVELETS 2023) da biblioteca *pyWavelets* (PYWAVELETS 2022) sucessivamente ao longo de 8 sub-bandas de frequência. Dois conjuntos de dados foram criados com janelas distintas, para posterior avaliação das redes neurais. Dada a característica de mudança rápida do sinal de digitação, estamos interessados em uma janela que possua suporte no tempo mais estreito. O primeiro conjunto de dados foi calculado utilizando-se a janela *Daubechies*, com 2 coeficientes, que possibilita uma boa localização no tempo e frequência das features dos sinais. O segundo conjunto dado foi calculado utilizando-se a janela *Symlet* com 5 coeficientes, que possui capacidade de extrair detalhes de frequências mais altas, já que tem uma resolução temporal mais fina.

Para exemplificar a decomposição descrita, a Figura 17 mostra a decomposição de um sinal do tipo *chirp* em coeficientes de aproximação e detalhe utilizando a janela *Symlet*. Um sinal chirp é um tipo de sinal senoidal, em que a frequência aumenta com o tempo. O sinal utilizado no exemplo segue Equação 5, onde t é tempo, k é a taxa de variação da frequência e

f_0 é a frequência inicial do sinal. Observando o gráfico dos coeficientes de detalhes, é possível notar que quando mais alta a camada, mais baixa é a frequência dos detalhes extraída.

$$x(t) = sin(2\pi(f_0 + kt)t) \quad \text{(eq. 3)}$$

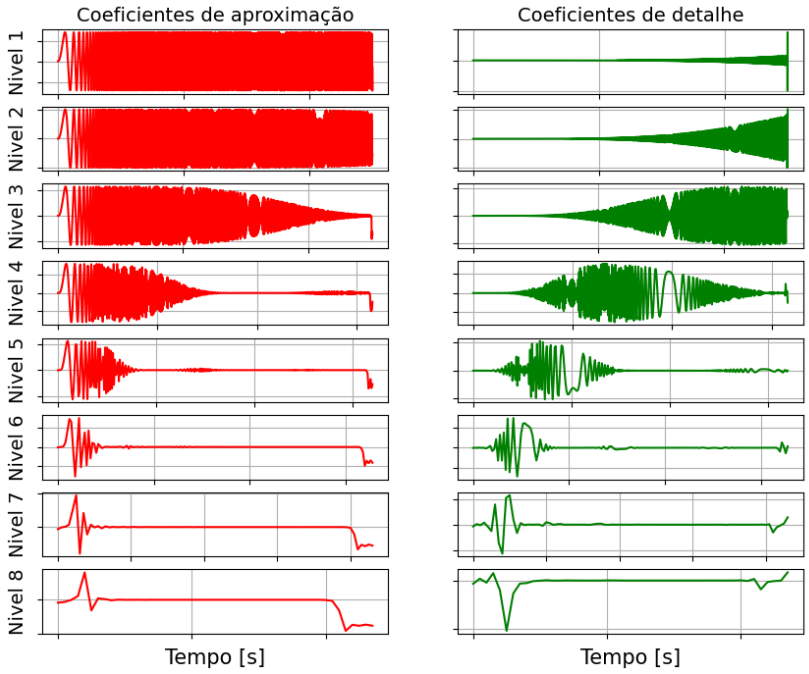

Figura 17 - Decomposição utilizando DWT em 8 camadas, com a janela *Symlet* no sinal de exemplo do tipo *chrip*.

Já a Figura 18 mostra a decomposição descrita no mesmo sinal temporal mostrado na Figura 13, utilizando a janela *Symlet* com 2 coeficientes. A Tabela 7 mostra os detalhes extraídos de cada sub-banda de frequência para o mesmo sinal para as janelas *Symlet*. As métricas, utilizadas para extração de detalhes, foram escolhidas de forma a extrair as informações relevantes de cada camada do sinal, como a amplitude pico-a-pico, e o desvio padrão, assimetria, entre outros, baseando-se em artigos mostrado na Seção 2 (PATTNAIK 2016) (ASIF 2023) e na análise dos resultados pelo autor. Vale ressaltar que os detalhes são extraídos dos coeficientes de detalhes, e não dos coeficientes de aproximação. Esse vetor de detalhes será utilizado como entrada da rede neural para identificação da tecla digitada.

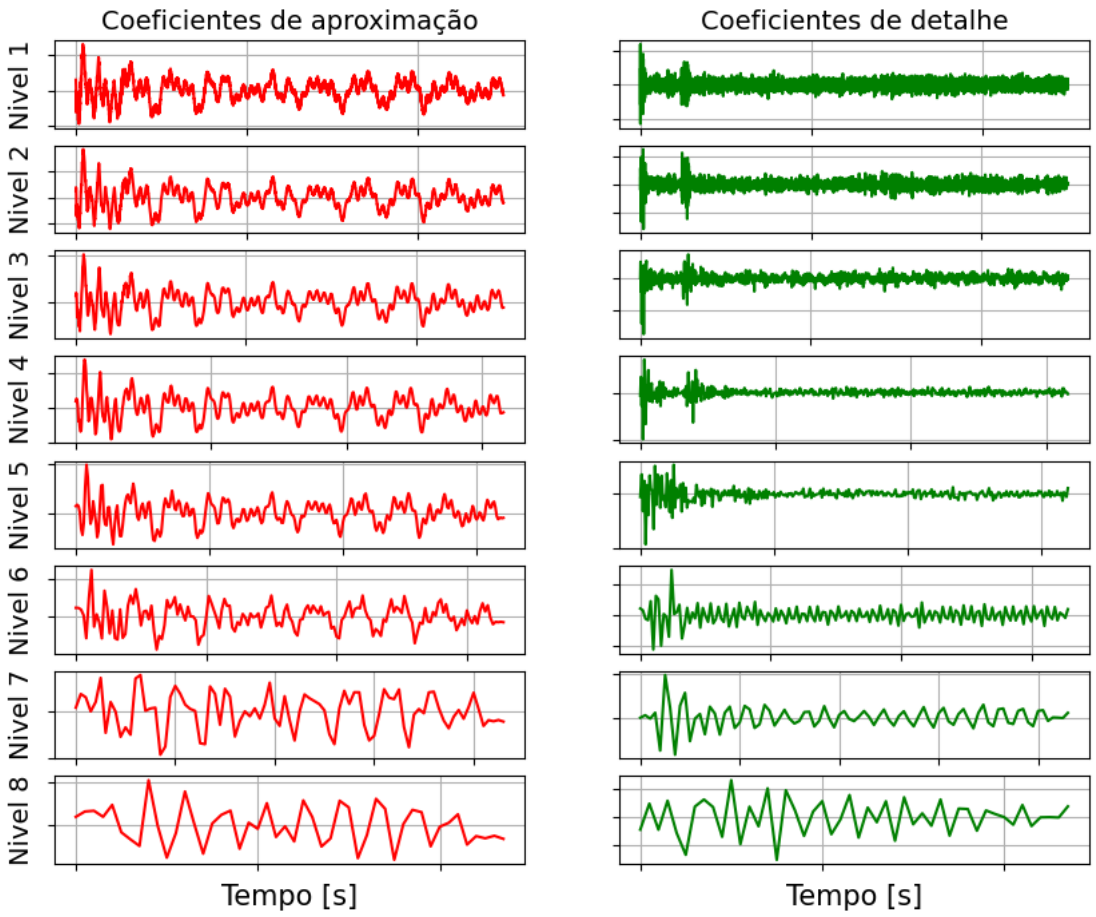

Figura 18 - Decomposição utilizando DWT em 8 camadas, com a janela *Symlet* no sinal temporal da Figura 13.

Tabela 7 - Detalhes extraídos de cada uma das sub-bandas de frequência utilizando a DWT.

	Symlet							
Camada	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Pico-a-pico	2.36e-03	7.24e-03	6.19e-03	8.30e-03	7.42e-03	2.60e-02	9.05e-02	7.09e-02
Mediana	8.33e-07	3.05e-06	-2.0e-05	-1.21e-5	-3.26e-6	-2.08e-4	7.87e-05	1.01e-03
Desvio Padrão	1.32e-04	4.44e-04	3.59e-04	4.62e-04	6.18e-04	2.83e-03	1.20e-02	1.51e-02
Entropia	4.80	4.50	4.17	3.19	3.67	3.03	2.81	2.53
Assimetria	5.54e-02	-3.8e-01	-2.11e+0	-1.65e+0	-1.30e+0	1.67e-01	-1.51e-1	-3.06e-1
95% percentil	2.01e-04	6.53e-04	4.81e-04	4.81e-04	5.97e-04	3.76e-03	1.33e-02	2.21e-02
75 % percentil	7.95e-05	2.42e-04	1.77e-04	1.67e-04	2.25e-04	1.78e-03	7.46e-03	9.49e-03

Além dos detalhes estatísticas, foram consideradas features espectrais, derivadas da análise de Fourier dos sinais. A magnitude dos componentes de frequência oferece informações

valiosas sobre a estrutura e dinâmica dos sinais analisados. A seleção de detalhes foi feita utilizando os 5 maiores picos de frequência, ou seja, no caso do dado mostrado na Figura 14, o vetor de detalhes seria composto pelos picos e frequências: 0.0027 na frequência de 9Hz, 0.0022 na frequência de 12Hz, 0.0017 em 20Hz, 0.0015 em 29Hz e finalmente 0.0008 em 58Hz. Vale ressaltar que, apesar da raia de 10Hz ser mais intensa que as frequências de 20, 29 e 58Hz, esta não foi considera pois não é o máximo loca. Outro ponto interessante destes detalhes é que, na maioria dos dados, os sinais de maior amplitude são as frequências naturais da mesa medidas na Seção 3.2, como nesse exemplo sendo 12Hz, 20Hz e 29Hz.

A integração dos detalhes estatísticos e espectrais forma um conjunto de características que alimentam o modelo de rede neural. Este conjunto de detalhes foi cuidadosamente projetado de forma a capturar a complexidade e a variação dos sinais de maneira a maximizar a eficiência da rede neural na distinção entre as diferentes classes de sinais.

O processo de seleção e extração de detalhes descrito nesta subseção é fundamental para a construção de um modelo de classificação eficaz. As técnicas empregadas visam a reduzir a dimensionalidade dos dados enquanto preservam as informações essenciais necessárias para uma classificação precisa. Os resultados obtidos, discutidos na seção seguinte, demonstram a eficácia deste método na melhoria da acurácia e desempenho do modelo de rede neural proposto.

4 PROCESSAMENTO DE SINAIS USANDO REDES NEURAIS

As redes neurais são modelos computacionais inspirados no funcionamento do cérebro humano, que permitem que um computador aprenda a realizar tarefas a partir de exemplos. Elas são compostas por camadas de neurônios artificiais, que processam informações e transmitem sinais para a próxima camada. Na propagação direta, os dados de entrada são passados pela rede neural através de suas camadas, e cada neurônio realiza uma combinação linear dos sinais que recebeu, multiplicando e ponderando por seus pesos de cada neurônio. Essa combinação é então passada por uma função de ativação, que determina a saída do neurônio e a transmite para a próxima camada. A função de ativação é responsável por introduzir não-linearidade na rede neural e permite que ela seja capaz de modelar funções complexas. Existem diversas funções de ativação, como a *sigmoid*, a *relu* e a *tanh*, cada uma com suas características e aplicações específicas (KERAS 2022). A Figura 19 mostra como os pesos e os bias (w_{kn} *e* b_k) são aplicados ao sinal de entrada (x_n) e o resultado a função de ativação (φ), para que tenhamos a saída (y_k) deste neurônio, onde k se refere ao índice do neurônio em sua camada e n se refere ao índice do neurônio da camada anterior, que gerou o sinal de entrada.

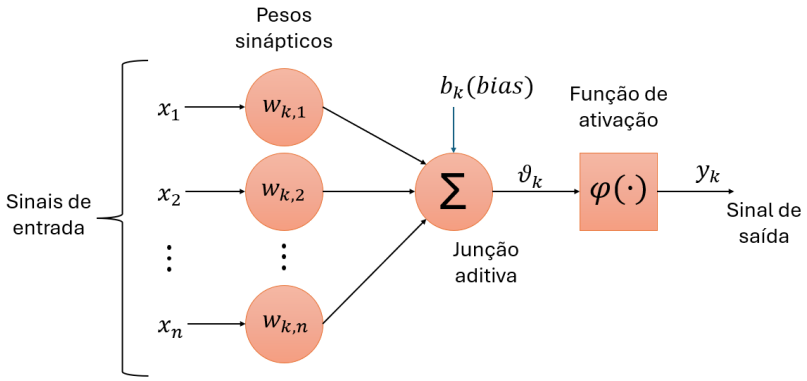

Figura 19 - Esquemático de um neurônio de uma camada escondida mostrando o fluxo de informação através dos pesos, do bias e da função de ativação.

Para o treinamento da rede, utilizamos a etapa de *backpropagation* (NIELSEN 2019), onde a rede neural ajusta os pesos de cada neurônio para minimizar o erro entre a saída que ela produz e a saída desejada. Esse ajuste é realizado através do cálculo do gradiente da função de erro em relação aos pesos atuais e então usado para atualizar os pesos através de um algoritmo

de otimização. O *backpropagation* é a chave para o aprendizado da rede neural e sua invenção mudou o rumo desta tecnologia, permitindo que ela ajuste seus pesos de forma a melhorar seu desempenho a cada iteração. É importante destacar que essa otimização exige uma grande quantidade de dados de treinamento para ser efetivo, e que a escolha adequada da arquitetura da rede e dos hiperparâmetros de treinamento (número de camadas, número de neurônios, tipo de função de ativação, etc) é fundamental para obter resultados satisfatórios.

Outro ponto importante a se considerar é que o treinamento de uma rede neural é completamente dependente dos dados de entrada que estão sendo analisados. Deste modo, caso os dados possuam viés, ou seja, por exemplo, o conjunto de dados da tecla "A" a ser classificado (classe), possui o dobro de dados do que o conjunto da tecla "B', isso influenciará a rede final encontrada. Uma forma de se avaliar esse viés de dados é utilizando a estratégia de *k-fold cross-validation* (KOHAVI 1995). O *k-fold cross-validation* é uma técnica comum de validação de modelos em machine learning, que tem como objetivo avaliar a capacidade de generalização do modelo treinado. Essa técnica consiste em dividir o conjunto de dados disponível em k partições ou "*folds*" de tamanho igual. Em seguida, o modelo é treinado k vezes, cada vez usando $k-1$ partições como dados de treinamento e a partição restante como dados de validação. O processo é repetido até que todas as partições tenham sido usadas como conjunto de validação. Vale explicitar que o conjunto de testes é o mesmo para todas as partições e é separado do conjunto completo antes da partição. Ao final do processo, é calculada a média dos resultados obtidos em cada iteração, para obter uma estimativa geral do desempenho do modelo.

Outras técnicas de avaliação de viés poderiam ter sido escolhidas como *Monte Carlo Cross-Validation* ou *Jackknife Resampling*. Entretanto, a técnica do *k-fold cross-validation* é especialmente útil quando se tem um conjunto de dados pequeno ou limitado, como o do problema tratado nesse trabalho, pois permite uma avaliação mais robusta do modelo sem comprometer a quantidade de dados disponíveis para treinamento. Vale ressaltar também que o conjunto de dados usado nesse trabalho possui quantidades próximas de impactos, ou seja, dados de digitação, por classe (tecla), e por posição do acelerômetro e sendo assim, não utilizaremos nenhuma técnica de regularização ou balanceamento, como por exemplo, o aumento artificial o tamanho do conjunto de dados (*data augmentation*) ou regularização L1 e L2.

As seções a seguir serão dedicadas à exploração de diferentes tipos de redes neurais artificiais aplicadas ao problema de identificação de sinais de vibração. Nessas seções

discutiremos a aplicação de três tipos distintos de redes neurais artificiais: redes totalmente conectadas (*Fully Conected*), redes neurais convolucionais (*CNN*s) e redes neurais recorrentes (*RNN*) sendo elas do tipo redes de memória de curto e longo prazo (*Long Short-Term Memory).* Para cada um desses tipos de rede serão apresentadas suas vantagens e desvantagens, bem como testes com os dados coletados nesse trabalho. Ao final do capítulo será possível comparar a acurácia de cada um dos tipos de rede aplicados ao problema em questão, bem como ter uma informação dos parâmetros importantes em cada um dos modelos.

4.1 Redes Neurais Totalmente Conectadas

As redes neurais densas, ou totalmente conectadas, são um tipo especial de rede onde todos os neurônios de uma camada anterior estão conectados com todos os neurônios de uma camada posterior, desde a primeira camada ou entrada até a última camada ou saída. Esse tipo de rede não é muito usado para análise de imagens, dada sua baixa performance computacional com dados grandes e sua dificuldade de extrair detalhes de dados contínuos. Apesar disso, tem aplicação prática em muitos casos de classificação de dados (LECUN 1998). O seu atributo de ser totalmente conectada acaba sendo um problema para redes muito profundas, ou seja, que possuem muitas camadas escondidas, pois a quantidade de cálculos para treinamento aumenta demasiadamente com o número de neurônios por camada e proporcionalmente ao número de camadas da rede. A Figura 20 mostra um esquema simples do arranjo de uma rede neural totalmente conectada. Observe como o número de ligações entre neurônios é grande mesmo para uma rede pequena e pouco profunda.

4.1.1 Dados no tempo

Primeiramente, nesta seção, os dados utilizados como entrada para a rede são os dados temporais mostrados na Seção 3.2, quando o sensor estava posicionado no ponto I. O primeiro parâmetro a ser avaliado será a profundidade da rede, ou seja, o número de camadas escondidas que resultam em boas aproximações para os resultados de teclas individuais. Como os dados

são de 1s de duração, em uma determinada direção de aquisição do acelerômetro, e adquiridos a 10kHz, cada dado de entrada tem a forma de um vetor com 10000 posições. Sendo assim, a camada entrada deve possuir 10000 neurônios, um para cada posição do vetor do dado de entrada. Além disso, para simplificação, todas as camadas escondidas foram programadas com o mesmo número de neurônios, sendo de 100 neurônios para cada camada. A camada de saída, ou de resultados, deve ter o número de neurônios igual ao número de teclas que devem ser classificadas. Assim, a camada de resultados tem 4 neurônios. Dados os parâmetros de tamanho da rede, a próxima escolha se dá com o tipo de algoritmo de ativação a escolher para cada neurônio. As funções de ativação *sigmoid* e *tangente hiperbólica* são amplamente utilizadas para dados contínuos como os de vibração (KERAS 2022) e, para os primeiros testes, utilizamos a função *sigmoid*, que retorna sempre resultados positivos para qualquer entrada.

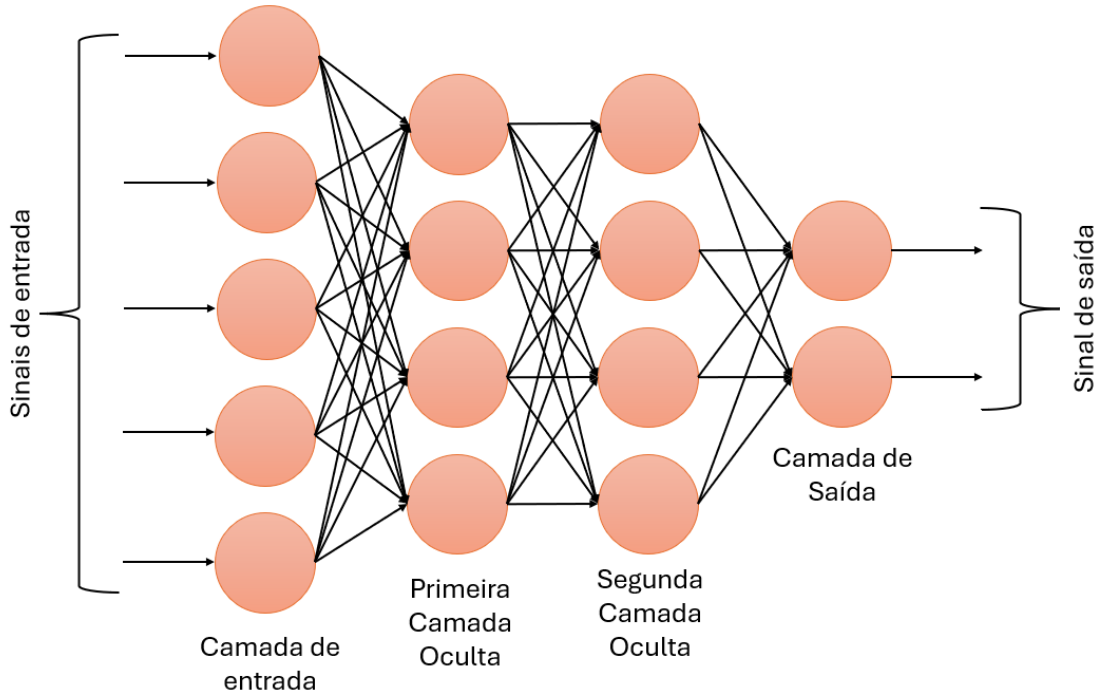

Figura 20 - Esquema de uma rede neural totalmente conectada com 2 camadas escondidas.

$$S(a) = \frac{1}{1+e^{-a}} \qquad \text{(eq. 5)}$$

Ao observar a Equação 5, da função de ativação escolhida, onde $S(a)$ é a saída do neurônio para uma entrada a, podemos ver que números muito altos terão resultados muito parecidos entre si, assim como números muito baixos, devido às assíntotas da função. A Figura

21 mostra o gráfico da saída dessa função no intervalo $-5 < a < 5$ e evidencia o comportamento assintótico mencionado acima. Vale ressaltar que todos os dados temporais adquiridos foram normalizados pelo maior valor dentro de um mesmo corte de 1s. Assim, todos os dados de entrada da rede estarão no intervalo de [-1,1]. Já para a camada de saída, como os dados provenientes da rede são tratados quanto à probabilidade daquele sinal de entrada pertencer à respectiva classe daquele neurônio de saída, utilizamos a função de ativação *softmax*, que é amplamente utilizada em problemas de classificação (KERAS 2022).

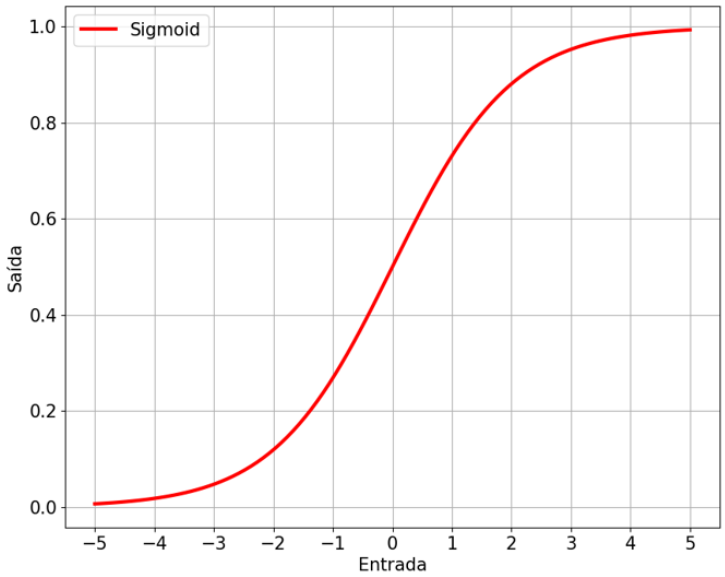

Figura 21 - Função de ativação *sigmoid* avaliada com entradas de -5 a 5.

Um ponto importante que deve ser analisado dessa escolha de normalização é que os dados de amplitude estarão mascarados por ela. Intuitivamente podemos imaginar que as teclas que estão mais próximas ao acelerômetro terão uma energia maior e consequentemente uma amplitude maior de aceleração, o que pode ser um fator importante a ser considerado pela rede e melhorará a eficácia de identificação. Com a normalização, a rede não terá mais essa informação de intensidade de forma explícita, mas ainda precisamos avaliar os efeitos deste decréscimo de informação na acurácia das redes neurais propostas. Entretanto, sabemos que o modo como uma pessoa tecla pode variar muito de dependendo do indivíduo, horário do dia, sequência que está sendo digitada e outros fatores aleatórios. Assim, pode-se concluir também que a informação de amplitude pode confundir o algoritmo, já que na realidade, este aspecto do sinal varia com fatores fora do controle da rede.

Outros parâmetros escolhidos para a rede inicial são:

-Otimizador: '*Adam*', por ser o mais usado e ter bons resultados em classificações;

-Função de perda: '*Categorical crossentropy*', que é otimizada para dados de classificação e trabalha com a saída no formato de probabilidade;

-Número de épocas: 200;

-Tamanho dos *batchs*: 10;

Os nomes mostrados são com base nos nomes utilizados pelas bibliotecas *keras* (KERAS 2022) e *tensorflow* (TENSORFLOW 2022).

Comumente em algoritmos de redes neurais, os dados disponíveis para treinamento são divididos em três subconjuntos: dados de treinamento, dados de validação e dados de teste. O primeiro subconjunto se refere aos dados que serão realmente utilizados para o treino dos parâmetros da rede, e fornecerão entradas e saídas conhecidas para o algoritmo de otimização. Já os dados de validação, são dados que não são usados para o treinamento, na verdade, o uso esses dados, dará-se para avaliar se a rede está de fato aprendendo a identificar os sinais medidos ou se está ocorrendo o *overfitting*, quando a rede aprende detalhes específicos sobre cada amostra ou conjunto de amostra, que fazem o algoritmo convergir para um mínimo que não representará todos os dados coletados, nesse caso resultando em uma acurácia muito alta no treinamento, mas baixa nos dados de validação. O resultado ideal esperado é uma acurácia alta em ambos os conjuntos de dado, demonstrando que a rede está aprendendo detalhes relevantes dos sinais de treino que podem ser generalizados para a descrição de novos dados. Já os dados de teste, são utilizados ao final do treinamento para que haja uma estimativa de performance da rede com dados nunca vistos pela mesma, gerando resultados mais precisos e confiáveis da acurácia.

Devido ao número pequeno de sinais para treinamento da rede, os dados adquiridos foram divididos na proporção de 75% para os dados de treinamento e 15% para os dados de validação e 10% para os dados de teste. Uma vez que a escolha de dados para cada conjunto é feita aleatoriamente, um conjunto muito pequeno de validação pode estatisticamente conter apenas dados com uma única característica que não está ou é pouco presente nos dados de treinamento. O modo de se evitar esse problema é utilizando um conjunto de dados grande, ou dividi-los de maneira mais equiparada, sendo que geralmente a divisão é de 60% dos dados para treino e 20% para validação e 20% para os dados de teste. Outra forma de se avaliar a performance da rede sobre todos os dados é utilizando o método *k-fold cross validation,* mencionado no começo da

seção, que será utilizado para todos os casos apresentados na proporção de 75% de treinamento e 15% de validação e 10% de dados de teste. Ou seja, os valores apresentados de acurácia correspondem à média dos valores encontrados nos treinamentos com diferentes porções de dados.

No caso em questão, a resposta da rede que estamos treinando é um vetor com quatro elementos. Cada elemento está associado a uma classe (tecla) e corresponde a probabilidade (valor entre 0 e 1) de que o dado de entrada seja pertencente tecla representada por aquele campo. Vale ressaltar que a soma de todos os campos deste vetor de saída resulta em 1 em todas as circunstâncias. Em um exemplo prático, gostaríamos que dada uma entrada qualquer, nossa rede nos devolvesse como resposta um vetor com três elementos 0 e um elemento com valor 1, ou seja, 100% de certeza que aquele sinal é da tecla corresponde a este campo com valor 1. Entretanto, não é isso que vemos, em um caso real, a resposta da rede para a classificação do sinal é a tecla correspondente ao campo com maior valor no vetor de saída. Para medir a diferença entre a resposta ideal e a encontrada e realimentar o processo de treinamento utilizamos a função de perda (*loss function*) que também é uma métrica importante para se avaliar o treinamento de uma rede neural. Em nosso caso, a função perde mede o quão distante a resposta encontrada esta da ideal utilizando o erro RMS para tal. O cálculo do erro RMS vem da camada de saída da rede, ou seja, o erro é calculado comparando o vetor de saída da rede, com as acurácias calculadas, em contraste com o vetor considerado ideal, sendo ele três elementos 0 e um elemento 1 na casa da tecla digitada. O erro calculado é o primeiro passo para o algoritmo de treinamento da rede começar a atualizar os pesos e *bias*.

Esse primeiro teste tem o objetivo de avaliar o comportamento da rede totalmente conectada quando utilizada com dados temporais na direção Y, com acelerômetro na posição I, descritos na Seção 3.3, para teclas próximas e teclas distantes. Os dados da direção Y foram escolhidos pois é a direção que está alinhada com a força da digitação, assim possuindo uma amplitude maior que as direções ortogonais. Exemplo de sinais para todas as teclas do conjunto de teclas distantes é mostrado na Figura 22. Utilizando o modelo descrito acima, foram treinadas redes com diferentes números de camadas, sendo eles: 4, 8, 12, 16, 20, 24, 30, 40, 50 e 60 camadas. Os resultados são mostrados na Tabela 8.

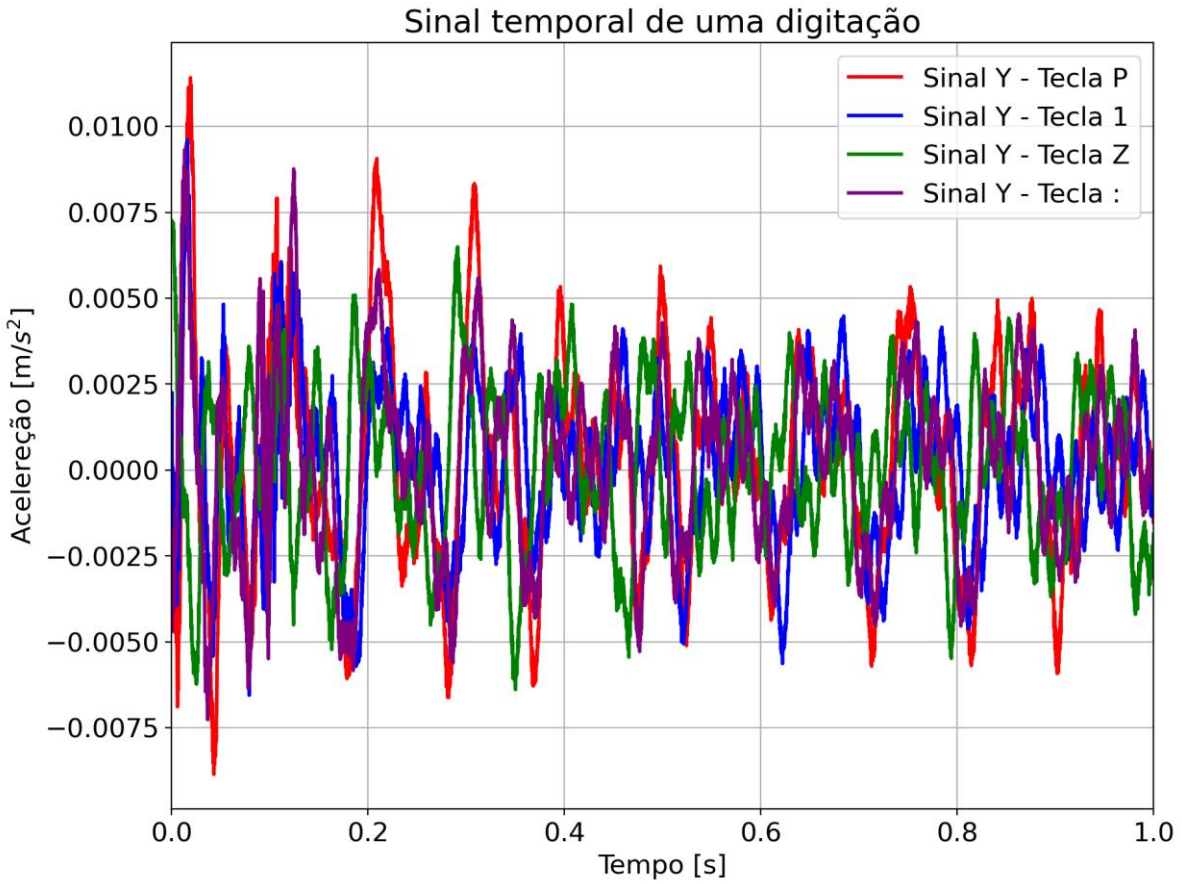

Figura 22 - Exemplo de sinal temporal para cada uma das teclas do conjunto de teclas distantes.

Tabela 8 – Resultados de acurácia obtidos no treinamento das redes neurais totalmente conectadas com diferentes números de camadas escondidas para os dados no tempo.

Número de camadas	**Teclas próximas**		**Teclas distantes**	
	Treinamento	**Teste**	**Treinamento**	**Teste**
4	32%	24%	38%	36%
8	35%	30%	46%	44%
12	40%	36%	54%	56%
16	42%	44%	62%	60%
20	40%	40%	64%	64%
24	40%	40%	66%	60%
30	52%	40%	65%	60%
40	71%	40%	78%	56%
50	100%	36%	90%	56%
60	98%	40%	94%	52%

Para as teclas distantes, é possível notar que há uma relação entre o número de camadas e a acurácia da rede para os sinais no tempo, uma vez que a rede menos profunda apresenta acurácia de até 40% na classificação dos dados de teste, e as redes com 16 até 30 camadas apresenta acurácia na casa de 60%. É possível notar que parece haver uma saturação na acurácia dos dados de teste da rede próximo a 60%, sendo que os dados para 16, 20, 24 e 30 camadas apresentam acurácias de mesma ordem. Outra forma de se analisar os resultados é olhando para a matriz de confusão do experimento, que mostra o resultado esperado e o resultado obtido pela rede. A matriz de confusão para a rede com 16 camadas para ambos os conjuntos de teclas é mostrada na Figura 23, a seguir. Podemos notar que não parece haver um *bias*, ou erro preferencial da rede, ou seja, a rede não parece estar confundindo uma dada tecla sempre com outra.

Figura 23 - Matriz de confusão para o experimento com rede totalmente conectada, entrada como sinal temporal de 1s e número de camadas escondidas de 16. À esquerda a matriz de confusão para as teclas próximas e à direita a matriz de confusão para as teclas distantes.

Já para as redes com 40 camadas ou mais, é possível notar um aumento significativo da acurácia de treinamento, sem que esse aumento seja refletido na acurácia dos dados de teste. Isso indica um problema de *overfitting*, quando um modelo muito complexo aprende detalhes específicos dos dados de treinamento, que são irrelevantes para a classificação dos demais dados. Sendo assim, quando isso ocorre, temos um aumento significativo da acurácia de classificação de treinamento, mas uma acurácia muito menor nos dados de teste. Esse comportamento evidencia a importância de se avaliar a acurácia da rede com dois conjuntos de dados distintos, e a aplicação do método de *k-fold cross-validation*, para evitar possíveis conclusões sobre acurácias que não representam o modelo.

Além disso, quando comparamos os resultados para o conjunto de teclas próximas e teclas distantes é possível notar uma diferença nos melhores resultados encontrados. Geralmente, para um mesmo número de camadas, menor que 30, os resultados para as teclas distantes são melhores que os resultados encontrados para as teclas próximas. Pode-se inferir que a rede tem maior dificuldade de identificar os sinais provenientes de um local próximo, uma vez que estes devem ser muito parecidos, já que a resposta do sistema deve variar pouco com pequenos deslocamentos da entrada de força. Apesar disso, o número de camadas para a saturação da acurácia parece ser o mesmo nos dois casos, sendo que os as redes com número de camadas maior ou igual a 16 não têm diferenças significativas até que se haja o *overfitting* acima de 40 camadas.

4.1.2 Dados no Domínio da Frequência

Como mostrado na subseção anterior, a utilização dos dados temporais em redes totalmente conectadas gerou resultados limitados de acurácia, próximos a 65%, quando avaliados com um conjunto de teclas distantes, e limitados a 42% quando utilizado o conjunto de teclas próximas. Para investigar mais a fundo alternativas de treinamento de redes com diferentes formatos de sinais, essa seção se dedica a apresentar os resultados obtidos quando os dados de entrada estão no formato de um vetor de frequência, e não mais um vetor temporal, como na seção anterior.

Os mesmos dados coletados nos conjuntos de teclas distantes e próximos, na direção de aquisição Y, é agora analisado em suas componentes de frequência. Como explicitado na seção 3.3, cada sinal de 10000 pontos temporais é tratado utilizando a transformada de Fourier (*fft*), que resulta em um sinal complexo no domínio da frequência, sendo que a informação absoluta corresponde à amplitude do sinal e a informação de ângulo corresponde à fase do sinal para aquela frequência. Nesta primeira aproximação, serão apenas considerados os dados de amplitude, e à frente será feita uma comparação com o treino usando os dados de fase e amplitude em conjunto. A resolução em frequência, e a frequência mínima são dadas pelo inverso do tempo de experimento, e nesse caso correspondem a 1Hz (inverso de 1s), e a frequência máxima adquirida, dada pela equação 1, é de 5kHz.

A rede utilizada é a mesma mostrada na subseção 4.1.1, para que haja uma comparação explícita entre as abordagens no domínio temporal e no domínio da frequência. Como a Transformada de Fourier retorna também 10000 pontos, não é necessário fazer ajustes na entrada da rede. Apesar da simetria entre o lado positivo e o lado negativo da transformada de Fourier, o dado completo, ou seja, espelhado, foi passado para a rede. A Figura 24 mostra a Transformada de Fourier para 4 amostras do conjunto de teclas distantes. É possível notar pequenos detalhes que diferem os 4 sinais do conjunto de teclas distante, e o objetivo da rede é aprender esses detalhes para poder distinguir entre os sinais.

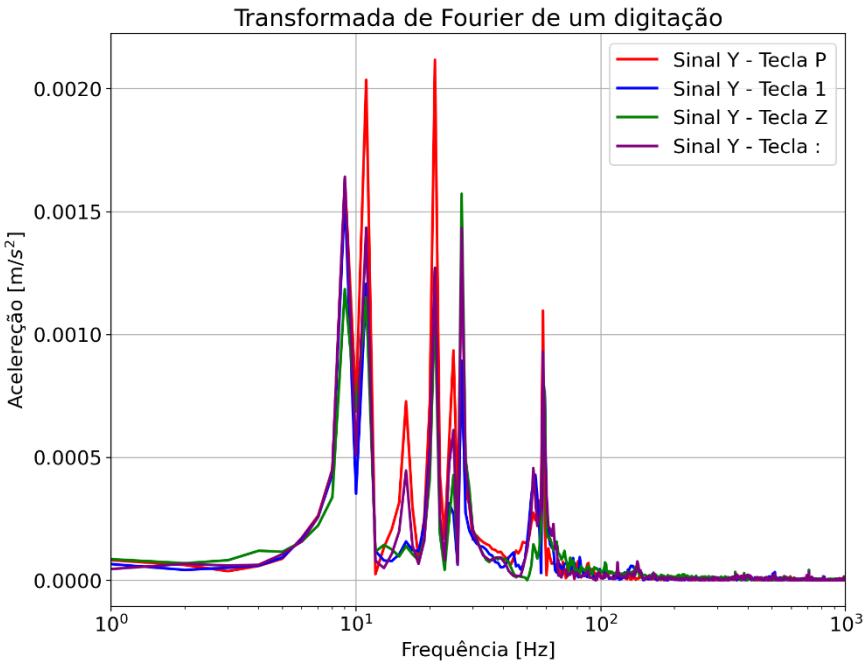

Figura 24 – Lado positivo da Transformada de Fourier para impactos do conjunto de teclas distantes.

A Tabela 9 resume os resultados encontrados para as redes com entrada das amplitudes da Transformada de Fourier. Assim como visto nos treinamentos com dados no tempo, a profundidade da rede parece melhorar a acurácia até por volta de 16 a 24 camadas, e então ficar uma porcentagem de acerto de aproximadamente 64% para os dados de teclas distantes e 44% para os dados de teclas próximas. Assim como nos dados temporais, notamos um aumento da acurácia de treinamento acima de 30 camadas, enquanto a acurácia dos dados de teste continua valores próximos e até menores, evidenciando novamente a ocorrência de *overfitting*.

Tabela 9 - Resultados de acurácia das redes neurais totalmente conectadas para os dados de entrada da Transformada de Fourier.

Número de camadas	Teclas próximas		Teclas distantes	
	Treinamento	Teste	Treinamento	Teste
4	32%	28%	33%	30%
8	33%	32%	45%	40%
12	36%	32%	52%	52%
16	43%	44%	64%	64%
20	42%	44%	65%	64%
24	45%	44%	67%	60%
30	48%	44%	70%	60%
40	60%	40%	83%	52%
50	90%	40%	91%	56%
60	95%	32%	92%	48%

Analisando mais a fundo o resultado, como estamos usando como entrada do modelo apenas a amplitude de cada frequência, pode-se inferir que a quantidade de informação contida no vetor de entrada para a rede de frequência é menor do que a informação contida no vetor temporal, mesmo que mostrada de outra forma. Novamente olhamos para as matrizes de confusão para as redes com 16 camadas, mostradas na Figura 25 e podemos ver que não há erros sistemáticos da rede, a não ser pela letra “S” estar sendo identificada muitas vezes, mesmo que de forma errônea nos dados de teclas próximas.

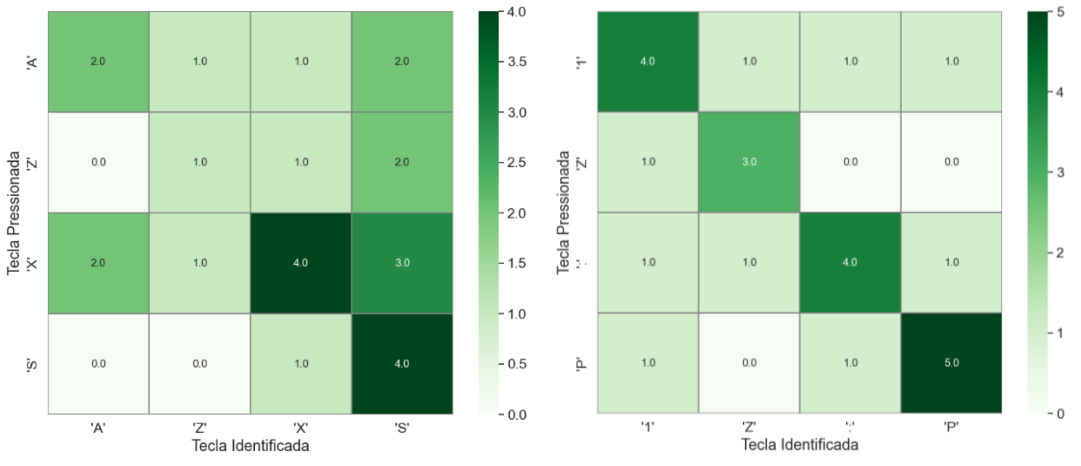

Figura 25 - Matriz de confusão para o experimento com rede totalmente conectada, entrada como sinal no domínio da frequência de 1 a 5000Hz e número de camadas escondidas de 16. À

esquerda, a matriz de confusão para as teclas próximas e, à direita, a matriz de confusão para as teclas distantes.

Os valores e acurácia máxima obtido são semelhantes para ambos os casos, o que pode resultar em duas hipóteses: mesmo com menos informação, a rede aprende melhor com os dados de frequência, ou as informações de fase não são relevantes para a aprendizagem de padrões e identificação dos sinais. Para comparação dos resultados, a Figura 26 mostra os resultados encontrados para as redes neurais com diferentes números de camadas ocultas para os dados de teclas próximas nos casos de dados temporais e em frequência, e a Figura 27 para os dados com teclas distantes.

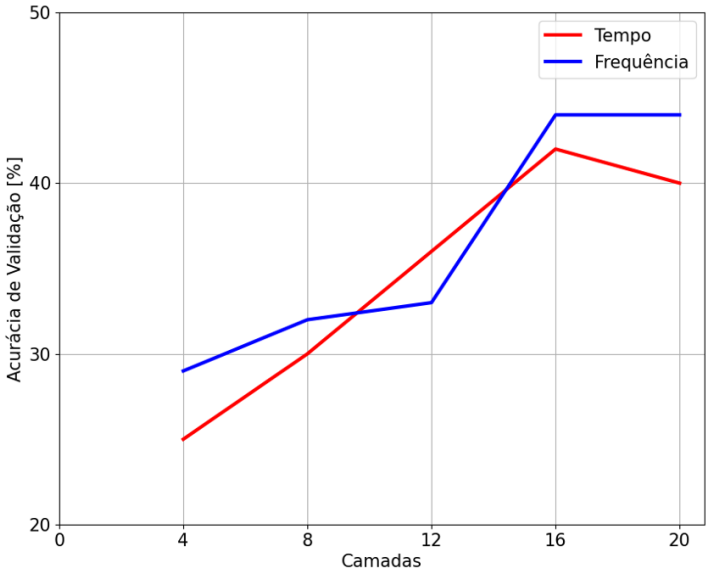

Figura 26 - Comparação da acurácia de redes neurais completamente conectadas, com diferente número de camadas, para entrada com dados temporais e em frequência, para o conjunto de teclas próximas.

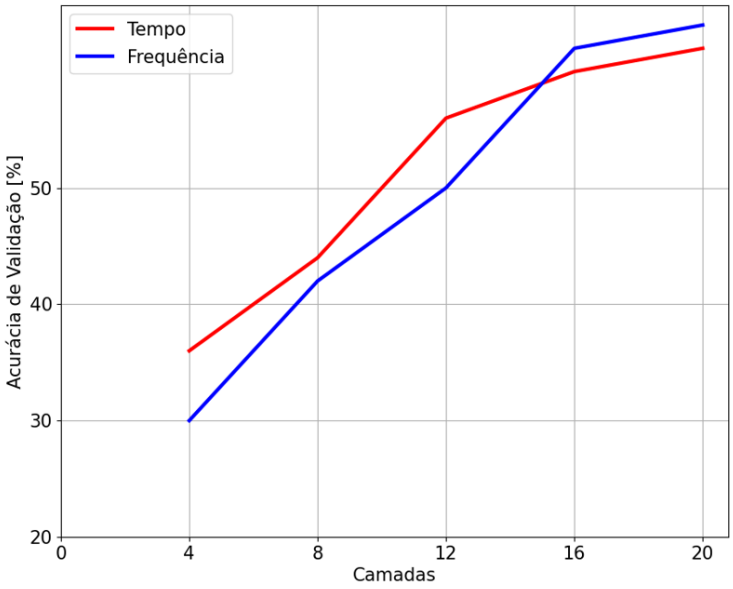

Figura 27 - Comparação da acurácia de redes neurais completamente conectadas, com diferente número de camadas, para entrada com dados temporais e em frequência, para o conjunto de teclas distantes.

4.1.3 Espectrogramas

Após os resultados limitados das últimas duas seções, o objetivo agora é avaliar se há uma maneira de melhorar a acurácia da rede totalmente conectada utilizando manipulações no dado de entrada. Baseado nos resultados dos sinais temporais e de frequência apresentados, o passo seguinte foi utilizar os sinais tratados na forma de espectrograma como entrada da rede neural. Utilizando os dados de espectrograma, espera-se que seja mais explícito para a rede as relações de diferentes frequências excitadas quando há o impacto em diferentes pontos do sistema, assim como nas Transformadas de Fourier utilizadas na seção anterior. Além disso, a forma do cálculo do espectrograma deixa mais evidente as relações transientes entre os sinais, uma vez que as análises em frequências são feitas em diferentes instantes dentro do mesmo sinal.

Os sinais de 1s adquiridos foram tratados com a função *spectrogram* do pacote s*cipy*. Sendo assim, a saída dos dados está com unidades de m/s^2. Os dados de espectrograma são armazenados em matrizes de tamanho 128x128 resultando em 16384 pontos de entrada para a rede. Para utilizar o mesmo formato das redes propostas para os dados temporais e em

frequência, a matriz dos dados do espectrograma foi rearranjada de forma que as colunas se concatenassem. Deste modo, o vetor de entrada para a rede tem o tamanho esperado de 16384 linhas.

Já quanto à normalização dos dados, os espectrogramas foram normalizados pelo maior valor da matriz, e posteriormente multiplicados por 256, resultando em dados de 0 a 256, que podem ser entendidos como um espectrograma em escala de cinza, como mostrado na Figura 28. Além disso, foram realizados testes com os espectrogramas das três direções medidas para se avaliar qual a melhor direção para utilização dos dados de entrada e ao final foi analisada uma rede treinada com os três dados simultaneamente.

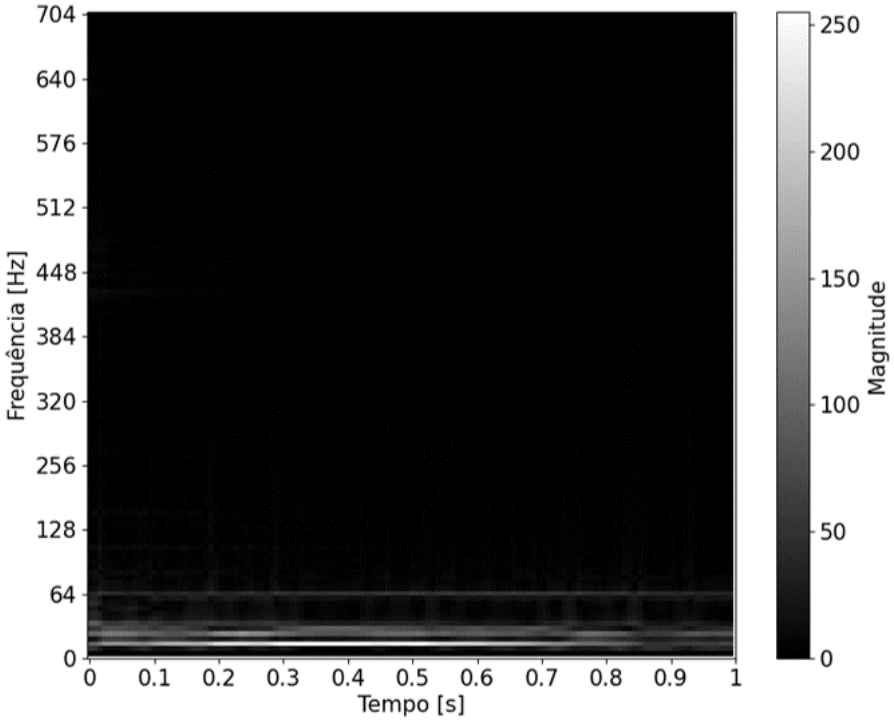

Figura 28 – Exemplo de espectrograma normalizado em escalas de cinza.

Devido aos testes das seções anteriores, para esse caso foi utilizada a rede com 16 camadas ocultas, função de ativação *sigmoid* para as camadas internas e *softmax* para a última camada. Relembrando, a resposta da rede que estamos treinando é um vetor com quatro elementos. Cada elemento está associado a uma classe (tecla) e corresponde a probabilidade (valor entre 0 e 1) de que o dado de entrada seja pertencente tecla representada por aquele campo. Para medir a diferença entre a resposta ideal e a encontrada e realimentar o processo de treinamento utilizamos a função de perda (*loss function*) que também é uma métrica importante para se avaliar o treinamento de uma rede neural. Em nosso caso, a função perda mede o quão

distante a resposta encontrada está da ideal utilizando o erro RMS para tal. A Figura 29 mostra valor da função perda ao longo do treinamento para rede de teclas distantes na direção Y.

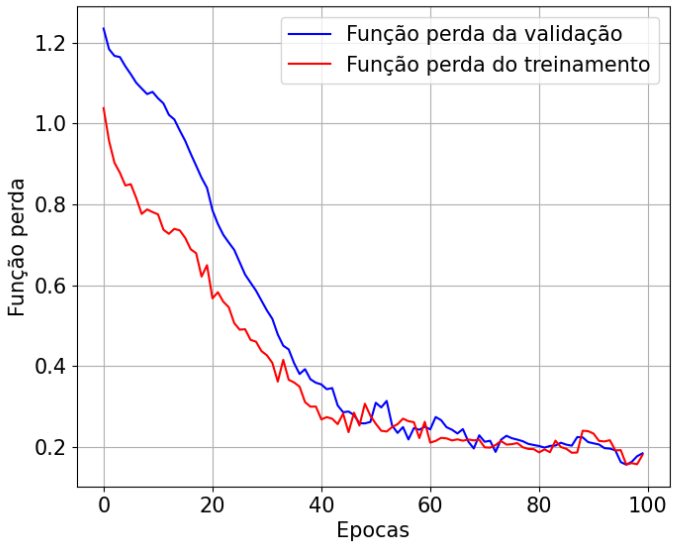

Figura 29 - Curvas de perda para os dados de treinamento e validação correspondentes à rede neural treinada com dados em Y para o conjunto de teclas distantes.

Tabela 10 – Resultados de acurácia para as redes totalmente conectas treinadas com dados de espectrogramas em diferentes direções.

Direção	**Teclas próximas**		**Teclas distantes**	
	Treinamento	**Teste**	**Treinamento**	**Teste**
X	35%	32%	38%	38%
Y	47%	45%	65%	65%
Z	38%	38%	40%	38%

Os resultados mostrados pelas redes para cada uma das direções estão compilados na Tabela 10. É possível notar que que a direção Y (vertical) gera os melhores valores de acurácia do que as direções X e Z, o que vai em linha com o esperado, uma vez que os sinais são mais intensos na direção de impacto durante a digitação. Além disso, não é possível notar uma melhora com relação aos resultados obtidos nas secções anteriores com os dados temporais e em frequência. Ainda assim, a acurácia do modelo fica em apenas 65% para as teclas distantes e 45% para as próximas. Já para as direções X e Z, não vemos uma diferença pronunciada entre

a acurácia da rede para teclas próximas e distantes. Isso pode ser explicado pela menor amplitude do sinal, que decai rapidamente ao nível de ruido do acelerômetro, limitando a quantidade de informação disponível em ambos os casos.

Uma nova rede foi treinada utilizando os dados das três direções simultaneamente, em uma forma de pareamento dos dados, visando fornecer o máximo de informação possível para a rede afim de maximizar o aprendizado e a acurácia do modelo. Os dados foram concatenados na forma $[X, Y, Z]$ resultando em um vetor de entrada de 49152 posições. A rede foi modificada para receber este novo vetor de entrada, e nas camadas internas foi feito um acréscimo de neurônios em cada uma, passando de 100 para 300, um acréscimo de 3 vezes, assim como no vetor de entrada. A saída da rede, como continuamos tentando identificar 4 teclas por conjunto de dados, continua com a saída de 4 classes.

Mesmo com o maior número de informações passado para a rede, não foi possível obter resultados melhores que os obtidos nos testes anteriores. Com o maior número de parâmetros a serem treinados, o tempo total de treinamento cresceu com relação aos testes anteriores e mesmo assim, não houve melhora nos resultados apresentados pela rede.

É possível que, mesmo com mais dados, a capacidade de aprendizagem da rede totalmente conectada está em seu limite não conseguindo extrair mais detalhes dos dados em questão. Em uma análise minuciosa, poderiam ser testados outros parâmetros para aumentar a acurácia da rede, como avaliar qual a melhor função de ativação, quais os melhores parâmetros de treinamento e o efeito de se utilizar os dados de fase da Transformada de Fourier. Em vista do demasiado número de opções a se otimizar, resolveu-se priorizar o estudo de outros tratamentos e outras redes neurais que serão apresentados nas próximas seções.

4.1.4 Extração de detalhes

Dados os resultados acima, fica claro que o sinal temporal, no domínio da frequência ou na forma de um espectrograma, como entrada da entrada da rede neural totalmente conectada, possibilita a identificação de uma parcela limitada das amostras. Como dito, há a possibilidade de que mesmo com mais parâmetros, a rede esteja em sua capacidade máxima de distinguir detalhes que possam ajudar na identificação dos sinais. Uma forma de testar essa hipótese e

contornar esse problema é utilizando método similar ao usado em PATTNAIK 2016 e ASIF 2023, ou seja, realizando-se a extração de detalhes antes da passagem dos dados à rede neural. Essa abordagem visa a explorar como a redução de dimensionalidade e a concentração nas características mais relevantes dos sinais podem influenciar a eficácia da classificação das teclas digitadas em um contexto de identificação de sinais de vibração.

Como mostrado na Seção 3, a extração de detalhes visa a diminuir a complexidade do sinal de entrada da rede neural, uma vez que o sinal bruto é pré-filtrado com a seleção de algumas variáveis de interesse. Dados os bons resultados mostrados em ASIF 2023, decidiu-se utilizar a Transformada Discreta de Wavelet (DWT, mostrada na Seção 3.4) como forma de realizar a extração de alguns detalhes. Essa decisão é baseada na similaridade dos sinais a serem identificados e no benefício da utilização de diferentes janelas wavelet para extração das diferentes features.

Dados os resultados mostrados na seção 4.1.3, apenas os sinais da direção Y da posição de medição I serão considerados neste momento. Os sinais foram tratados utilizando as janelas *Daubechies* (Seção 3.4) e *Symlet*, formando dois conjuntos de dados distintos, onde cada conjunto possui o subconjunto de dados de teclas próximas e de teclas distantes. As janelas foram utilizadas respectivamente com 2 e 5 coeficientes, para que as janelas pudessem destacar os detalhes desejados. A utilização dessas duas janelas tenta analisar onde estão os detalhes de maior importância para classificação do sinal, se nas oscilações bruscas, evidenciada pela janela *Daubechies* ou se nas oscilações de alta frequência destacadas na janela *Symlet*.

Além dos detalhes extraídos utilizando a DWT, também foram adicionados aos dados de entrada os detalhes extraídos da Transformada de Fourier, sendo eles os 5 maiores picos, em ordem de intensidade, e suas respectivas amplitudes. Essa adição se baseia na observação da Figura 24 onde é possível notar um comportamento ligeiramente diferente na dinâmica dos picos dada as teclas digitadas. Apesar de não constantes, como dito na Seção 4.1.2, dadas as pequenas variações mesmo após a normalização do sinal, essas pequenas diferenças podem ajudar esta nova rede a identificar os sinais.

Dados os resultados discutidos anteriormente, observou-se a importância da escolha no número de camadas das redes neurais totalmente conectadas, para alcançar uma classificação eficiente dos sinais medidos. Como nas análises realizadas nas seções 4.1.1 e 4.1.2, onde a profundidade da rede e o número de camadas escondidas foram avaliados para otimizar o desempenho da rede, esta seção visa a aplicar um raciocínio similar para refinar ainda mais a

nossa escolha com os novos dados de entrada. Através deste processo, espera-se validar a hipótese de limitação da capacidade de aprendizado da rede e estabelecer uma configuração arquitetônica que maximize a eficácia da classificação dos sinais de vibração, levando em consideração as informações obtidas da extração de detalhes.

A nova configuração da nova rede terá entrada ajustada para acomodar as features da DWT e as features espectrais extraídas da Transformada de Fourier, resultando em uma entrada com 66 neurônios, sendo 7 para cada um dos detalhes das 8 sub-banda de decomposição, e mais 10 para os 5 maiores picos da Transformada de Fourier e suas respectivas intensidades. Os outros hiperparâmetros da rede foram mantidos como nas seções anteriores, sendo eles: otimizador '*Adam*', função de cálculo de perda '*Categorial crossentropy*', número de épocas 200 e tamanho dos *batchs* de 10 amostras. O número de camadas foi novamente variado, mas dessa vez entre 4 a 16 camadas, dada a maior simplicidade dos dados de entrada, correspondendo aos valores: 4, 8, 12, 16. Visto o número limitado de amostras, os sinais de entrada foram divididos nos conjuntos de treinamento, validação e teste com porcentagens de 75%, 15% e 10% respectivamente, assim como nos testes anteriores.

Tabela 11 - Resultados de acurácia das redes neurais totalmente conectadas para os dados de entrada da extração de detalhes utilizando janela *Symlet* com 5 coeficientes.

Número de camadas	**Teclas próximas**		**Teclas distantes**	
	Treinamento	**Teste**	**Treinamento**	**Teste**
4	47%	48%	81%	76%
8	83%	80%	95%	88%
12	84%	80%	95%	92%
16	83%	76%	92%	92%

Os resultados obtidos estão compilados nas Tabelas 11 e 12. À primeira vista, fica claro que há uma vantagem, de aproximadamente 20%, no caso da janela Symlet, com relação aos métodos apresentados anteriormente. Isso mostra a superioridade da extração de detalhes com relação ao uso dos sinais brutos (no domínio do tempo ou da frequência), indicando que, dada a complexidade do sinal bruto, a rede totalmente conectada chega a uma saturação que pode ser superada com a diminuição da complexidade e a concentração das características mais relevantes. Além disso, podemos observar que, dada a menor complexidade dos dados de

entrada, o máximo de acurácia da rede é atingido com menos camadas, sendo ele 12, para ambos os casos.

Tabela 12 - Resultados de acurácia das redes neurais totalmente conectadas para os dados de entrada da extração de detalhes utilizando janela *Daubechies* com 2 coeficientes.

Número de camadas	**Teclas próximas**		**Teclas distantes**	
	Treinamento	**Teste**	**Treinamento**	**Teste**
4	56%	52%	57%	28%
8	58%	52%	70%	68%
12	63%	60%	76%	72%
16	65%	60%	77%	72%

Também fica clara a diferença entre os resultados obtidos para as diferentes janelas wavelet, o que parece indicar uma maior concentração de informação nos detalhes evidenciados pela janela *Symlet,* ou seja, uma maior concentração de detalhes parece estar presente nas altas frequências do sinal adquirido. Esses dados, de certa forma, vão ao encontro do esperado do sistema mecânico analisado. As baixas frequências (0 – 100Hz) são dominadas pelas frequências naturais da mesa, como mostrado na Seção 3.2, sendo que as respostas da mesa variam pouco dadas as pequenas diferenças de posição de impactos de cada tecla, comparadas à dimensão da mesa. Além disso, dada a baixa massa do teclado, é esperado que as frequências naturais deste sejam em frequências mais altas e, dada sua forma esbelta e alongada, as pequenas variações de posição de impacto podem excitar modos de vibrar diferentes dentre essas altas frequências naturais.

Dada a boa resposta do método de extração de detalhes, resolveu-se também utilizar este método para uma primeira avaliação da influência da posição de medida do sensor de aceleração. Para isso, os dados coletados das posições de I a IV na direção Y foram tratados utilizando a DWT com janela *Symlet* e a extração de detalhes espectrais, tanto para o conjunto de teclas distantes, quanto para o conjunto de teclas próximas. A rede utilizada foi a de maior acurácia do experimento anterior, ou seja, a rede com 12 camadas escondidas. Os resultados estão mostrados na Tabela 13.

Tabela 13 - Resultado de acurácia da rede totalmente conectada quando avaliada utilizando a técnica de extração de detalhes, para os diferentes pontos de medida do acelerômetro.

Posição do sensor	Teclas próximas		Teclas distantes	
	Treinamento	Teste	Treinamento	Teste
I	84%	80%	95%	92%
II	90%	88%	94%	92%
III	82%	76%	87%	88%
IV	76%	76%	61%	56%

Os dados mostram uma melhora de aproximadamente 10% nos resultados, quando o acelerômetro é posicionado nas extremidades laterais do teclado, ou seja, I, II e III, para o caso das teclas distantes. Isso pode ocorrer pelo fato de que essas posições não acarretam a simetria entre os impactos de nenhuma das teclas, ao contrário da configuração IV onde as teclas “P” e “:” estão posicionadas a distancias similares do sensor que as teclas “1” e “Z”. Essa hipótese também é corroborada quando olhamos a matriz de confusão do resultado IV, mostrada na Figura 30, onde vemos que a rede classifica erroneamente algumas amostras entre os pares “P” e “1”. Apesar de as teclas estarem posicionadas em regiões diferentes com relação a borda do teclado, parece haver uma influência da simetria com relação ao sensor de aceleração, que não está presente nos outros estudos, mesmo no caso em que as teclas estão posicionadas muito próximas, ou mesmo para as teclas “Z” e “:” do mesmo conjunto.

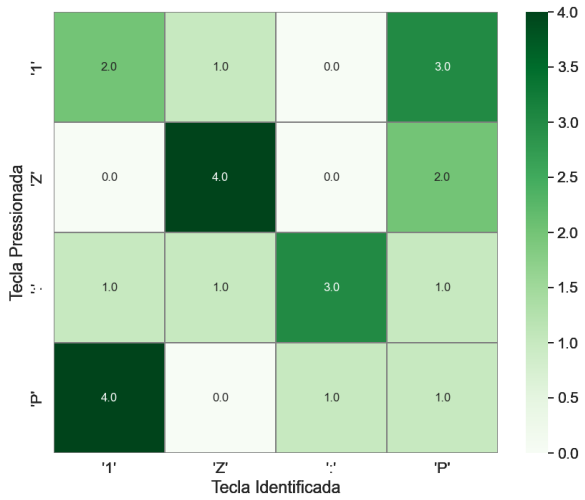

Figura 30 - Matriz de confusão para o conjunto de teclas distantes quando o acelerômetro estava posicionado na configuração IV.

Outro resultado interessante é que o conjunto de teclas próximas tem uma melhor acurácia de identificação quando o sensor está posicionado na configuração II, ou seja, mais próximo à localização desse conjunto de teclas. Esse resultado é esperado, uma vez que, com o sensor posicionado mais próximo, mais energia do impacto ao conjunto de teclas próximas chegará ao sensor, com menos ruído introduzido ao sinal, dado a menor distância de propagação.

Em suma, o método de extração de detalhes provou ter superioridade com relação à utilização do dado bruto como entrada da rede neural. Além disso, o fato de que os dados tratados com a janela *Symlet* obtiveram maior acuraria indica uma possível relação entre as características relevantes dos sinais de digitação e as altas frequências e estas serão privilegiadas nas próximas seções. Por fim, o último estudo mostrou uma baixa correlação entre a posição do acelerômetro, e a acurácia de identificação dos sinais, para o caso em que o acelerômetro está posicionado nas laterais do teclado. A posição I será utilizada para os experimentos das próximas seções.

4.2 Redes Neurais LSTM

Nesta seção exploraremos a performance das redes LSTM (*long short-term memory*) aplicadas ao problema de identificação de teclas digitadas. As redes do tipo LSTM (HOCHREITER 1998) são um tipo especial de redes recursivas (*RNN*) e são amplamente utilizadas em problemas envolvendo dados temporais. A principal característica das RNNs é a capacidade de manter uma "memória" do que foi processado anteriormente. Isso é feito através de laços recorrentes na rede, que permitem que a saída de uma célula unitária, em um determinado instante de tempo seja utilizada como entrada para essa mesma unidade em instantes subsequentes. Essas redes são particularmente úteis para tarefas onde a ordem e a dependência temporal dos dados são importantes, como em reconhecimento de fala, tradução automática, análise de séries temporais e geração de texto.

As redes do tipo LSTM foram criadas como tentativa de solução do problema do "*vanishing gradient*" presente em redes neurais recursivas (BENGIO 1994). O problema do "vanishing gradient" nas redes neurais recorrentes (RNNs) ocorre quando os gradientes que são propagados de volta através do tempo durante o treinamento tornam-se extremamente

pequenos. Isso impede que os pesos das camadas anteriores sejam atualizados de forma eficaz, dificultando o aprendizado de dependências a longo prazo nas sequências de dados. Como resultado, a RNN perde a capacidade de lembrar informações importantes de instantes temporais distantes, limitando seu desempenho em tarefas que exigem a captura de relacionamentos de longo alcance. A diferença da rede LSTM para outras redes está em sua célula unitária, como mostrado na Figura 31.

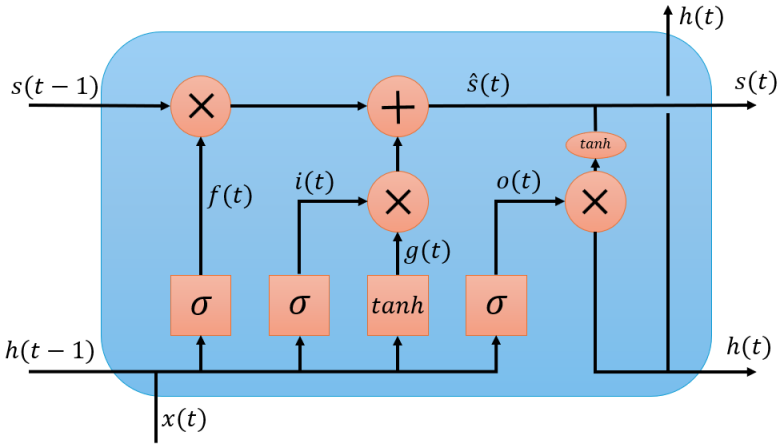

Figura 31 – Célula unitária de uma rede neural do tipo LSTM.

Uma célula unitária de uma rede neural LSTM é composta por três portões que regulam o fluxo de informações que entram e saem da célula: o portão de entrada (*input gate*), representado por i(t), o portão de esquecimento (*forget gate*), representado por f(t), e o portão de saída (*output gate*), representado por o(t).

O primeiro passo no funcionamento de uma célula LSTM é a entrada de um novo dado x(t) na célula. O portão de entrada decide qual parte dessa informação deve ser adicionada ao estado da célula, o que é feito através de uma operação de multiplicação ponto a ponto entre x(t) e a saída da função gate i(t), que determina quanto dessa entrada deve ser mantida.

Em seguida, o estado interno da célula é atualizado, somando a nova informação filtrada pelo portão de entrada. O próximo passo é o portão de esquecimento decidir quanto da informação antiga armazenada no estado s(t−1) deve ser mantida. O portão de esquecimento calcula essa quantidade usando uma função de ativação sigmoide que considera tanto o estado interno atual quanto a nova entrada.

Finalmente, o portão de saída regula a quantidade de informação que será transmitida para a próxima camada ou célula. Ele utiliza uma função de ativação sigmoide para calcular quanto

do estado interno atual s(t) deve ser enviado para a saída, combinando essa informação com a entrada atual. A saída final h(t) é então obtida pela multiplicação ponto a ponto do estado s(t) com a saída da função de ativação tangente hiperbólica.

Esses três portões operam em conjunto para permitir que a célula LSTM armazene informações relevantes a longo prazo e esqueça informações irrelevantes a curto prazo, tornando as LSTMs ferramentas poderosas para modelagem de sequências de dados. Devido à sua ampla utilização em problemas que envolvem dados temporais, as redes LSTM têm grande potencial para a classificação de sinais de teclas. Em abordagens iniciais, o uso de LSTMs em tais problemas simples ajuda a entender melhor o funcionamento e os parâmetros dessa rede complexa.

Com os bons resultados obtidos na seção anterior, o próximo passo foi aplicar a rede LSTM para os sinais coletados na Seção 3. A arquitetura de rede utilizada inicialmente foi com apenas uma camada escondida do tipo LSTM e 100 neurônios. Os sinais de entrada foram os sinais temporais na direção Y, para o acelerômetro posicionado na configuração I, recortados no tamanho do vetor de 10000 posições. Ao contrário do esperado, a acurácia para a rede ficou próxima a 25%, mesmo valor em que uma escolha aleatória acertaria o sinal correto (4 classes). Esse resultado se repetiu para o conjunto de dados de teclas distantes e próximas.

Ante os resultados negativos, outras arquiteturas de redes foram testadas, com mais camadas escondidas e mais neurônios por camadas, mas nenhuma foi capaz de aumentar a acurácia da rede. Outro teste foi feito adicionando os dados das direções X e Z ao vetor de entrada, que tomou a forma de 3 canais com 10000 amostras em cada um, mas o resultado obtido ainda foi de uma acurácia próxima a 25%.

A baixa relação sinal-ruído do sinal de entrada pode ter sido um fator determinante para a qualidade dessa rede. Outra hipótese é que as frequências presentes no sinal podem ser não constantes ao longo do tempo de aquisição, devido a possíveis não-linearidades no sistema e em seu coeficiente de amortecimento, fazendo com que a LSTM não tenha um bom desempenho.

4.3 Redes Neurais Convolucionais

Ainda tentando melhorar a acurácia do modelo, uma alternativa utilizada foram as redes neurais convolucionais (*CNN – convolutional neural networks*). As *CNN*s (O'SHEA 2015) são um tipo especial de rede, muito utilizado no reconhecimento, segmentação e extração de detalhes de imagens. O principal diferencial desta rede está no tipo de camada que dá o nome a ela, a camada convolucional, onde diferentes filtros varrem a amostra de entrada realçando detalhes e padrões para as camadas seguintes. Esse processo é chamado de convolução, uma vez que há duas imagens sendo multiplicadas.

Este tipo de rede foi considerado para esta aplicação devido à sua possível aplicação sobre os dados de imagem dos espectrogramas. Geralmente, as redes convolucionais são construídas como a sucessão de um bloco de três camadas: a convolucional, a de ativação e a de *pooling*. A primeira, como já dito, é uma camada onde uma série de filtros, como mostra a Figura 32, varrem a amostra destacando ou atenuando detalhes e padrões. Supondo uma imagem de 32×32 pixels, e camada de cor única, após aplicação de uma camada convolucional com 12 filtros e uma convolução de 1×1 pixel, a saída dessa camada é um volume 3D de $32 \times 32 \times 12$ dados, correspondendo a uma imagem de mesmo tamanho gerada por cada filtro, ou seja, a próxima camada processara 12 imagens distintas.

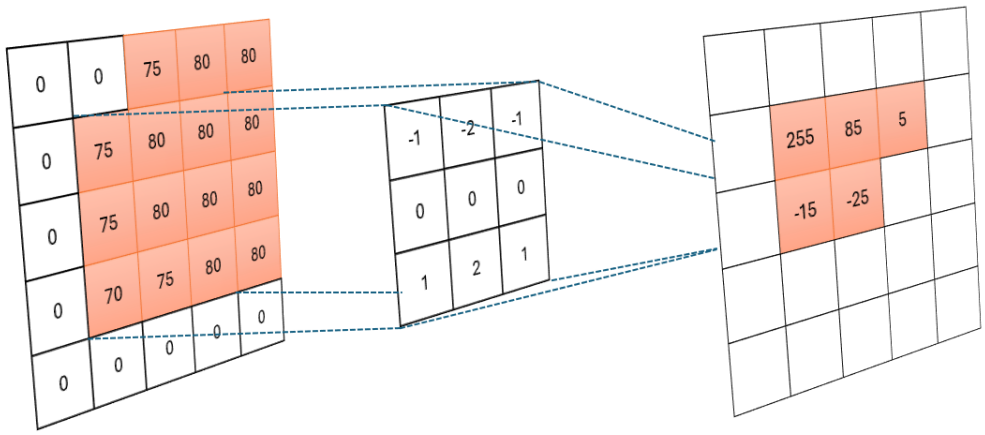

Figura 32 - Exemplo de uma camada convolucional, onde um filtro (centro) percorre a figura de entrada (esquerda) resultando na convolução mostrada à direita.

O tamanho da imagem de saída pode variar conforme o tamanho dos filtros e passo utilizados para a aplicação destes. Além disso, a profundidade do filtro deve ser a mesma que a escala de cores utilizada na imagem, ou seja, para o caso cinza, o filtro terá um tamanho $n \times m \times 1$, mas para uma imagem com padrão de cores RGB, o filtro terá um tamanho $n \times m \times 3$, sendo uma imagem a ser multiplicada com cada classe de cor. Mesmo assim, a saída é um volume com profundidade sendo o número de filtros. Isso também vale para quando há uso de várias camadas de convolução em serie. Sendo assim, caso uma camada de convolução seja aplicada a um bloco de dados de tamanho $n \times m \times p$, os filtros da camada seguinte devem ter profundidade p.

Seguida de uma camada de convolução, necessariamente está uma camada de ativação, onde cada dado do volume gerado pela camada convolucional será analisado com uma função de ativação escolhida, assim como na rede completamente conectada. Neste caso, dada uma entrada da forma $32 \times 32 \times 12$ dados, a saída terá a mesma forma, já que a função de ativação é aplicada a cada dado de entrada. As funções que podem ser escolhidas para as redes convolucionais são as mesmas já mencionadas para as redes totalmente conectadas, vide Seção 4.1. Entretanto, a mais usada geralmente é a função de ativação *relu* (*rectified linear unit*). Nas implementações atuais das bibliotecas, como a camada de convolução precisa necessariamente ser seguida de uma camada de ativação, a camada de ativação é aglutinada à camada convolucional para simplicidade de representação.

Após a aplicação do bloco convolucional e ativação, a próxima camada pode ser outra camada convolucional ou uma camada de *pooling*. As camadas de *pooling* são utilizadas para diminuir o tamanho dos dados a serem analisados por uma camada posterior e podem ser 2D ou 3D, e terem os tipos de máximo, média e soma. O método mais utilizado é o *pooling* 2D, onde apenas os tamanhos verticais e horizontais das imagens são reduzidos, mantendo-se a profundidade. O *pooling* recorta uma amostra de tamanho $n \times m \times p$, definidos pelo usuário, do bloco de dados proveniente de uma camada convolucional e gera apenas um dado $(1 \times 1 \times 1)$ de saída. Esse dado de saída pode ser o maior valor da amostra recortada (*max pooling*), pode ser a média dos dados recortados (*average pooling*) ou a soma de todos os dados (*sum pooling*).

Como exemplo, a Figura 33 mostra um *pooling* 2D de valor de máximo, utilizando uma janela de 2×2 e passo de 2 pixels por janela, aplicados a uma imagem teórica de 4×4 pixels e apenas uma profundidade de cor. A janela de *pooling* seleciona apenas o maior dado de cada

seção 2 × 2, reduzindo a imagem de 4 × 4 para 2 × 2. Diferentes tamanhos de janela e passos podem mudar o tamanho de saída para o desejado pelo usuário.

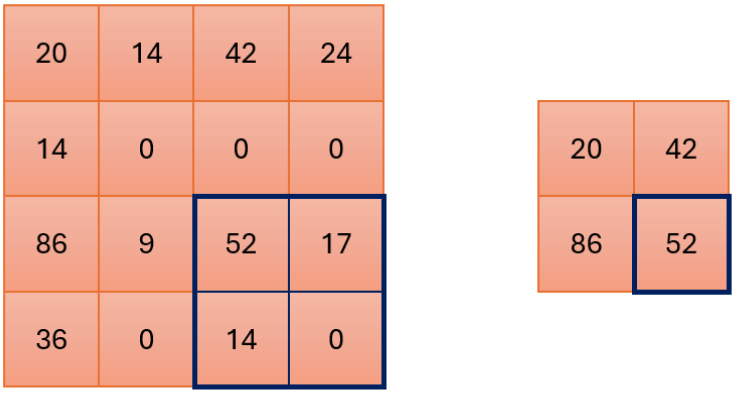

Figura 33 - Exemplo da camada de *pooling*, onde a janela azul varre a imagem (esquerda) e seleciona o maior valor de cada posição.

Como última ou últimas camadas, utilizam-se camadas totalmente conectadas que relacionam as saídas das camadas convolucionais com as ativações de cada uma das classes possíveis para a imagem. O dado final é novamente um vetor 1D com a probabilidade de a imagem de entrada ser de cada uma das classes que foram treinadas. A Figura 34 sumariza os processos que ocorrem em uma rede convolucional simples e demostram o fluxo completo de dados através dela.

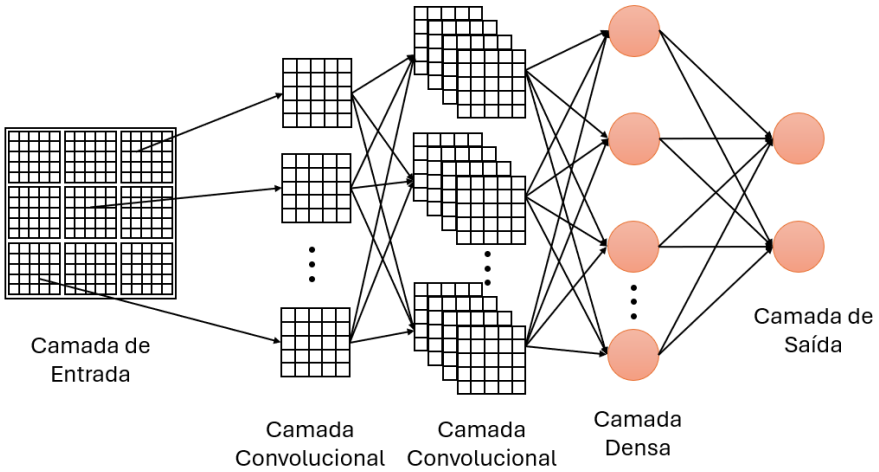

Figura 34 - Exemplo da arquitetura completa de uma rede neural convolucional simples.

4.3.1 Redes convolucionais e imagens de espectrogramas

Como mencionado anteriormente, as redes neurais convolucionais possuem ótimo desempenho com relação à classificação de imagens. Isso sugere que a aplicação de redes convolucionais a saídas de imagem dos resultados de vibração é um caminho promissor, e inclusive já explorado, como mostrado na Seção 2 (LEE 2018).

Para esta seção, os dados de vibração coletados no ponto I serão utilizados para o cálculo dos espectrogramas formando imagens como mostradas na Seção 3. As imagens de espectrograma geradas têm o tamanho de 128×128, e assim a camada de entrada da nossa rede convolucional também o terá. Como os dados de cada pixel correspondem a apenas um valor, o espectrograma passado para a rede terá apenas uma dimensão de profundidade e está mostrado na Figura 35.

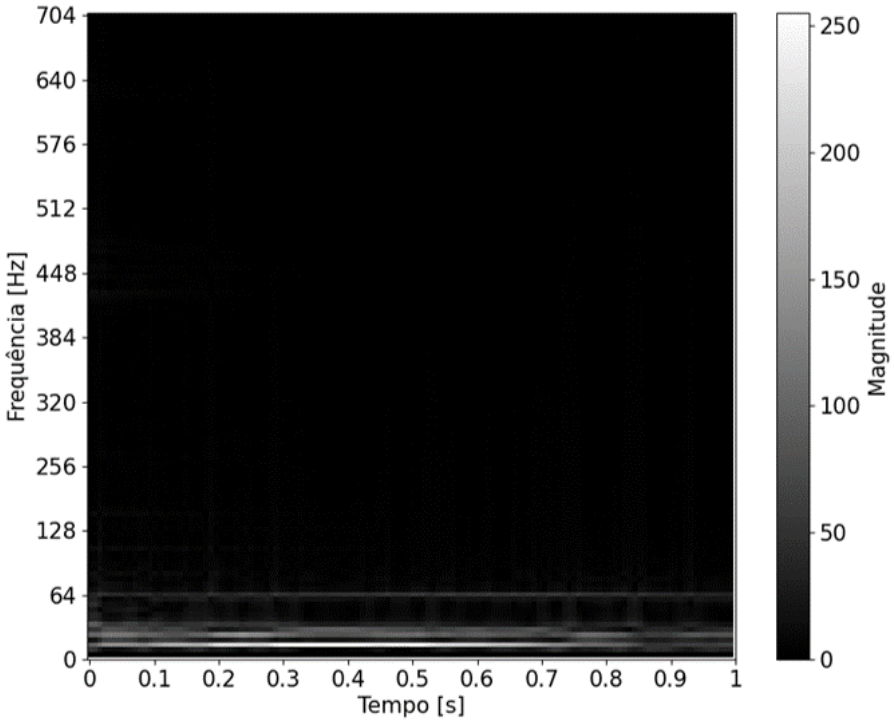

Figura 35 - Espectrograma normalizado passado como estrada para a rede neural convolucional, com tamanho de 128 x 128 pixels.

A arquitetura utilizada para os próximos testes foi baseada na rede *LeNet - 5* (LECUN 1998), chamada aqui de *LeNet - 5 Modificada*, e possui a arquitetura descrita na Tabela 14, com

2 camadas convolucionais, intercaladas com camadas de *pooling*, e saída de 4 parâmetros, cada um correspondendo à probabilidade de o dado de entrada pertencer a uma das 4 classes (teclas). O modelo foi treinado utilizando o minimizador *RMSprop* disponível na biblioteca *keras* (KERAS 2022), com taxa de aprendizagem de 1e-3, 100 épocas e tamanho de *batch* de 64 amostras. As acurácias obtidas para os dois conjuntos de teclas estão mostradas na Tabela 15 e a função perda durante a aprendizagem, para caso das teclas distantes está mostrada na Figura 36.

Como é possível observar na Tabela 15, a acurácia obtida chega mais próxima a mais bem conseguida nos testes com as redes totalmente conectadas. Além disso, a nova acurácia supera em quase 10%, passando de 65% para 74%, os resultados obtidos pelo espectrograma na seção 4.1.3, um aumento de 9% que pode também ajudar na identificação de teclas sobrepostas. Claramente, foi possível treinar filtros que destacam detalhes importantes para a rede e fazem com que a acurácia cresça com relação aos resultados anteriores. Esse fato é muito importante para a solução do problema e mostra que a utilização de imagens é uma boa frente a ser explorada.

Tabela 14 - Arquitetura da rede *LeNet-5* modificada utilizada.

LeNet Modificada	
Camada	**Configuração**
Entrada	128 x 128 x1 pixels
Convolucional	5 x 5 x 6 pixels
Pooling	2 x 2 pixels
Convolucional	5 x 5 x 16 pixels
Pooling	2 x 2 pixels
Densa / *sigmoid*	120 neurônios
Densa / *sigmoid*	84 neurônios
Saída	4 classes

Tabela 15 - Acurácia encontrada utilizando a rede *LeNet-5* modificada para ambos os conjuntos de dados.

Teclas próximas		**Teclas distantes**	
Treinamento	**Teste**	**Treinamento**	**Teste**
67%	65%	78%	74%

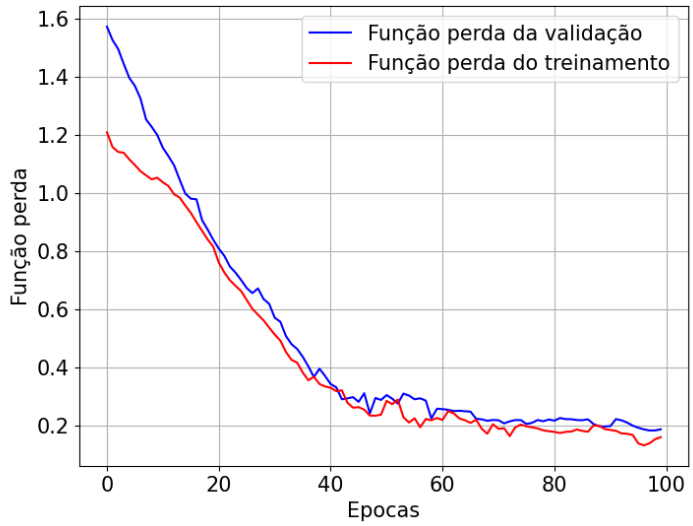

Figura 36 - Curva de aprendizado para o caso de teclas próximas da rede neural *LeNet*-5 modificada.

Um ponto importante a ser salientado é que o número de parâmetros treináveis do modelo da rede CNN é muito maior que as redes totalmente conectadas utilizadas. Para comparação, as redes com 24 camada utilizadas com tipo totalmente conectado possuíam um total de aproximadamente 9 milhões de parâmetros a serem treinados. Já nesta rede convolucional o número de parâmetros é de mais de 71 milhões, o que representa quase 8 vezes o número de parâmetros da rede totalmente conectada usada. Sendo assim, o processo de treinamento da rede convolucional é mais caro computacionalmente, limitando os algoritmos de otimização que podem ser aplicados para melhorar a arquitetura da rede.

Ainda visando explorar mais a capacidade das redes convolucionais, outro teste foi executado utilizando o espectrograma das 3 direções disponíveis. Os espectrogramas foram empilhados formando uma imagem com 3 escalas de cores (RGB), onde cada escala representa

o espectro de aceleração em uma das direções, como mostrado na Figura 37. A rede utilizada foi a mesma, com apenas uma mudança na camada de entrada de $128 \times 128 \times 1$ para $128 \times 128 \times 3$ pixels.

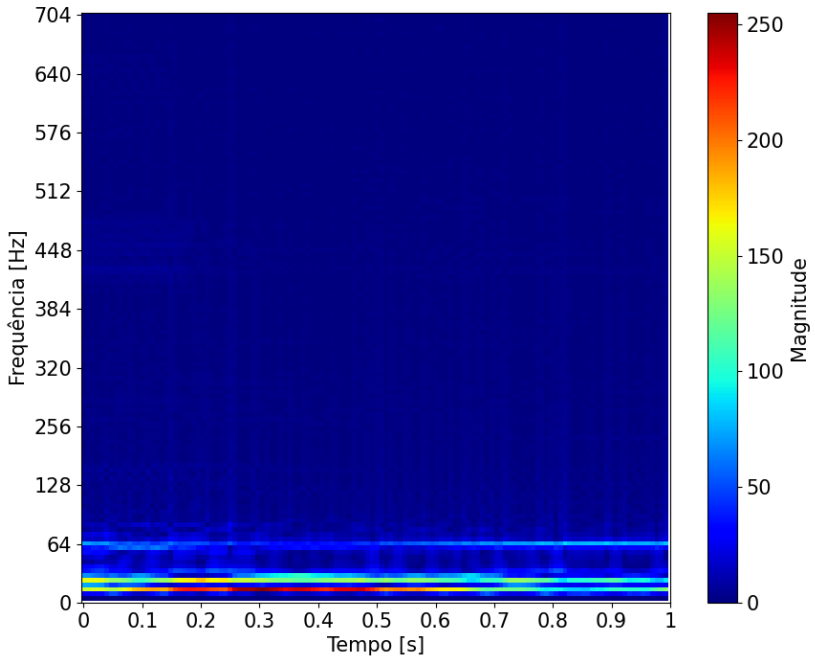

Figura 37 - Espectrograma das 3 direções disponíveis, X, Y, e Z, onde cada cor da escala RGB foi alimentada com um espectrograma de uma direção única, formando a imagem colorida utilizada como entrada para a rede neural.

A Tabela 16 compara os resultados para os dois conjuntos de dados e para as duas abordagens utilizadas. Nota-se que não há aumento da acurácia significativo da rede, com a maior entrada de dados, resultado também observado no caso das redes totalmente conectadas.

Tabela 16 - Comparação entre os resultados de acurácia das redes treinadas com sinais em 1 direção e 3 direções.

Conjunto de dados	**Teclas próximas**		**Teclas distantes**	
	Treinamento	**Teste**	**Treinamento**	**Teste**
1D	67%	65%	78%	74%
3D	65%	64%	77%	75%

4.3.2 Redes convolucionais e imagens da transformada de wavelet

Outro modo bastante utilizado de análise de sinais de vibração mecânica e acústica utilizando imagens 2D é a CWT, que inclusive é também utilizada com redes do tipo convolucional. A CWT é uma ferramenta para avaliar sinais em que há diferente detalhes variando de tamanho em escalas de tempo, como mencionado na seção 3.3. Para a calcular a CWT, funções de base, ou filtros, são aplicados ao sinal temporal, processo semelhante à filtragem da rede convolucional. A grande vantagem do uso da transformada wavelet é sua característica de multi-resolução, devido ao processo de aplicação das funções base, como explicado e mostrado na seção 3.3.

As funções de janela utilizadas pela transformada wavelet são variadas no tamanho temporal para que realçem os diferentes detalhes do sinal. A biblioteca *python* utilizada para o cálculo das transformadas wavelet foi a *pyWavelets*. A função base utilizada foi a *mexh* com 128 escalas, para que a imagem final tenha o tamanho de 128 pixels esperados pela rede. A Figura 38 mostra o dado resultante da aplicação da transformada wavelet em um sinal da tecla ‘Z’. Após ser normalizado, o padrão de cores foi escolhido em escalas de cinza pois é o mesmo que será dado de entrada para a rede convolucional.

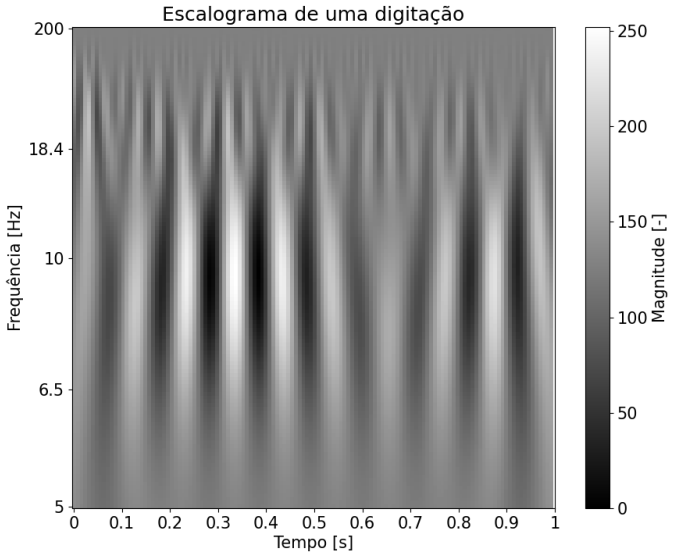

Figura 38 – Escalograma da CWT normalizada entre 0 e 256, apresentada em escalas de cinza.

A rede utilizada para identificação das imagens de wavelet foi a mesma utilizada na seção anterior, com entrada de imagem com 128×128 pixels, e os parâmetros de treinamento também foram mantidos os mesmos. A Tabela 17 compila os dados de acurácia para as redes treinadas e compara com os dados já obtidos para outros casos (totalmente conectada treinada com os dados de extração de detalhes). Percebe-se que há uma paridade entre os resultados obtidos pela rede totalmente conectada em conjunto com a extração de detalhes e a rede convolucional em conjunto com o escalograma da CWT. Mesmo assim, esses resultados mostram que a CWT parece superior ao espectrograma para destacar detalhes do sinal temporal, quando avaliada com uma rede do tipo convolucional. Outras funções base e números de escalas serão analisados nas próximas seções.

Tabela 17 - Comparação de resultados de alguns casos apresentados até o momento.

Tipo de rede neural	**Teclas próximas**		**Teclas distantes**	
	Treinamento	**Teste**	**Treinamento**	**Teste**
Fully connected	84%	80%	95%	92%
CNN Espec.	67%	65%	78%	74%
CNN Wavelet	71%	70%	82%	80%

4.3.3 Transferência de aprendizado

Como mostrado no início da seção, as redes neurais do tipo convolucionais utilizam filtros de imagens para destacar e ocultar detalhes das imagens de entrada. É de se esperar que o treinamento da identificação de detalhes em imagens produza filtros semelhantes em problemas diferentes. As imagens dos filtros mostrada na Figura 39 ilustram exemplos de filtros específicos para detalhes verticais, horizontais, entre outros. Ela é retirada de uma das camadas convolucionais da rede mostrada em MATLAB CNN (2023). Sendo assim, parece factível que os filtros treinados para uma rede possam ser usados em outras redes de identificação de imagens. O uso de filtros de uma rede neural convolucional para outras finalidades é comum e tem a vantagem de poupar tempo de treinamento de uma nova rede neural, além de ser possível utilizar filtros treinados com milhares de imagens para um problema que não possui tantos dados. Além disso, como apenas uma pequena porção da rede precisa ser retreinada, o custo

computacional para o treinamento deste tipo de rede é muito menor do que para o treinamento de uma rede a partir do início. Essa técnica é chamada de transferência de aprendizado, ou em inglês *transfer learning.*

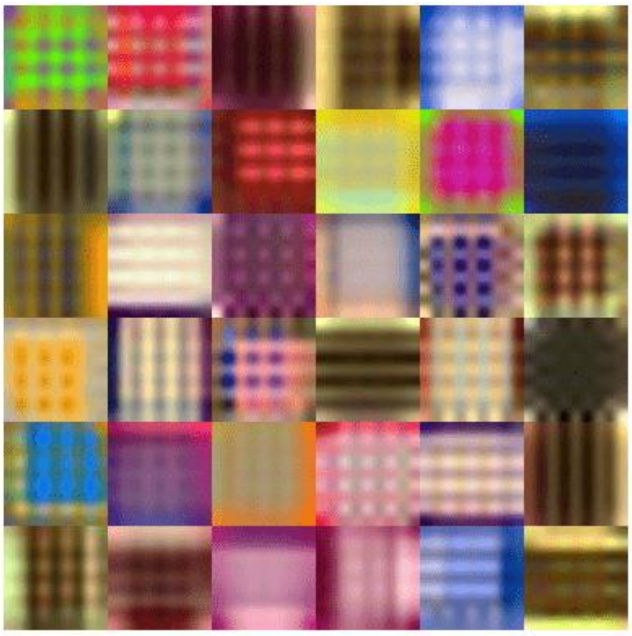

Figura 39 - Filtros de uma camada convolucional, mostrando o aprendizado de detalhes de diferentes formas e texturas (MATLAB CNN 2023).

Para o problema em questão, fica evidente que o reaproveitamento de uma rede neural, já com arquitetura otimizada e treinada com milhões de imagens, pode ajudar no aumento de acurácia para a identificação de teclas digitadas. Em 2010 foi criado o ILSVRC (*ImageNet Large Scale Virtual Recognition Challenge*) (ILSVRC 2022) uma competição anual que, além de premiar as melhores rede neurais para detecção de objetos, acumula imagens em seu banco, de mais de 1000 diferentes classes para treinamento dessas redes

Nesse projeto, foi aplicado o método de *transfer learning* utilizando algumas das redes vencedoras do ILSVRC para o problema de identificação de teclas. Obviamente, as redes do ILSVRC não foram treinadas para identificação de diferentes espectrogramas ou imagens da transformada wavelet, mas sim para a classificação de objetos, animais e paisagens. Apesar disso, os milhares de filtros finamente otimizados por esses treinamentos podem conter as informações necessárias para identificar os padrões entre diferentes dados de sinais de vibração.

Nesse problema, utilizamos primeiramente a rede *ResNet* (*residual net*), vencedora do ILSVRC de 2015 e que apresentou acurácia de 96.4% para identificação de imagens dentro de 1000 classes disponibilizadas. Os parâmetros da rede *ResNet50* estão disponíveis e podem ser facilmente baixados utilizando a biblioteca *keras*. A rede *ResNet* é composta por 152 camadas convolucionais organizadas de forma que a profundidade dos filtros cresce até 512 camadas de profundidade, culminando em uma camada totalmente conectada que ativa as saídas das classes corretas. A saída, por sua vez, possui as 1000 classes já mencionadas, e a imagem de entrada possui 128×128 pixels.

Para nosso problema, os parâmetros de cálculo do espectrograma e da CWT foram cuidadosamente escolhidos na seção 3.3, para que a imagem resultante tenha o tamanho esperado pela rede *ResNet*. Além disso, os pesos e bias das camadas convolucionais foram congelados e apenas a camada totalmente conectada foi retreinada, agora com apenas quatro saídas, representando as quatro classes de teclas para cada um dos conjuntos de dados.

O modelo foi retreinado com o otimizador *RMSProp*, com taxa de aprendizagem de 2e-5, utilizando 200 épocas e tamanho de *batch* de 64 amostras. Os resultados obtidos pelo retreinamento da rede utilizando os dois conjuntos de dados e as imagens de espectrograma e transformada de wavelet estão compilados na Tabela 18. Como se pode notar, o algoritmo de transferência de aprendizado superou todas as abordagens anteriores, aumentando a acurácia da rede para 90% no caso das teclas distantes com a imagem da CWT e para 83% no caso do conjunto de teclas próximas. Além disso, mesmo para o pior caso, os resultados obtidos se aproximam dos melhores resultados já apresentados, vide Seção 4.1.

Tabela 18 - Comparação de acurácia para treinamento utilizando *transfer learning* e entradas de espectrograma e escalograma (transformada de wavelet).

Entrada	**Teclas próximas**		**Teclas distantes**	
	Treinamento	**Teste**	**Treinamento**	**Teste**
Espectrograma	80%	78%	90%	88%
Wavelet	83%	83%	92%	90%

Os resultados animadores encorajam a continuação do estudo de identificação de sinais de vibração utilizando redes convolucionais e o método de transferência de aprendizagem. Na

próxima seção, tentaremos otimizar a rede mudando alguns parâmetros disponíveis e ganhando mais sensibilidade da influência de cada um deles.

4.3.4 Otimização dos parâmetros de treinamento com algoritmo genético

Otimização utilizando algoritmo genético é uma técnica de busca que utiliza conceitos da teoria da evolução para encontrar soluções para problemas complexos (GOLDBERG 1989). O algoritmo genético trabalha com uma população de soluções candidatas que são representadas por cromossomos e evolui essas soluções ao longo de várias gerações. Cada indivíduo da população é avaliado em relação a um critério de aptidão e a seleção, cruzamento e mutação são aplicados para gerar uma nova geração de soluções. Esse processo de seleção natural tende a favorecer as soluções mais aptas, de forma que a cada geração as soluções tendem a ser melhores em relação ao critério de aptidão, ou função custo. O algoritmo genético é amplamente utilizado em problemas de otimização em que não é possível encontrar uma solução exata em tempo hábil e pode ser adaptado para problemas de diversas áreas, como engenharia. O ponto negativo da utilização de algoritmos genéticos é que a melhor condição encontrada, não necessariamente é a melhor condição possível, e não é possível dizer se a resposta atual é a melhor sem a análise de todos os possíveis casos e combinações de genes (SASTRY 2005).

O processo evolutivo mencionado acima é alcançado com algumas modificações nos melhores indivíduos de cada iteração, onde podemos citar o *crossover* e a mutação, que serão explicados a seguir. Além disso, o algoritmo é escrito para que o melhor indivíduo de uma geração seja passado para a geração seguinte. Essa propriedade leva o nome de elitismo e garante que a melhor solução encontrada até um determinado momento não se perca nas iterações posteriores. A definição da aptidão de um indivíduo em relação ao outro é dada pela sua função custo, que no nosso caso será a acurácia final da rede treinada com os parâmetros daquele indivíduo, ou seja, um indivíduo que os parâmetros resultem em uma acurácia da detecção de teclas de 80% é considerado mais apto do que um que resulte em acurácia de 60%.

O *crossover* é responsável por gerar novas soluções através da combinação de informações de dois indivíduos diferentes. O processo de crossover consiste em selecionar dois indivíduos pais na população e combinar suas informações genéticas para gerar um ou mais

indivíduos filhos. O ponto de crossover é definido aleatoriamente e determina o local onde as informações genéticas dos pais são combinadas. Após a geração dos filhos, a população é atualizada com os novos indivíduos para análise da próxima geração. O crossover é uma ferramenta que permite a geração de soluções que não estavam presentes na população inicial, além de permitir a exploração de regiões do espaço de busca que não foram exploradas anteriormente. Em nosso caso, os indivíduos pais são escolhidos pelo *ranking*, ou seja, os indivíduos com melhores acurácia têm mais probabilidade de serem escolhidos como pais da próxima geração.

Já a mutação é outra ferramenta utilizada para se encontrar uma solução ótima. Ela é responsável por introduzir novas informações genéticas na população, permitindo a exploração de novas regiões do espaço de busca. Ela é um operador crucial que garante a diversidade da população e ajuda a evitar a convergência prematura para soluções não ótimas. O processo de mutação consiste em alterar aleatoriamente o valor de um gene em um indivíduo da população. A probabilidade de mutação é um parâmetro importante que determina a frequência com que a mutação ocorre. Se a probabilidade de mutação for muito baixa, a população pode ficar presa em ótimos locais e não explorar outras regiões do espaço de busca. Por outro lado, se a probabilidade de mutação for muito alta, a população pode se tornar muito diversificada e perder informações valiosas. Em nosso caso, utilizamos probabilidade de mutação de 35%, que é da ordem de valores recorrentes em diferentes trabalhos com algoritmo genético.

Neste contexto, utilizamos um algoritmo genético para otimizar alguns parâmetros da nossa rede neural e treinamento dela. Nossa população foi formada por indivíduos com seis genes, ou seja, seis parâmetros que serão utilizados. Esses parâmetros são: dados de entrada, modelo da rede neural (Tabela 19), tamanho do batch de treinamento, taxa de aprendizagem, otimizador do treinamento e função de ativação da camada. Para os modelos de rede utilizados, além dos modelos já mencionados anteriormente (*LeNet* e *ResNet50*) foi adicionado também o modelo *AlexNet*, que é uma evolução da rede *LeNet* (KRIZHEVSKY 2012).

Além da adição da rede *AlexNet*, para o gene “otimizador” foi adicionado o algoritmo *SDG* (*Stochastic gradient descente*) que é o algoritmo mais simples utilizado para o treinamento de redes neurais. A Tabela 20 compila os possíveis dados que esses genes podem conter. Vale ressaltar que todos os indivíduos foram treinados com 200 épocas e divisão do conjunto de dados de treinamentos, validação e testes, de respectivamente, 75% e 15% e 10%, para os dois conjuntos de teclas. Para o algoritmo utilizamos a biblioteca *pyGAD* (PYGAD 2022).

Tabela 19 - Tipos de redes neurais utilizadas como gene no algoritmo genético.

Tipos de redes utilizadas para evolução
LeNet Modificada
AlexNet Modificada
Totalmente conectada (4.3.1)
ResNet50 (*transfer learning*)

Tabela 20 Parâmetros utilizados para formação dos diferentes genes.

Parâmetro do gene	**Possível Valor / Configuração**
Entrada	Espectrograma, Wavelet (*mexh*), Wavelet (*gaus5*)
Rede neural	Vide Tabela 19
Tamanho do *Batch*	16, 32, 64, 128 imagens
Taxa de aprendizado	1e-3, 1e-4, 1e-5
Otimizador	SDG, *Adam*, *RMSProp*
Função de ativação	Sigmoid, ReLu, Tangente Hiperbólica

A combinação de todos os genes entre si resultaria em 1080 redes a serem treinadas, mas com o algoritmo genético, espera chegar-se em bons resultados com apenas 200 treinamentos, sendo 20 gerações com 10 indivíduos em cada uma. A respeito das configurações para a otimização, a Tabela 21 lista as configurações usadas para o algoritmo genético, como elitismo, probabilidade e tipo de mutação e tipo de crossover.

Tabela 21 - Parâmetros utilizados na configuração do algoritmo genético.

Parâmetro	**Valor / Configuração**
Crossover	*SinglePoint*
Seleção de pais	*Ranking*
Probabilidade de Mutação	3.5%
Tipo de mutação	Aleatória
Elitismo	Melhor Indivíduo

Na Tabela 22 são apresentados os melhores indivíduos obtidos, e na Figura 40 o gráfico da acurácia do melhor indivíduo por geração, mostrando a evolução da resposta encontrada, que chegaram a 96% e 86% no caso de teclas distantes e próximas, respectivamente. Pela análise dos melhores indivíduos, fica claro que o método de transferência de aprendizagem é superior do que aos de treinamento completo dos outros modelos. O treinamento da rede *ResNet50* com o conjunto de dados *Imagenet*, que possui milhares de imagens, contribui para um reconhecimento de detalhes mais aprimorado do que os outros métodos dedicados testados. A concordância do parâmetro de wavelet *gaus5* e do otimizador *RMSProp* nos dois melhores indivíduos também mostra a superioridade destes métodos com relação aos outros avaliados.

Tabela 22 - Parâmetros dos melhores indivíduos encontrados na otimização genética para ambos os conjuntos de dados.

Parâmetro do gene	**Melhores Indivíduos**	
	Teclas Próximas	**Teclas Distantes**
Entrada	Wavelet (*gaus5*)	Wavelet (*gaus5*)
Rede neural	*CNN Transfer Learning*	*CNN Transfer Learning*
Tamanho do *Batch*	32 imagens	64 imagens
Taxa de aprendizado	1e-4	1e-3
Otimizador	*Adam*	*Adam*
Função de ativação	*Sigmoid*	*Sigmoid*

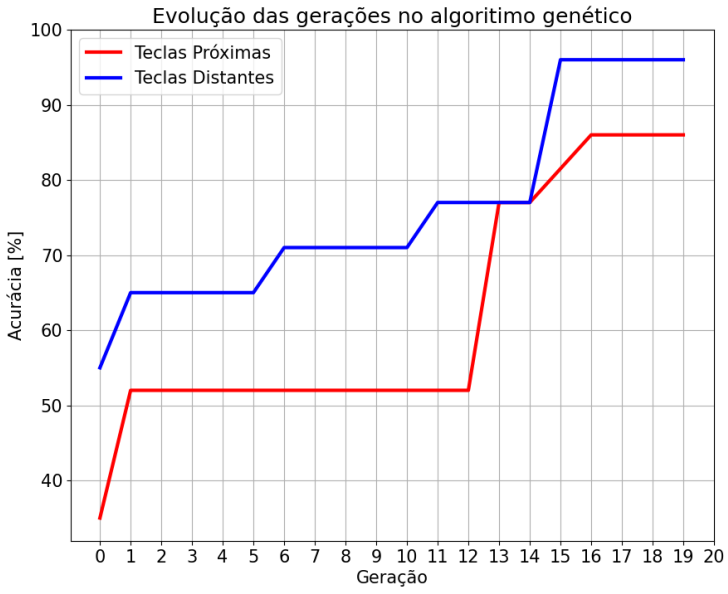

Figura 40 - Evolução do melhor individuo por geração ao longo da otimização.

4.3.5 Otimização dos parâmetros da rede com Optuna

Outra tentativa de se buscar melhores configurações para nossa rede neural, foi a utilização da biblioteca Optuna (OPTUNA 2022). Optuna é uma biblioteca de otimização de hiperparâmetros em Python que usa algoritmos de busca para encontrar os melhores parâmetros para um determinado modelo. Ela é capaz de otimizar hiperparâmetros de modelos de aprendizado de máquina, *deep learning*, processamento de linguagem natural e outras áreas. A biblioteca usa uma abordagem baseada em tentativas, em que cada tentativa de execução é avaliada de acordo com um objetivo definido, como acurácia ou erro quadrático médio. Optuna também oferece recursos como a possibilidade de definir restrições de parâmetros e otimização multiobjetivo. Com Optuna, é possível realizar experimentos de forma automatizada e obter resultados mais precisos em um menor tempo de processamento em comparação com a programação do zero do algoritmo. Em nossa tentativa, os parâmetros mostrados na Tabela 23 foram avaliados pata tentar buscar uma melhor acurácia da rede.

Tabela 23 - Parâmetros utilizados para a otimização utilizando a biblioteca Optuna.

Parâmetro do gene	**Intervalo de busca**
Dropout	0 – 0,5
Número de camadas escondidas	1 – 4
Neurônios por camadas escondidas	16 - 256
Learning rate	1e-5 – 1e-1

O hiperparâmetro *learning rate* controla o tamanho do passo que o algoritmo de otimização dará em cada iteração durante o processo de ajuste dos pesos da rede. Em outras palavras, a taxa de aprendizagem determina o quão rápido a rede neural irá aprender e ajustar seus pesos em resposta aos erros de previsão durante o treinamento. Se a taxa de aprendizagem for muito alta, a rede pode saltar sobre os mínimos locais e perder a convergência para a solução ótima. Por outro lado, se a taxa de aprendizagem for muito baixa, o processo de treinamento pode levar muito tempo para convergir para uma solução satisfatória.

Já o número de camadas e a quantidade de neurônios por camada foram escolhidos para a otimização, uma vez que um número inadequado de camadas ou neurônios por camadas pode levar a uma rede neural muito simples e incapaz de capturar a complexidade dos dados. Por outro lado, uma rede neural com muitas camadas e neurônios pode ser muito complexa e levar a problemas de *overfitting*.

Apesar dos esforços, a otimização destes parâmetros não resultou em uma melhor acurácia para a rede, mantendo-se as configurações anteriores descritas na Seção 4.3.4. A hipótese levantada é que a quantidade limitada de dados de treinamento faz com que não seja possível uma boa validação de todos os parâmetros e que seja necessário um maior conjunto de dados para uma otimização bem-sucedida.

4.3.6 Avaliação da influência da posição de medida

Considerando a melhora do resultado com as sucessivas otimizações mostradas acima, resolveu-se, novamente, utilizar a configuração otimizada da rede neural convolucional mostrada acima (Tabela 22) para avaliar a influência da posição de medida do acelerômetro nos resultados de acurácia da rede. O objetivo deste experimento é avaliar uma possível melhora na acurácia das redes, considerando uma medida simples como a mudança de localização do acelerômetro durante a medida dos sinais. Para isso, os escalogramas dos sinais da direção Y das posições de I a IV, mostrados na seção 3, foram utilizando para treinamento das redes neurais convolucionais mostradas na tabela 19. Assim como nos testes anteriores, os conjuntos de dados foram divididos em 3 partes: 75% para treinamento, 15% para validação e 10% para dados de teste, e o número de épocas foi de 200. Os resultados obtidos são mostrados na Tabela 24 a seguir.

Tabela 24 - Resultado de acurácia da rede convolucional quando avaliada utilizando a técnica de *transfer learning*, com entrada como os escalogramas, para os diferentes pontos de medida do acelerômetro.

Posição do sensor	**Teclas próximas**		**Teclas distantes**	
	Treinamento	**Teste**	**Treinamento**	**Teste**
I	91%	86%	99%	96%
II	86%	88%	100%	92%
III	87%	88%	97%	96%
IV	97%	96%	99%	92%

Podemos identificar que, assim como na seção 4.1.4, não houve melhora ou piora significativa na acurácia das redes com as diferentes posições do acelerômetro, mantendo todos os resultados próximos aos valores de referência da posição I. Um dos resultados que chama atenção é que, diferentemente da seção 4.1.4, a posição IV, que está localizada no ponto de simetria entre as teclas “P” e “1”, desta vez não apresenta resultados inferiores as outras posições, revelando que a rede proposta foi capaz de distinguir detalhes ou padrões não evidenciados na análise com extração de detalhes, naquela ocasião.

Um ponto de similaridade com a seção 4.1.4 é a melhora na acurácia de classificação para o conjunto de dados de teclas próximas quando comparamos o ponto I e II. Quando a rede é treinada com os dados da posição II, ou seja, quando o acelerômetro é posicionado mais próximo do canto inferior esquerdo do teclado, é possível notar um salto de 2% na acurácia do conjunto em questão. Novamente, este resultado é esperado, dada a menor distância de propagação do sinal da digitação até o acelerômetro, resultando em menor adição de ruído e maior amplitude de aquisição.

5 SUPERPOSIÇÃO DE SINAIS

Até o momento, apenas foram considerados dados de medidas em que uma única tecla foi digitada. Em um caso real de digitação, várias teclas são digitadas em um curto espaço de tempo, o que pode fazer com que parte do sinal de uma tecla lido pelo acelerômetro seja superposto com parte do sinal da tecla seguinte. Segundo DHAKAL (2018), a velocidade média de digitação é por volta de 41 palavras por minuto, o que pode ser traduzido em aproximadamente 246 teclas por minuto (com uma média de 5 letras por palavra e mais uma tecla de espaço entre as palavras), ou seja, 4.1 toques por segundo, resultando em um tempo de 0.24s entre cada digitação, o que é menor que o tempo de 1s em que o sinal de cada tecla estava sendo medido nos testes descritos nas seções anteriores. Assim, em torno de 76% do sinal de aceleração de uma tecla numa sequência de digitação, considerando 1s de amostragem, estará superposto com o sinal da tecla seguinte.

Com os sinais não-superpostos, a melhor acurácia obtida na rede foi de 96% (teclas distantes), quando treinada e avaliada com sinais de 1s, no contexto da rede neural convolucional e dados de entrada na forma de escalogramas da CWT. A fim de se estudar o problema de superposição, alguns testes e propostas são feitos no sentido de se entender e melhorar a acurácia das redes para o problema de superposição. Começaremos avaliando a acurácia das redes já mostradas quando avaliadas com sinais superpostos, e em seguida tentaremos limitar a quantidade de dados passados para a rede para tentar diminuir a influência da superposição. Após isso, mostraremos alguns casos mais próximos do caso de uma digitação real e discutiremos como a rede se comporta.

5.1 Superposição de sinais simples

Como uma primeira tentativa de avaliar os casos de superposição de sinais, utilizamos as duas melhores redes já treinadas na seção anterior, para tentar identificar casos em que duas ou mais teclas estão superpostas. Para a criação de novos sinais superpostos, usou-se a hipótese de que é possível aproximar o sistema como um sistema linear e invariante no tempo (*LTI – Linear and time invariant*), que possui a propriedade da superposição, em que a resposta do sistema

dada uma entrada $a + b$ é igual à resposta do sistema dada uma entrada a somado à resposta do sistema dada uma entrada b. Sendo assim, superpusemos artificialmente os sinais já coletados, somando suas respostas temporais não filtradas das teclas que queríamos superpor. Esta metodologia é conveniente e necessária uma vez que para o caso do teclado completo, fazer a aquisição de todas as teclas superpostas umas com as outras iria resultar em um número impraticável de dados a serem coletados. Mesmo em nosso caso simplificado de 4 teclas, o número de aquisições para a superposição de 2 a 2 teclas aumentaria de 4 aquisições simples para 24 aquisições compostas mais 4 simples, e ao passo em que o número de teclas, n, a serem analisadas aumenta, o número de aquisições aumenta com o fatorial de n.

A Figura 41 exemplifica os dois tipos de sinais que serão utilizados. O gráfico *superposição A* mostra uma superposição de sinais onde a tecla de interesse foi a primeira digitada. Entretanto, devido ao longo tempo de aquisição (1s), outra tecla foi digitada enquanto o sinal da primeira ainda estava sendo medido, o que somou os sinais do instante $t = 0.33s$ em diante. Já no gráfico de *superposição B*, o sinal que estamos tentando analisar é justamente o da segunda tecla digitada, como se estivéssemos tentando encontrar a sequência de teclas digitadas. O sinal da segunda tecla já está somado com o sinal das teclas anteriores, o que pode modificar a forma como a rede neural interpreta esse dado. A porção sombreada mostra a parte do sinal que vai ser desconsiderada de cada gráfico, levando em conta o tamanho total do sinal de 1s.

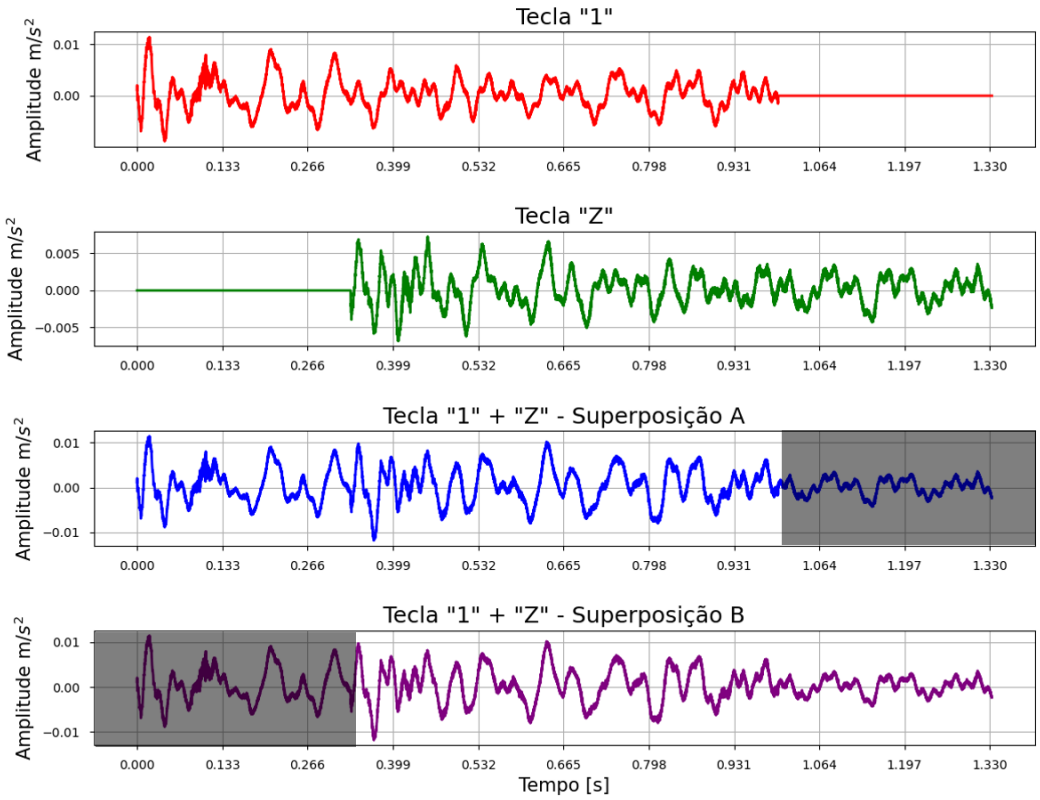

Figura 41 - Exemplos das diferentes superposições que serão utilizadas na Seção 5.

Ao todo, 1000 sinais foram criados para cada conjunto de 4 teclas, próximas e distantes, considerando a posição I do acelerômetro, e eles foram dados como entrada para as redes treinadas na Seção 4. Os resultados estão resumidos na Tabela 25. Fica claro que essa mudança nos dados de entrada é suficiente para confundir o algoritmo da rede neural treinada com teclas individuais, fazendo com que a acurácia caia drasticamente. Nas duas redes utilizadas (Seção 4.1.4 e Seção 4.3.4), o resultado obtido foi pelo menos 30% menor do que a acurácia obtida para as teclas individuais na Seção 4. Também não parece haver uma relação entre os diferentes tipos de abordagem utilizadas e a piora na acurácia de classificação dos dados. Nas próximas subseções, tentaremos utilizar diferentes propostas para melhorar a acurácia da rede quando exposta a dados superpostos.

Tabela 25 – Acurácia das redes para entrada de sinais superpostos na condição A e B com duração de 1 segundo mostrados na Figura 41.

Rede Neural	Dado de Entrada	Teclas Próximas	Teclas Distantes
Tot. Conectada	Extr. De detalhes	39%	57%
Convolucional	Escalograma	42%	54%

5.2 Limitação do tamanho temporal de dados para treinamento

Como uma tentativa de se melhorar a acurácia das redes para os casos de superposição de sinais, foi proposto treinar a rede com dados temporais menores, isto é, sinais de aceleração coletados em um espaço de tempo mais curto. Com dados menores há uma menor chance de o sinal a ser analisado pela rede esteja contaminado pelo sinal de outra tecla posterior, além de estarmos utilizando a região do sinal com maior energia (durante e logo após o pico de aceleração). Apesar disso, com dados menores também há a perda de informação que está presente na continuação do sinal. Nas análises da sessão 4, o sinal foi utilizado por completo para se avaliar a eficácia da classificação, utilizando toda a informação disponível, e dados os bons resultados nesta sessão novas complexidades são introduzidas ao problema, a fim de se simular o caso mais real. No caso dos sinais de 1s, levando em conta as informações apresentadas acima de tempo médio de digitação, até 4 teclas poderiam estar presentes no mesmo sinal de aceleração, em momentos temporais diferentes, dificultando a acurácia da rede.

Para esse experimento, utilizamos os mesmos dados medidos na Seção 3. Entretanto, ao invés de treinar a rede com dados de 1s, diminuímos esse dado de treinamento para tempos menores, entre 0.2s e 0.8s de comprimento, contados a partir do instante em que o sinal começou a ser aquisitado. A Figura 42 exemplifica os dados que foram analisados, mostrando como um dado de 1s foi cortado em dados menores. A área sombreada corresponde aos dados que foram descartados.

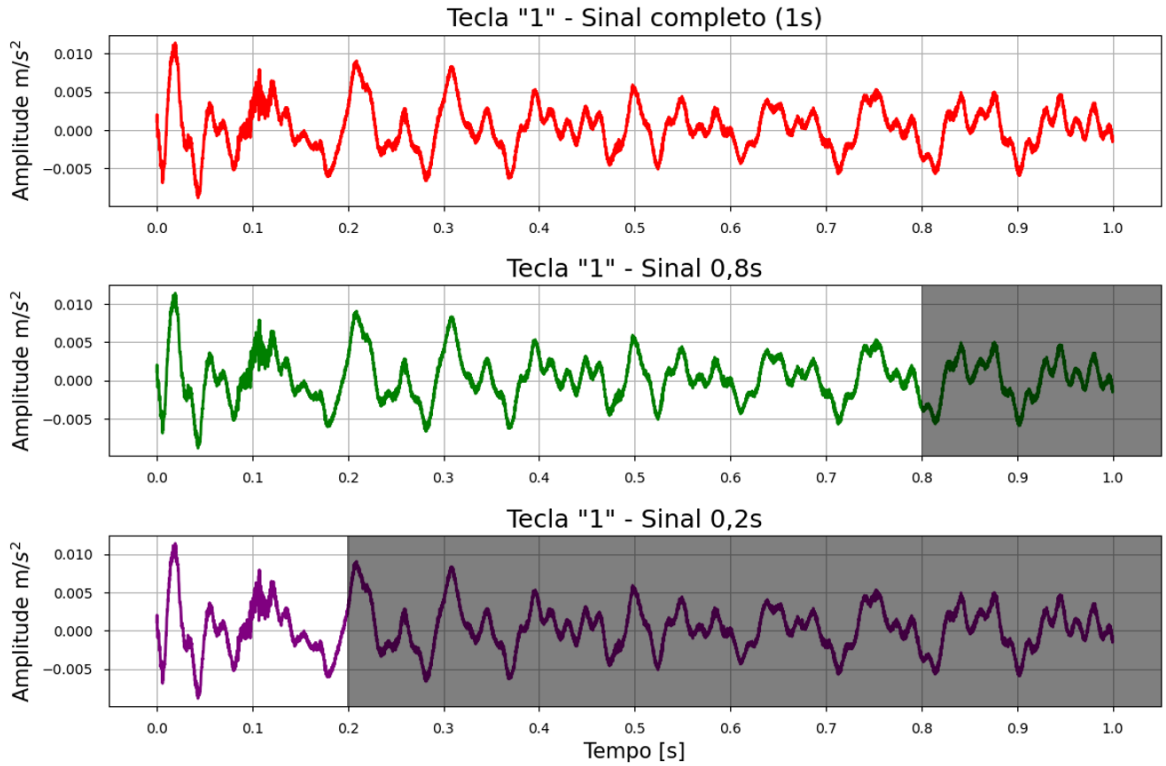

Figura 42 - Exemplo dos cortes em sinais de 1s para treinamento das redes com sinais menores.

Como citado, espera-se que, com dados menores, haja menos informação disponível para o aprendizado da rede. Entretanto, devido ao decaimento logarítmico da amplitude do sinal com o passar do tempo, e como estaremos sempre utilizando os sinais nos momentos mais próximos do impacto, a quantidade de informação perdida não deve ser um problema para a rede neural. Assim como a amplitude é maior nos instantes logo após o impacto, espera-se que esses instantes também contenham mais informações sobre a tecla do que momentos mais adiante, quando parte do sinal já foi dissipada pelo amortecimento intrínseco do sistema.

Novamente, utilizamos ambos os conjuntos de dados, de teclas próximas e teclas distantes, para o acelerômetro posicionado na configuração I, para poder avaliar as performances das redes de forma mais ampla. Os parâmetros obtidos pelo algoritmo genético na seção anterior foram utilizados para o treinamento das redes convolucionais com o *transfer learning* e os parâmetros encontrados para o treinamento das redes totalmente conectadas na seção 4.1.4 foram utilizados novamente nesta seção.

Os resultados dos treinamentos estão apresentados nas Tabelas 26 e 27. Fica notável que o tamanho dos dados possui influência na acurácia final da rede convolucional e na rede totalmente conectada, mesmo assim sendo de apenas 4%. Para ambos os conjuntos de dados, a

acurácia se mantém mesmo para dados menores. Podemos concluir que os sinais de tempos menores não são drasticamente mais difíceis de serem aprendidos, resultando em acurácias próximas às já obtidas anteriormente. Isso apoia a hipótese de que a parte mais significativa do sinal se concentra nos primeiros instantes do sinal, próximos ao impacto de pressionamento de tecla.

Tabela 26 - Acurácia encontrada com as redes convolucionais com treinamento de sinais menores.

Tamanho da entrada	**Teclas próximas**		**Teclas distantes**	
	Treinamento	**Teste**	**Treinamento**	**Teste**
0.8 segundos	81%	76%	87%	88%
0.6 segundos	83%	80%	90%	92%
0.4 segundos	84%	80%	89%	88%
0.2 segundos	83%	80%	95%	92%

Tabela 27 - Acurácia encontrada com as redes totalmente conectadas com treinamento de sinais menores.

Tamanho da entrada	**Teclas próximas**		**Teclas distantes**	
	Treinamento	**Teste**	**Treinamento**	**Teste**
0.8 segundos	87%	76%	97%	92%
0.6 segundos	80%	72%	93%	92%
0.4 segundos	81%	76%	99%	96%
0.2 segundos	85%	76%	99%	96%

Para avaliar a resposta da rede com os sinais sobrepostos, novamente utilizaremos os sinais como na seção 5.1, exceto que como precisamos de sinais com 0.2s de duração, não poderemos utilizar os sinais em que há uma segunda tecla sendo pressionada em $t = 0.33s$, pois o sinal seria igual a digitação de uma tecla simples. Sendo assim, para o caso das redes treinadas com sinais de 0,2s de duração, teremos apenas sinais superpostos como *superposição b*, da Figura 41. Fica claro que, quanto menor o dado necessário para a identificação da tecla, é menos provável que haja uma segunda digitação dentro do período de aquisição a ser considerado, ou seja, é muito mais provável que haja duas digitações dentro de um período de 1s do que em um período de 0.2s. A taxa de respostas corretas está nas Tabelas 28 e 29 a seguir.

Tabela 28 - Porcentagem de acertos da rede convolucional quando avaliadas com sinais superpostos.

Tamanho da entrada	**Teclas próximas**	**Teclas distantes**
0.8 segundos	42%	52%
0.6 segundos	45%	50%
0.4 segundos	46%	54%
0.2 segundos	67%	68%

Tabela 29 - Porcentagem de acertos da rede totalmente conectada quando avaliadas com sinais superpostos.

Tamanho da entrada	**Teclas próximas**	**Teclas distantes**
0.8 segundos	39%	49%
0.6 segundos	40%	45%
0.4 segundos	42%	51%
0.2 segundos	65%	74%

Apesar do decréscimo de acurácia quando os dados estão superpostos, as redes com 0.2s não perderam tanta eficiência quanto as redes mostradas na seção 5.1, o que revela um ponto positivo para a abordagem do treinamento com sinais mais curtos. Ambos as abordagens, quando treinadas com sinais de 0.4s, 0.6s e 0.8s obtiveram acurácia próxima as das redes da seção anterior, provavelmente em virtude de os dados de entrada terem casos em que a segunda tecla está dentro do período de aquisição da primeira tecla digitada.

Outro ponto que pode ser observado é a maior acurácia atingida pela rede totalmente conectada, no caso do conjunto de teclas distantes, mostrando vantagem sobre a análise da rede convolucional. Essa vantagem pode ser devida a uma melhor percepção, ou extração dos detalhes de alta frequência remanescente no sinal de 0.2s, com ajuda da janela *Symlet*, que destaca sinais de alta frequência dentro do dado.

Além dos experimentos acima, estudou-se a possibilidade de treinar a rede com sinais já superpostos. Alguns problemas foram ressaltados sendo eles: os sinais superpostos vão diminuir a informação utilizada para o aprendizado da rede para a tecla de interesse, uma vez o que o sinal puro será somado com outro da tecla anterior; treinar a rede com sinais já superpostos

exige garantir que não haverá *bias* nos dados de treinamento. Isto quer dizer que deve haver dados suficientes para superpor, por exemplo, a tecla “D”, com todas as outras teclas de interesse, para não condicionar o acerto desta com a digitação em sequência com alguma outra tecla do teclado. Sendo assim, foi priorizado o estudo de outros parâmetros em relação ao treinamento com sinais superpostos.

5.3 Identificação de senhas

Se aproximando mais do caso final de interesse, de identificação de uma senha, esta seção tem o objetivo de analisar o comportamento da rede quando avaliada com um dado mais próximo do de uma medida de uma digitação completa. Nesta seção, os sinais de digitação são compostos pela medida de aceleração de 4 teclas, cada uma amostrada num intervalo de 0,8s. Os últimos 0,4s do sinal de cada tecla são superpostos com os primeiros 0,4s do sinal da tecla seguinte. São consideradas as seguintes sequências de teclas:

$$vetor_1 = [\text{"1"}, \text{"}P\text{"}, \text{"}Z\text{"}, \text{":"}];$$
$$vetor_2 = [\text{"A"}, \text{"}S\text{"}, \text{"}Z\text{"}, \text{"}X\text{"}];$$

Essas sequências são selecionadas para representar os grupos de dados de teclas distantes ($vetor_1$) e próximas ($vetor_2$). O sinal final fica como o mostrado a seguir na Figura 43, e tem um tempo total de 2 segundos. Em um primeiro momento, decidiu-se avaliar o comportamento da rede quando os dados de entrada de têm tamanho de 0.2s e esse intervalo se inicia a cada 0.2s de aquisição. O comportamento ideal esperado da rede, seria de acusar picos de probabilidade para a classe da tecla correspondente, quando dentro do sinal analisado estão os momentos de digitação. Ou seja, no intervalo de tempo de 0s a 0.2s (tecla “1” digitada) da Figura 43 esperamos que a rede retorne uma alta probabilidade de que aquele sinal seja da tecla “1”, assim como o histograma ideal da Figura 44. Já no intervalo de 0.4s a 0.6s esperamos que a classe correspondente à tecla “Z” tenha uma probabilidade alta, e assim por diante nos intervalos de 0.8s a 1s (tecla “P” digitada) e de 1.2s a 1.4s (tecla “:” digitada).

Já para os intervalos em que não há uma digitação efetiva dentro do sinal de 0.2s analisado, esperamos probabilidades baixas para todas as classes assim como mostrado na Figura 44, para os intervalos de 0.2s a 0.4s, de 0.6s a 0.8s, de 1s a 1.2s e de 1.4s em diante. O comportamento próximo ao ideal da rede facilitaria a implementação do código para a aplicação real sem a necessidade da identificação do momento exato de digitação de uma tecla

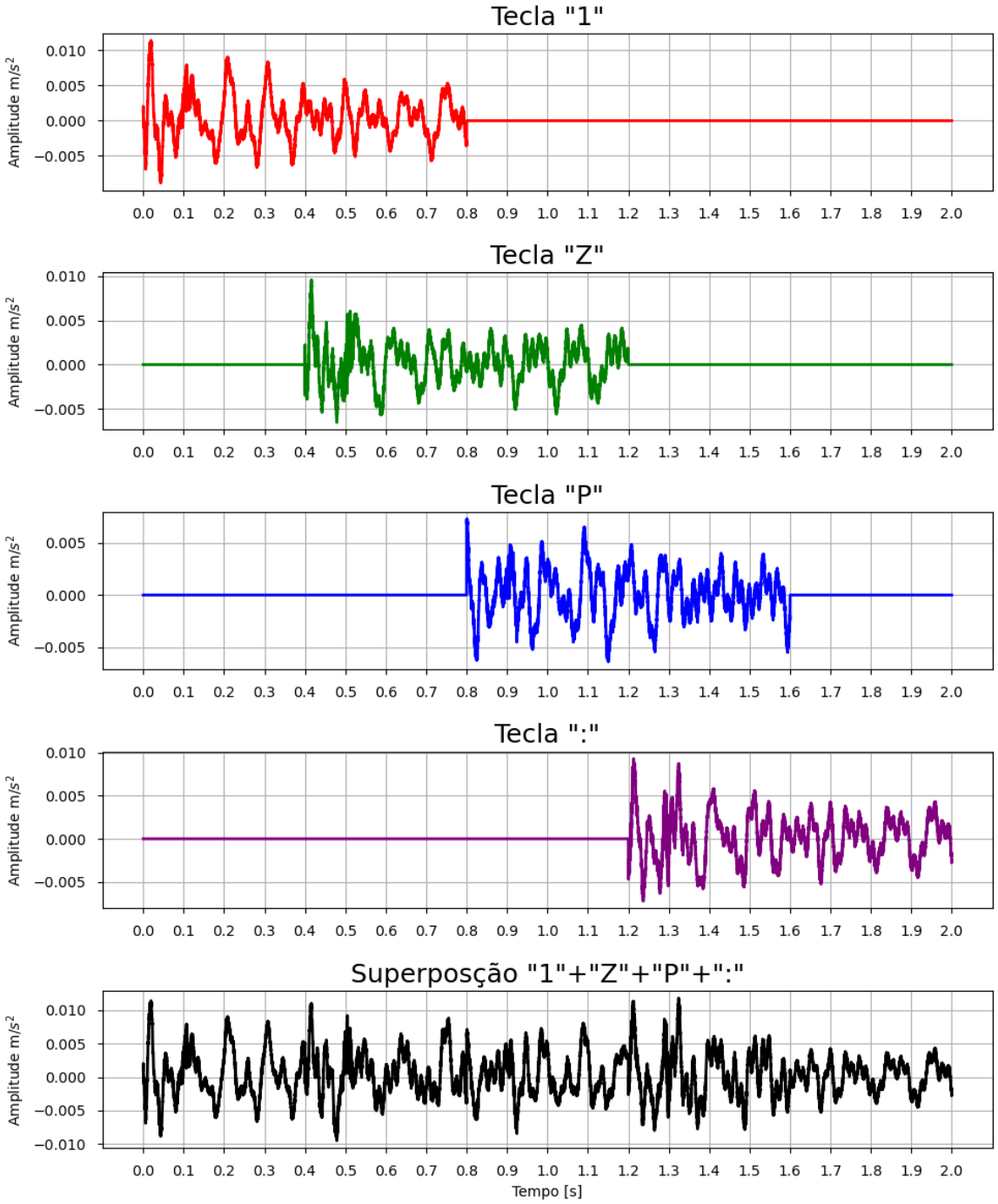

Figura 43 – Sinal completo para as teclas distantes com superposição de 0,4s.

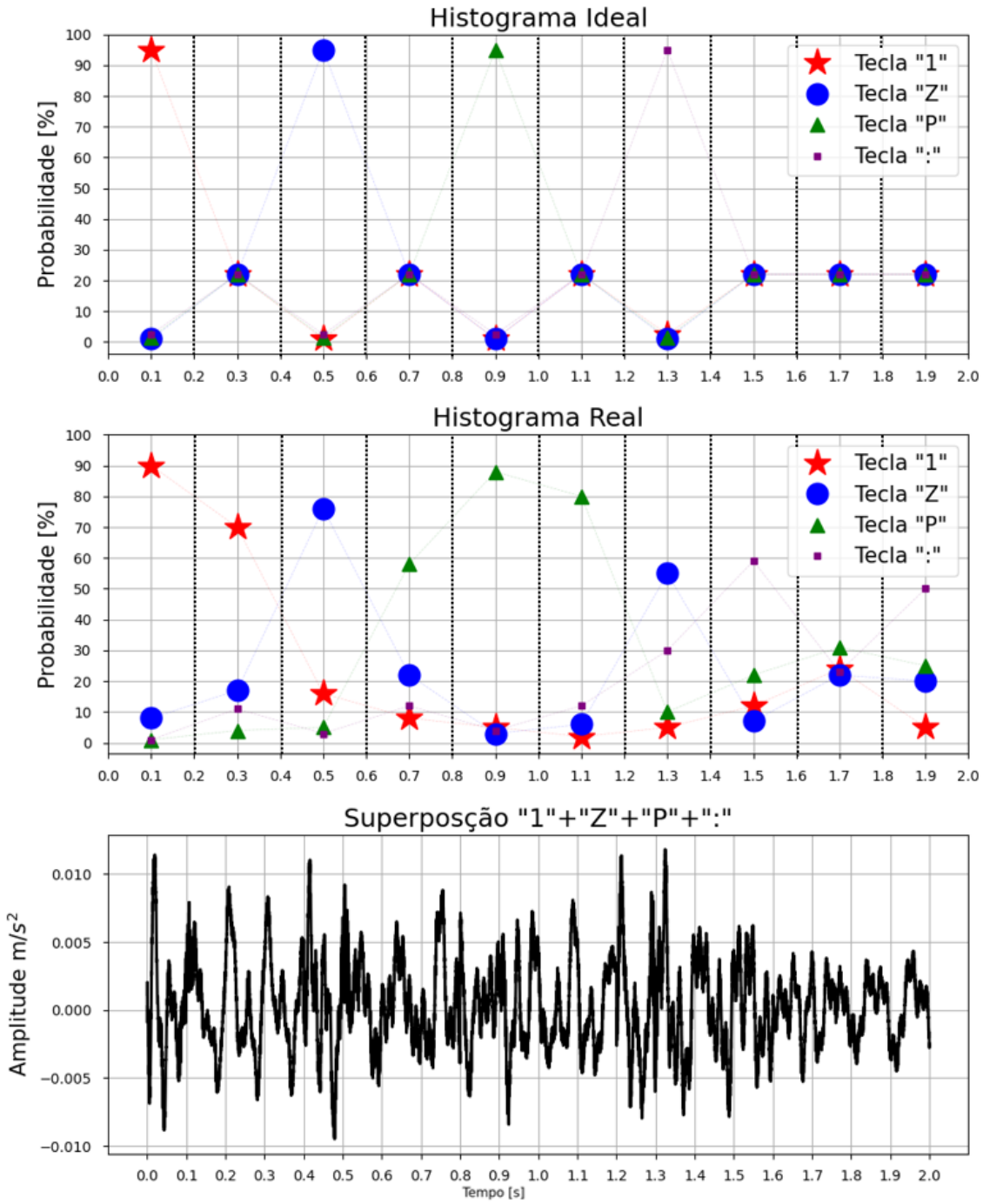

Figura 44 - Histograma de acurácia de saída da rede convolucional de 0.2s quando avaliado com dados superposto mostrados na Figura 43. As linhas tracejadas verticais demarcam os intervalos de tempo que foram alimentados a rede.

A Figura 44 mostra que o comportamento da rede convolucional, treinada com as imagens do escalograma, em momentos entre a digitação de teclas, não corresponde ao esperado. Mesmo sem a presença de um impacto de digitação, a rede tenta identificar os padrões aprendidos durante a etapa de treinamento e erroneamente, atribui probabilidades altas, no vetor de saída, a alguma tecla mesmo quando elas não foram digitadas.

Para excluir esses dados errôneos, fica clara a necessidade de um algoritmo para identificação do momento de digitação, para que a rede apenas avalie os sinais em momento em que há de fato uma digitação. Na Seção 3, esta identificação de digitações foi feita de forma manual, mas alguns algoritmos podem ser utilizados, como: transformada de Hilbert (ABEYSEKERA 2001), valor limite (ou seja, quando o sinal detectado é maior que um valor proposto anteriormente, é considerado um pico), algoritmo utilizando uma regra de dispersão e média (BRAKEL 2014) ou a utilização de outra rede neural como em MESSIAS 2023.

5.4 Resumo de Resultados

O diagrama mostrado na Figura 45 apresenta um resumo dos resultados obtidos nas seções 4 e 5, mostrando a evolução do problema e das acurácias para casos mais complexos e a diferença entre os tipos de redes utilizados. As abreviações são: “FC” para *fully connected* ou totalmente conectada, “ATP” para acurácia de teclas próximas, “ATD” para acurácia de teclas distantes, “PTP” para probabilidade de teclas próximas (quando redes já treinadas são avaliadas com o conjunto de dados superposto de teclas próximas), “PTD” para probabilidade de teclas distantes (quando redes já treinadas são avaliadas com o conjunto de dados superposto de teclas distantes), “Espec.” para espectrograma, “CWT” para a transformada continua wavelet, “EXD” para o método de extração de detalhes, “Op” para *Optuna*, “TL” para *transfer learning* e “GA” para algoritmo genético. A estrela azul demarca o caso com maior sucesso para identificação de teclas com sinais puros, utilizando a técnica de extração de detalhes, a estrela verde marca o melhor caso utilizando o algoritmo de *Transfer Learning* e a estrela amarela demarca a melhor rede para casos mais gerais, ou seja, com superposição.

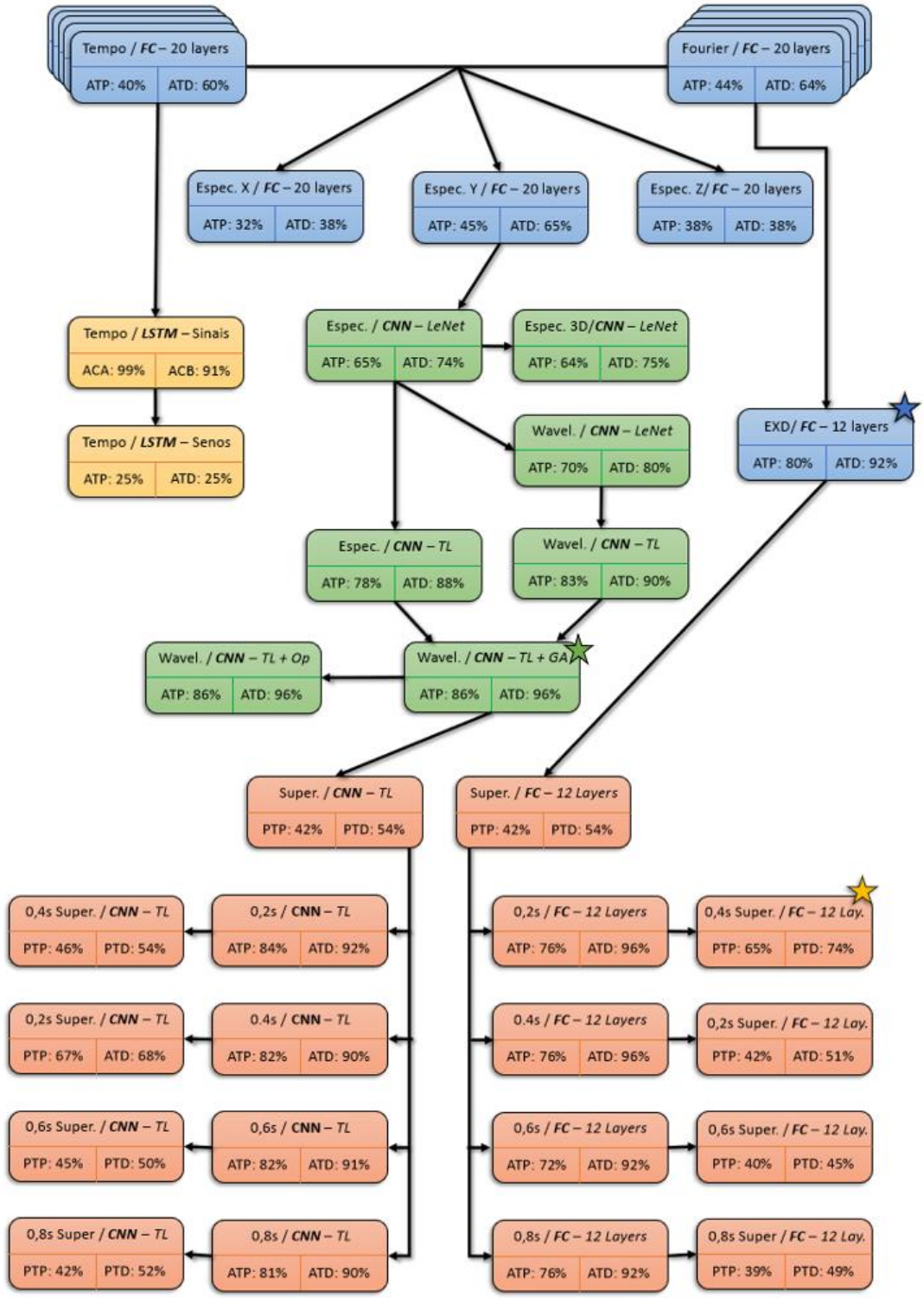

Figura 45 - Diagrama de resumo dos resultados encontrados até o presente momento.

6 CONCLUSÃO

Neste trabalho, exploramos diferentes modelos de redes neurais para identificação de sinais de aceleração resultante da digitação de caracteres em um teclado de computador. Buscou-se analisar quais os melhores tratamentos que esse sinal pode receber, bem como avaliar diferentes tipos de rede (convolucionais, LSTM, totalmente conectadas) e métodos de treinamento, a fim de se obter uma maior taxa de classificações corretas e a possibilidade de se decodificar textos digitados. Além disso, diferentes dados, superpostos e puros, foram analisados para se identificar os principais fatores que contribuem taxa de identificação.

Para o caso de dados sem superposição, a rede que obteve melhor resultado foi a rede *ResNet50*, com treinamento utilizando *transfer learning*, retreinada com as imagens da transformada wavelet, atingindo acurácia de 96% quando avaliada com sinais de teclas individuais de 1s. A rede totalmente conectada, avaliada com a extração de detalhes no sinal de 1s também teve desempenho.

Como mencionado, há fortes indícios de que as informações de maior interesse se encontram nas altas frequências dos sinais analisados. Essa informação pode ser corroborada com alguns resultados encontrados ao longo do trabalho, como, por exemplo, o fato de que a janela *Symlet*, que destaca melhor detalhes dos sinais de mais alta frequência, possui uma melhor aplicabilidade com relação a outra janela utilizada (*Daubecheis)*. Outro ponto que corrobora a afirmação acima, é o fato de que o conteúdo de baixa frequência (0 – 100Hz) é dominado pela resposta dinâmica da mesa, como mostrado na seção 3.2, o que pode ser um problema dada a possível variabilidade de resposta dela.

Mais um dado que vai ao encontro desse indício, é o fato de que a diminuição do sinal temporal de 1s para 0.2s não tem grande influência na acurácia da rede. Além de o dado de 0.2s ser concentrado no período de maior energia do sinal, as altas frequências se atenuam mais rápido com o passar do tempo, dado o amortecimento.

Já para o caso das redes para identificação das teclas superpostas, a melhor acurácia ficou em torno de 74%, quando a rede totalmente conectada foi treinada com a extração de detalhes em dados de 0.2s e avaliada com sinais superpostos de mesma duração. Essa rede também mostra uma vantagem (20%) nesse caso com relação à rede convolucional, mostrando a eficiência da extração de detalhes perante o reconhecimento do sinal tratado. Ao longo dos dados apresentados ficou claro o efeito negativo da superposição na identificação de sinais.

Mesmo com a acurácia menor do que os casos de sinais puros, os métodos mostrados na seção 5 apresentam melhora no resultado da rede quando utilizada para analisar sinais reais superpostos. Estes métodos também podem ser facilmente aplicados em estudos mostrados nas seções 1 e 2, que envolvem a classificação de sinais com baixa relação sinal-ruído.

6.1 Trabalhos futuros

Os resultados ainda podem ser melhorados com uso conjunto de outras técnicas já mostradas na Seção 2, como a latência entre digitação de teclas consecutivas, como no caso do *keyloggers* de Marquardt, a utilização de mais dados de outros sensores ou utilização de uma classificação hierárquica em que diferentes redes são treinadas para se especializar em diferentes regiões do teclado, e uma rede é treinada para pré-dividir o dado entre essas regiões do teclado, para decidir qual rede classificar o dado adquirido.

Além disso, como próximos passos deste estudo, recomenda-se a aquisição de um conjunto de dados maior, com mais classes (teclas) e mais dados por classe (impactos/digitações). Outro possível próximo passo é um estudo de possíveis características que possam ser extraídas da alta frequência do sinal, além das mencionadas nesse texto. Um conjunto de dados maior poderá ajudar a aumentar a robustez da rede treinada. A utilização de mais classes também é um passo necessário para garantir a acurácia da rede quando ele precisar identificar sinais entre as dezenas de teclas que compõem o teclado de computador. Outro ponto a ser considerado é a aquisição de sinais complementares, como som, ou outros sinais que possam ser adquiridos remotamente, sem serem invasivos ao sistema, que, em conjunto com os sinais de aceleração, podem contribuir para aumentar a acurácia de classificações das digitações.

Referências

ABEYSEKERA, S. S. **An efficient Hilbert transform interpolation algorithm for peak position estimation.** Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing, 2001.

ALLEMANG, R., **Vibrations: Experimental Modal Analysis**, ref: UC-SDRLCN-20-263-663/664, Structural Dynamics Research Laboratory, University of Cincinnati, 2001.

ALTES, R. **Detection, estimation, and classification with spectrograms**. The Journal of the Acoustical Society of America, edição 67, pagina 1230, junho de 1998. Disponível em: https://asa.scitation.org/doi/10.1121/1.384165

ASIF, M., *et al*, **Hybrid Deep Learning and Discrete Wavelet Transform-Based ECG Biometric Recognition for Arrhythmic Patients and Healthy Controls**. Sensors (Basel). 10 de maio de 2023

AZIZ, S., *et al*, **ECG-based machine-learning algorithms for heartbeat classification**. Sci Rep 11, 18738 (2021).

BACKES, M. *et al*. **Acoustic Side-Channel Attacks on Printers.** Saarland University, Computer Science Department, Saarbrucken, Germany, 2010.

BENGIO, Y. *et al.* **Learning long-term dependencies with gradient descent is difficult.** IEEE transactions on neural networks, 1994**.** 5(2), 157-166.

BERGER, Y. *et al.* **Dictionary Attacks Using Keyboard Acoustic Emanations**. Alexandria, Virginia, EUA, CCS'06, 2006.

BRAKEL, J.P.G. et al. **Robust peak detection algorithm using z-scores**. Stack Overflow. Disponível em: https://stackoverflow.com/questions/22583391/peak-signal-detection-in-realtime-timeseries-data/22640362#22640362 Acessado em: 20/01/2023.

BUENO, C. *et al.* **Vibration assessment at the carnauba beamline at the Sirius/LNLS.** Proceedings of the 11th International Conference on Mechanical Engineering Design of Synchrotron Radiation Equipment and Instrumentation, 2020.

CAI, L.; CHEN, H., **TouchLogger: Inferring Keystrokes on Touch Screen from Smartphone Motion**. 6th USENIX Workshop on Hot Topics in Security, 2011. Disponível em: https://www.usenix.org/legacy/events/hotsec11/tech/final_files/Cai.pdf

CHIN, C.H. *et al.* **Vibration fatigue analysis through frequency response function of variable amplitude loading**. *J Mech Sci Technol* **36**, 33–43 (2022).

COELHO, T. *et al*. **Automatic detection of diffraction-apex using fully convolutional networks**. Rio de Janeiro, RJ: 16th International Congress of the Brazilian Geophysical Society, 2019

CWT PYWAVELETS, **Continuous Wavelet Transform (CWT).** Disponível em: https://pywavelets.readthedocs.io/en/latest/ref/cwt.html?highlight=cwt. Acessado em 21/10/2022

DHAKAL, V. *et al*, **Observations on Typing from 136 Million Keystrokes.** In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI ’18).

DI NARDO, F. *et al*, **Machine Learning for Detection of Muscular Activity from Surface EMG Signals.** Sensors (Basel) Journal, 2022. doi: 10.3390/s22093393

DJAIDIR, B. **Faults detection in gas turbine rotor using vibration analysis under varying conditions**. *Journal of theoretical and applied mechanics, 2017, 55*(2), pp.393-406.

FAVERO, A. *et al*, **Reconhecimento de Caracteres Matemáticos utilizando Deep Learning.** Junho de 2020. Disponível em: http://cidamo.com.br/2020-06-30-reconhecimento-de-caracteres/ . Acessado em: 18/04/2023.

FFT SCIPY, **Scipy Documentation, fft function.** Disponível em: https://docs.scipy.org/doc/scipy/reference/generated/scipy.fft.fft.html. Acessado em: 20/10/2022.

GAIOTTO, T. G., LABAKI, J. **Análise estatística da latência de digitação para a identificação de senhas**. Trabalho de Conclusão de Curso. Faculdade de Engenharia Mecânica, Unicamp, 2022.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning.** *Addison-Wesley, 1989.*

HARRISON, J. *et al.* **A Practical Deep Learning-Based Acoustic Side Channel Attack on Keyboards.** IEEE European Symposium on Security and Privacy Workshop, SiLM'23 (EuroS&PW), 2023.

HOCHREITER, S., SCHMIDHUBER, J., **Long Short-Term Memory.** Monique, Alemanha, 1997.

ILSVRC, **Imagenet - ImageNet Large Scale Visual Recognition Challenge.** Disponível em: https://www.image-net.org/index.php . Acessado em 20/12/2022.

KELLY, A. **Cracking Passwords using Keyboard Acoustics and Language Modeling.** Master of Science, Artificial Intelligence, School of Informatics, University of Edinburgh, 2010

KERAS, **Keras: The Python deep learning API**, 2022. Disponível em: https://keras.io/ . Acessado em: 20/12/2022.

KOHAVI, R. **A study of cross-validation and bootstrap for accuracy estimation and model selection.** *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (IJCAI), 1995.

KRIZHEVSKY, A. *et al.* **ImageNet Classification with Deep Convolutional Neural Networks.** Advances in Neural Information Processing Systems 25 (NIPS 2012).

LAARADJ, S. H. *et al.* **Vibration-based fault diagnosis of dynamic rotating systems for real-time maintenance monitoring**. *The International Journal of Advanced Manufacturing Technology, 2023.*

LECUN, Y. *et al*. **Gradient-based learning applied to document recognition**. in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

LEE, H. *et al*. **Power Signal Classification with Combinational Spectrogram-based CNN for Embedded System Health Management**. 18th International Conference on Control, Automation and Systems (ICCAS), 2018. Disponível em: https://ieeexplore.ieee.org/document/8571740

LEVY, J. *et al*. **Classification of audio signals using spectrogram surfaces and extrinsic distortion measures**. EURASIP Journal on Advances in Signal Processing, 2022. Disponivel em: https://asp-eurasipjournals.springeropen.com/articles/10.1186/s13634-022-00933-9

LOY, C. C. *et al*. **Keystroke Patterns Classification Using the ARTMAP-FD Neural Network**. Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007), 2007. Disponível em: https://doi.org/10.1109/IIH-MSP.2007.218

MAITI, A. *et al*. **Smartwatch-Based Keystroke Inference Attacks and Context-Aware Protection Mechanisms**. The 11th ACM, 2016. Disponível em: http://dx.doi.org/10.1145/2897845.2897905

MAJUMDER S. & DEEN M. J. **Smartphone Sensors for Health Monitoring and Diagnosis**. *Sensors*. 2019; 19(9):2164.

MALADKAR, K. **Overview of Convolutional Neural Network in image classification** Janeiro de 2018. Disponível em: https://analyticsindiamag.com/convolutional-neural-network-image-classification-overview/ . Acessado em: 18/04/2023

MALLAT, S. **A Wavelet Tour of Signal Processing: The Sparse Way** terceira edição, editora academic press Elsevier, 2009.

MARQUARDT, P. *et al*. (**sp)iPhone: Decoding Vibrations From Nearby Keyboards Using Mobile Phone Accelerometers**. Chicago, Illinois, EUA, CCS'11, outubro de 2011. Disponível em: https://www.cise.ufl.edu/~traynor/papers/marq-ccs11.pdf

MATLAB CNN, **Visualize Features of a Convolutional Neural Network.** Disponível em: https://www.mathworks.com/help/deeplearning/ug/visualize-features-of-a-convolutional-neural-network.html . Acessado em: 20/12/2022

MESSIAS, M. E. X.; LABAKI, J. **An acoustic keylogger for remote attack using deep learning**. XXXI Congresso de Iniciação Científica da Unicamp, PIBIC 2023.

MLM, **LSTMs for Human Activity Recognition Time Series Classification**, Machine Learning Mastery, 2018. Disponível em: https://machinelearningmastery.com/how-to-develop-rnn-models-for-human-activity-recognition-time-series-classification/. Acessado em: 20/12/2022.

MOHAMED, A. *et al.* **Classification of Radar and Communications Signals using Wideband Autonomous Cognitive Radios**. Albuquerque, NM, EUA, Communications and Information Sciences Laboratory (CISL), Department of Electrical and Computer Engineering, University of New Mexico, 2018. Disponível em: http://ece-research.unm.edu/cisl/publication/WACR/conf_2018_12_01.pdf

MONACO, J. V. **SoK: Keylogging Side Channels.** Aberdeen Proving Ground, MD, EUA, U.S. Army Research Laboratory, 2018. Disponível em: https://vmonaco.com/papers/SoK-%20Keylogging%20Side%20Channels.pdf

NIELSEN, M., **Neural Networks and Deep Learning**. 2019 Livro online, disponível em: http://neuralnetworksanddeeplearning.com/ . Acessado em: 20/12/2022.

NARAIN, S. *et al.* **Single-stroke Language-Agnostic Keylogging using Stereo-Microphones and Domain Specific Machine Learning**. Oxford, Reino Unido, WiSec'14, julho 2014. Disponível em: https://dl.acm.org/doi/10.1145/2627393.2627417

OLOFSSON, J. **Accelerometer Mounting.** Swedish National Testing and Research Institute, Physics and Electrotechnics, 1995.

OPPENHEIM, A. V. et al **Digital Signal Processing**, 1st Edition, Pearson, 1975.

OPTUNA, **Optuna Python Library**. Disponível em: https://optuna.org/ . Acessado em: 19/11/2022

O'SHEA, K. *et al* **An Introduction to Convolutional Neural Networks.** Computer Vision and Pattern Recognition, 2015.

OZTURK, E. *et al*. **RF-Based Low-SNR Classification of UAVs Using Convolutional Neural Networks**. Electrical Engineering and Systems Science - Signal Processing, 2020. Disponível em: https://doi.org/10.48550/arXiv.2009.05519

PAOLANTI, M., *et al.*, **Machine Learning approach for Predictive Maintenance in Industry 4.0.** 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), 2018.

PATTNAIK, S. et al. **DWT-based feature extraction and classification for motor imaginary EEG signals.** International Conference on Systems in Medicine and Biology (ICSMB), 2016. doi: 10.1109/ICSMB.2016.7915118

PYGAD, **PyGAD - Python Genetic Algorithm.** Disponível em: https://pygad.readthedocs.io/en/latest/ . Acessado em: 20/12/2022.

PYWAVELETS, **PyWavelets - Wavelet Transforms in Python.** Disponível em: https://pywavelets.readthedocs.io/en/latest/. Acessado em 21/10/2022.

RIBEIRO, M. S.; LABAKI, J. **A platform for mobile sensor signal acquisition and cloud neural network feeding**. XXX Congresso de Iniciação Científica da Unicamp, PIBIC 2022.

RILEY, R. *et al*. **Classification of low-SNR side channels**. Orlando, Florida, EUA, SPIE Defense + Security, 2018. Disponível em: https://doi.org/10.1117/12.2304461

ROSENBLATT, F., **Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms**. Spartan Books, 1962, EUA. Disponível em: https://babel.hathitrust.org/cgi/pt?id=mdp.39015039846566&view=1up&seq=9

SARVAYA, J. & PRATNAIK, S., **Automatic Image Registration Using Mexican Hat Wavelet, Invariant Moment, and Radon Transform.** (IJACSA) International Journal of Advanced Computer Science and Applications, Special Issue on Image Processing and Analysis, 2010.

SASTRY, K. **Search Methodologies**. Springer, Boston, MA. Disponível em: https://doi.org/10.1007/0-387-28356-0_4

SCHUCHMANN, S. **Analyzing the Prospect of an Approaching AI Winter**. Darmstadt University of Applied Sciences, 2019. Disponível em: https://www.researchgate.net/publication/333039347_Analyzing_the_Prospect_of_an_Approaching_AI_Winter

SCIPY, **SciPy Documentation**. Disponível em: https://docs.scipy.org/doc/scipy/index.html. Acessado em 20/10/2022.

SIRIUS, **Sirius, acelerando o futuro da ciência brasileira**. Disponível em https://lnls.cnpem.br/sirius/ . Acessado em: 20/12/2022.

SPECTROGRAM SCIPY, **Scipy Documentation, spectrogram function** Disponível em: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.spectrogram.html#scipy.signal.spectrogram . Acessado em 20/10/2022.

TENSORFLOW, **Tensorflow, machine learning library**. Disponível em: https://www.tensorflow.org/ . Acessado em: 20/12/2022.

UPRETEE, P. & YUKSEL, M. **Accurate classification of heart sounds for disease diagnosis by using spectral analysis and deep learning methods.** Livro: Data Analytics in Biomedical Engineering and Healthcare, 2021 Páginas 215 – 232.

VINICIUS, M. **Redes Neurais Artificiais**. Junho de 2017. Disponível em: https://www.monolitonimbus.com.br/redes-neurais-artificiais/ . Acessado em: 18/04/2023.

VUAGNOUX, M.; PASINI, S., **Compromising electromagnetic emanations of wired and wireless keyboards**. Montreal, Canada, 18th USENIX Security Symposium (Usenix Security '09), 2009. Disponível em: https://infoscience.epfl.ch/record/140523?ln=en

WIDODO, *et al*. **Detecting Buried Human Bodies Using Ground-Penetrating Radar**. Canadian Center of Science and Education, janeiro de 2016. Disponível em: http://dx.doi.org/10.5539/esr.v5n2p59

XIMING, L., **When Keystroke Meets Password: Attacks and Defenses**. Tese de doutorado em Singapore management University, Information Systems Department, 2019. Disponível em: https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=1248&context=etd_coll

ZANETTI, S. *et al*. **Reference evapotranspiration estimate in Rio de Janeiro State using artificial neural networks**. Revista Brasileira de Engenharia Agrícola e Ambiental, abril de 2008.

ZOU, B. *et al*. **Research on Signal Modulation Classification under Low SNR Based on ResNext Network**. Electronics 2022, 11, 2662. Disponível em: https://www.mdpi.com/2079-9292/11/17/2662.

ZHOU, V. **CNNs, Part 1: An Introduction to Convolutional Neural Networks**. Novembro de 2020. Disponível em: https://victorzhou.com/blog/intro-to-cnns-part-1/ . Acessado em: 18/04/2023.

ZHOU, Z. *et al*. **The Next Failure Time Prediction of Escalators via Deep Neural Network with Dynamic Time Warping Preprocessing**. Applied Sciencies Journal, agosto de 2020. Disponível em: https://www.mdpi.com/2076-3417/10/16/5622

ZHU, K., & HUANG, W. **Fault diagnosis of rolling element bearings using an improved local mean decomposition method**. 2019, *Journal of Sound and Vibration, 445*, 29-47.