

**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

João Paulo Brognoni Casati

**Análise e Simulação de Sinais de Acelerômetro Gerados
por Vibração de Impacto de Veículos em Lombada
Utilizando Inteligência Artificial**

São Carlos

2021

João Paulo Brognoni Casati

**Análise e Simulação de Sinais de Acelerômetro Gerados
por Vibração de Impacto de Veículos em Lombada
Utilizando Inteligência Artificial**

Tese apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, para obtenção do título de Doutor em Ciências - Programa de Pós-Graduação em Engenharia Elétrica.

Área de concentração: Sistemas Elétricos de Potência

Orientador: Prof. Dr. Ruy Alberto Corrêa Altafim

Trata-se da versão corrigida da tese. A versão original se encontra disponível na EESC/USP que aloja o Programa de Pós-Graduação de Engenharia Elétrica.

São Carlos

2021

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

B336a Brognoni Casati, João Paulo
Análise e Simulação de Sinais de Acelerômetro Gerados por Vibração de Impacto de Veículos em Lombada Utilizando Inteligência Artificial / João Paulo Brognoni Casati; orientador Ruy Alberto Corrêa Altafim. São Carlos, 2021.

Tese (Doutorado) - Programa de Pós-Graduação em Engenharia Elétrica e Área de Concentração em Sistemas Elétricos de Potência -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2021.

1. inteligência artificial. 2. processamento de sinais. 3. acelerômetro. 4. aprendizagem de máquina. 5. cidades inteligentes. 6. deep learning. 7. geração de dados. 8. redes neurais artificiais. I. Título.

FOLHA DE JULGAMENTO

Candidato: Bacharel **JOÃO PAULO BROGNONI CASATI**.

Título da tese: “Análise e simulação de sinais de acelerômetro gerados por vibração de impacto de veículos em lombada utilizando inteligência artificial”.

Data da defesa: 14/06/2021.

Comissão Julgadora

Resultado

Prof. Titular **Ruy Alberto Corrêa Altafim**

Aprovado

(Orientador)

(Escola de Engenharia de São Carlos – EESC/USP)

Prof. Associado **Rogério Andrade Flauzino**

Aprovado

(Escola de Engenharia de São Carlos – EESC/USP)

Prof. Dr. **Ricardo Augusto Souza Fernandes**

Aprovado

(Universidade Federal de São Carlos/UFSCar)

Prof. Dr. **Mardson Freitas de Amorim**

Aprovado

(Universidade Federal da Paraíba/UFPB)

Prof. Dr. **Danilo Hernane Spatti**

Aprovado

(Instituto de Ciências Matemáticas e de Computação/ICMC-USP)

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica:
Prof. Dr. **João Bosco Augusto London Junior**

Presidente da Comissão de Pós-Graduação:
Prof. Titular **Murilo Araujo Romero**

*Este trabalho é dedicado à
minha filha, Beatriz.*

AGRADECIMENTOS

Ao Prof. Dr. Ruy Alberto Corrêa Altafim, meu orientador, por toda atenção e dedicação neste projeto de pesquisa científica.

Aos membros da banca de qualificação, Profs. Drs.: Frank José Affonso e Celso Aparecido de França; pela participação e dedicação nas sugestões de correções deste trabalho.

Ao corpo docente que ministrou as minhas disciplinas, Profs. Drs.: Adilson Gonzaga, Ivan Nunes da Silva, Marcelo Andrade da Costa Vieira e Maria Stela Veludo de Paiva; pelos conhecimentos e conselhos acadêmicos, que somados resultaram no profissional que sou.

Aos funcionários da USP de São Carlos, sempre atenciosos e dispostos a ajudar em qualquer tipo de problema, em especial à Marisa Helena R. Vicente Fortulan, Leonardo Erivelto Soares de Oliveira, Roseli Ribeiro e Rosane A. Aranda.

Aos Profs. Drs.: Adilson Gonzaga e Osvaldo Severino Júnior, pela disponibilização das imagens de ground truths utilizadas neste trabalho.

À minha filha Beatriz Nunes Casati e à minha esposa Siméia Rafaela Nunes Casati por todo apoio, pelo incentivo e por serem as razões da minha dedicação.

Aos meus pais, Paulo Carvalho Casati e Cassandra Marta Alves Brognoni Casati pelo amor, carinho e instrução que foram fundamentais na formação do meu caráter.

Por último, mas não menos importantes, a todos os meus amigos e colegas pessoais, profissionais e da pós-graduação, que sempre estiveram disponíveis a me ajudar em qualquer tipo de problema, os quais sabem que não é necessário listar os seus nomes aqui, porque verdadeiramente conhecem a sua importância para mim.

*“O maior bem do homem
é uma mente inquieta.”*

Isaac Asimov

RESUMO

CASATI, J. P. B. **Análise e Simulação de Sinais de Acelerômetro Gerados por Vibração de Impacto de Veículos em Lombada Utilizando Inteligência Artificial**. 2021. 108p. Tese (Doutorado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2021.

Analisar veículos e seu comportamento pode ser útil para o monitoramento de diversos problemas de tráfego, como danos à pavimentação e engarrafamentos, assim como a classificação de tipo de veículo. O objetivo deste trabalho é realizar uma análise aprofundada do comportamento da vibração causada pelo impacto de veículos em um obstáculo utilizando inteligência artificial. Para a análise destes dados são utilizadas técnicas de *deep learning* que possibilitam a utilização dos 3 eixos de dados do acelerômetro de forma crua, além da simulação de amostras artificiais que permitem melhorar os resultados e resolver problemas de *oversampling*. Conclui-se com este trabalho que os resultados alcançados são promissores e permitem avanços em aplicações que possam considerar tráfego de veículos, monitoramento e soluções em cidades inteligentes.

Palavras-chave: Inteligência artificial. Processamento de sinais. Acelerômetro. Aprendizagem de máquina. Cidades inteligentes. 6. Deep learning. Geração de dados. Redes neurais artificiais.

ABSTRACT

CASATI, J. P. B. **Analysis and Simulation of Accelerometer Signals Generated by Vehicle Impact Vibration in Speed Bump Using Artificial Intelligence.** 2021. 108p. Tese (Doutorado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2021.

Analyzing vehicles and their behavior can be useful for monitoring various traffic problems, such as damage to pavement and traffic jams, as well as vehicle type classification. The objective of this work is to carry out an in-depth analysis of the vibration behavior caused by the impact of vehicles on an obstacle using artificial intelligence. For the analysis of this data, deep learning techniques are used, which allow the raw use of the 3 axes of the accelerometer data, in addition to the simulation of synthetic samples that improve the results and solve oversampling problems. This work concludes that the results achieved are promising and allow advances in applications considering vehicle traffic, monitoring and solutions in smart cities.

Keywords: Artificial intelligence. Signal processing. Accelerometer. Machine learning. Smart cities. Deep learning. Data augmentation. Artificial neural networks.

LISTA DE FIGURAS

Figura 1 – Diagrama de execução da experimentação inicial.	29
Figura 2 – Diagrama de execução da experimentação principal.	30
Figura 3 – Exemplo de neurônio artificial (<i>perceptron</i> simples).	37
Figura 4 – Arquitetura de rede neural PMC (<i>Perceptron</i> Multicamadas), em inglês MLP (<i>Multi-Layer Perceptron</i>).	40
Figura 5 – Comportamento das camadas em relação ao espaço amostral para problemas de classificação de padrões. Em (a) é exibida a topologia da rede neural e (b) exibe o espaço amostral e a capacidade de separação fornecida pela topologia.	41
Figura 6 – Arquitetura simplificada de uma célula da rede neural LSTM.	43
Figura 7 – Utilização de LSTM em camadas.	44
Figura 8 – Exemplo de arquitetura padrão para <i>autoencoders</i>	45
Figura 9 – Exemplo de operação de convolução em uma matriz 2D, com máscara de 2x2.	47
Figura 10 – Camada convolucional em uma rede neural artificial.	48
Figura 11 – Processo de execução de uma rede geradora (GAN).	49
Figura 12 – Processo de execução de uma rede discriminadora (GAN).	50
Figura 13 – Foto real do Acelerômetro de Piezoeletreto.	52
Figura 14 – Foto real da lombada que foi desenvolvida.	53
Figura 15 – Foto aproximada do posicionamento dos tacos na parte inferior da lombada.	53
Figura 16 – A lombada confeccionada pintada e posicionada para fazer a aquisição dos dados.	54
Figura 17 – Tela principal do aplicativo <i>KaptAccel</i>	56
Figura 18 – QR Code utilizado para acesso ao aplicativo <i>KaptAccel</i>	57
Figura 19 – Gráfico que representa o comportamento de 500 amostras no tempo.	58
Figura 20 – Atribuição de classe para saída desejada em amostras com mais de 1 sinal. Exemplo utilizando 5 sinais por amostra.	64
Figura 21 – Exemplo de amostra real de tamanho 10, onde as colunas representam os eixos do acelerômetro e as linhas as capturas ao longo do tempo.	71
Figura 22 – Gráfico que representa a variação dos valores de EQM para os treina- mentos dos diferentes tamanhos de entrada.	76
Figura 23 – Gráfico que apresenta um comparativo do percentual de acurácia dos testes para os diferentes tamanhos de entrada.	79

Figura 24 – Gráfico que apresenta a comparação entre duas amostras de tamanho 20, para a classe 1. No segundo quarto do Eixo T uma amostra real e no terceiro quarto uma amostra sintética.	81
Figura 25 – Gráfico que apresenta uma segunda comparação entre duas amostras de tamanho 20, para a classe 1. No segundo quarto do Eixo T uma amostra real e no terceiro quarto uma amostra sintética.	82
Figura 26 – Gráfico que apresenta uma comparação entre duas amostras de tamanho 20, para a classe 2. No segundo quarto do Eixo T uma amostra real e no terceiro quarto uma amostra sintética.	82
Figura 27 – Gráfico que apresenta uma comparação entre duas amostras de tamanho 20, para a classe 3. No segundo quarto do Eixo T uma amostra real e no terceiro quarto uma amostra sintética.	83
Figura 28 – Gráfico que apresenta uma comparação entre duas amostras de tamanho 15, para a classe 1. No segundo quarto do Eixo T uma amostra real e no terceiro quarto uma amostra sintética.	83
Figura 29 – Gráfico que apresenta uma segunda comparação entre amostras de tamanho 1, para a classe 2. O segundo quarto do Eixo T possui 20 amostras reais o terceiro quarto 20 amostras sintéticas.	84
Figura 30 – Gráfico que apresenta um comparativo entre os valores de erro alcançados pela rede MLP utilizando as amostras reais.	87
Figura 31 – Gráfico que apresenta um comparativo entre os valores de acurácia alcançados pela rede MLP utilizando as amostras reais.	88
Figura 32 – Matriz de confusão do resultado do teste de classificação com amostras reais utilizando MLP, amostras de tamanho 15 e 4 classes (valores em %). Legenda: quantidade de amostras.	89
Figura 33 – Gráfico que apresenta um comparativo entre os valores de erro alcançados pela rede MLP utilizando amostras sintéticas.	90
Figura 34 – Gráfico que apresenta um comparativo entre os valores de acurácia alcançados pela rede MLP utilizando as amostras sintéticas.	91
Figura 35 – Matriz de confusão do resultado do teste de classificação com amostras sintéticas utilizando MLP, amostras de tamanho 15 e 4 classes (valores em %). Legenda: quantidade de amostras.	92
Figura 36 – Matriz de confusão do resultado do teste de classificação com amostras sintéticas utilizando MLP, amostras de tamanho 1 e 2 classes (valores em %). Legenda: quantidade de amostras.	93
Figura 37 – Gráfico que apresenta um comparativo entre os valores de erro alcançados pela rede CNN utilizando as amostras reais.	95
Figura 38 – Gráfico que apresenta um comparativo entre os valores de acurácia alcançados pela rede CNN utilizando as amostras reais.	96

Figura 39 – Matriz de confusão do resultado do teste de classificação com amostras reais utilizando CNN, amostras de tamanho 20 e 4 classes (valores em %). Legenda: quantidade de amostras.	97
Figura 40 – Gráfico que apresenta um comparativo entre os valores de erro alcançados pela rede CNN utilizando amostras sintéticas.	98
Figura 41 – Gráfico que apresenta um comparativo entre os valores de acurácia alcançados pela rede CNN utilizando as amostras sintéticas.	99
Figura 42 – Matriz de confusão do resultado do teste de classificação com amostras sintéticas utilizando CNN, amostras de tamanho 20 e 2 classes (valores em %). Legenda: quantidade de amostras.	99
Figura 43 – Matriz de confusão do resultado do teste de classificação com amostras sintéticas utilizando CNN, amostras de tamanho 15 e 4 classes (valores em %). Legenda: quantidade de amostras.	100

LISTA DE TABELAS

Tabela 1 – Resultados do Erro Quadrático Médio no treinamento	75
Tabela 2 – Resultados com entrada de 10 amostras	76
Tabela 3 – Resultados com entrada de 20 amostras	77
Tabela 4 – Resultados com entrada de 30 amostras	77
Tabela 5 – Resultados com entrada de 40 amostras	78
Tabela 6 – Resultados com entrada de 50 amostras	78
Tabela 7 – Resultados de treinamento da MLP utilizando apenas amostras reais. .	86
Tabela 8 – Resultados da acurácia da classificação do teste da MLP utilizando apenas amostras reais.	87
Tabela 9 – Resultados de treinamento da MLP e adição de amostras sintéticas. . .	89
Tabela 10 – Resultados de acurácia da classificação do teste da MLP utilizando amostras sintéticas adicionadas às reais.	91
Tabela 11 – Resultados de treinamento da CNN utilizando apenas amostras reais. .	94
Tabela 12 – Resultados da acurácia da classificação do teste da CNN utilizando apenas amostras reais.	95
Tabela 13 – Resultados de treinamento da CNN e adição de amostras sintéticas. . .	97
Tabela 14 – Resultados de acurácia da classificação do teste da CNN utilizando amostras sintéticas adicionadas às reais.	98

LISTA DE QUADROS

Quadro 1 – Medidas da lombada (em centímetros).	54
Quadro 2 – Explicação detalhada das diferentes categorias de veículos analisadas na experimentação principal.	59
Quadro 3 – Números importantes sobre a etapa de aquisição de dados para a experimentação principal.	60
Quadro 4 – Divisão das amostras para o experimento inicial de validação.	62
Quadro 5 – Divisão das amostras para o experimento inicial de validação.	63
Quadro 6 – Topologia da rede neural geradora de simulações.	66
Quadro 7 – Topologia da rede neural discriminadora de simulações.	67
Quadro 8 – Parâmetros de treinamento da rede neural GAN.	67
Quadro 9 – Topologia da rede neural MLP utilizada para classificação dos dados reais.	69
Quadro 10 – Parâmetros utilizados para treinamento das redes neurais de classificação das amostras utilizando a arquitetura MLP.	70
Quadro 11 – Topologia da rede neural CNN utilizada para classificação.	72

LISTA DE ABREVIATURAS E SIGLAS

CCE	<i>Categorical Cross-Entropy</i>
CNN	<i>Convolutional Neural Networks</i>
CSV	<i>Comma-Separated Values</i>
EQM	Erro Quadrático Médio
GAN	<i>Generative Adversarial Networks</i>
IA	Inteligência Artificial
LSTM	<i>Long-Short Term Memory</i>
ML	<i>Machine Learning</i>
MLP	<i>Multi-Layer Perceptron</i>
RNN	<i>Recurrent Neural Networks</i>
VNS	<i>Variable Noise Superposition</i>

SUMÁRIO

1	INTRODUÇÃO	27
1.1	Trabalhos Relacionados	30
1.2	Objetivos	32
2	APRENDIZAGEM DE MÁQUINA	35
2.1	Introdução	35
2.2	Redes Neurais Artificiais	37
2.2.1	Redes Perceptron Multicamadas	39
2.3	<i>Deep Learning</i>	40
2.3.1	<i>Long-Short Term Memory</i>	42
2.3.2	<i>Autoencoders</i>	44
2.3.3	<i>Convolutional Neural Networks</i>	46
2.3.4	<i>Generative Adversarial Networks (GAN)</i>	48
3	AQUISIÇÃO DE DADOS	51
3.1	Lombada	52
3.2	Aplicativo Móvel	54
3.3	Experimentos de Aquisição	56
3.3.1	Experimentação Inicial	57
3.3.2	Experimentação Principal	58
4	PROCESSAMENTO DOS DADOS	61
4.1	Experimentação Inicial	61
4.2	Experimentação Principal	62
4.2.1	Geração de Simulações	65
4.2.2	Classificação MLP	68
4.2.2.1	Dados Reais	68
4.2.2.2	Dados Simulados	70
4.2.3	Classificação Convolutacional	71
5	RESULTADOS E DISCUSSÃO	75
5.1	Experimentação Inicial	75
5.2	Experimentação Principal	80
5.2.1	Geração de Simulações	80
5.2.2	Classificação MLP	85
5.2.2.1	Dados Reais	85
5.2.2.2	Dados Simulados	88

5.2.3	Classificação Convolucional	94
5.2.3.1	Dados Reais	94
5.2.3.2	Dados Simulados	96
6	CONCLUSÃO	103
	REFERÊNCIAS	105

1 INTRODUÇÃO

Aplicações utilizando inteligência artificial (IA), mais precisamente redes neurais artificiais, são estudadas desde a década de 1940. Entretanto o custo computacional, a escassez de dados e a falta de arquiteturas que pudessem resolver estes problemas, sejam elas software, hardware ou algoritmos, limitavam a utilização da IA em soluções de engenharia mais complexas (SILVA; SPATTI; FLAUZINO, 2010).

Ainda de acordo com Silva, Spatti e Flauzino (2010), a partir dos anos 1980 os pesquisadores voltaram a ter interesse de pesquisa nesta área. Porém, os estudos se intensificaram nos anos 1990.

Como apresentado no artigo de Goodfellow, Bengio e Courville (2016), o desenvolvimento de novas soluções e tecnologias possibilita que, hoje em dia, seja possível criar aplicações mais complexas e que consideram um universo maior de conhecimento. Algumas destas soluções são:

- Hardware específico;
- Bibliotecas de software;
- Algoritmos otimizados;
- Abundância de dados;

De acordo com Teffé, Branco e Vicente (2019), juntamente às novas tecnologias de IA que foram sendo desenvolvidas, surgiram novas demandas para aplicações. As cidades, seu planejamento e necessidades configuram um vasto campo para este tipo aplicação.

De acordo com diferentes pesquisas realizadas por órgãos governamentais e não governamentais ao redor do mundo, a população que reside em áreas urbanas está em constante crescimento. Estima-se que no ano de 2050, 68% da população será urbana (SÁNCHEZ-CORCUERA *et al.*, 2019).

Além da migração de pessoas para áreas urbanas, tem-se também o crescimento populacional mundial como fator que contribui para o aumento do número de pessoas que moram nas cidades (TEFFÉ; BRANCO; VICENTE, 2019).

No contexto brasileiro, as cidades acompanham este cenário, tornando os centros urbanos do país cada vez mais populosos, o que demanda soluções para que a acomodação, transporte e qualidade de vida das pessoas sejam menos afetada com a superpopulação em algumas áreas (TEFFÉ; BRANCO; VICENTE, 2019).

Seja para cidades que foram planejadas para absorver sistemas inteligentes ou aquelas que se adaptam para que haja alguma melhoria advinda da tecnologia, Sánchez-Corcuera *et al.* (2019) pontuam que as aplicações são diversas e possuem uma gama diversificada de objetivos.

Por estes motivos, diversas *startups* vêm sendo criadas no país, buscando soluções que possam se adaptar à realidade brasileira (DISTRITO, 2020).

Além do potencial deste trabalho ser aplicado à diversas soluções em cidades inteligentes, principalmente referentes à tráfego, esta tese traz uma análise aprofundada de diversos aspectos da classificação de sinais, que também permitem aplicações para outras finalidades.

Os sinais aqui utilizados advêm de acelerômetros, que são dispositivos capazes de detectar vibrações e movimentações, possibilitando seu uso para diversas finalidades.

Sua capacidade é de medir a aceleração de um objeto em uma dada direção. Isto é possível pois o deslocamento do dispositivo pode ser medido em três eixos, gerando assim um sinal de três dimensões (x, y, z) (POLIZZI; FAIN; MASPERO, 2020).

Ao analisar o sinal obtido por meio do acelerômetro, pode-se estimar o deslocamento do dispositivo (ou de um objeto no qual ele está acoplado) em qualquer direção, além do valor do sinal oferecido ser proporcional à real aceleração (THOMAZINI; ALBUQUERQUE, 2020).

Atualmente, os acelerômetros são comumente desenvolvidos utilizando a tecnologia MEMS (*Micro Electrical-Mechanical Systems*), terminologia que define dispositivos desenvolvidos utilizando micro-usinagem e são eletro-mecânicos (POLIZZI; FAIN; MASPERO, 2020).

Hoje em dia, a utilização do acelerômetro evoluiu e se difundiu comercialmente, possibilitando o acesso à dispositivos de alta precisão acoplados em *smartphones*, o que possibilita seu uso em aplicações corriqueiras e com baixo custo. [REF]

A partir do cenário apresentado, o trabalho aqui proposto utiliza estes sensores com o objetivo de analisar dados gerados por acelerômetros a partir do impacto de veículos em um obstáculo.

Dentre as tecnologias já apresentadas, tem-se também o uso de técnicas de inteligência artificial que permitem gerar novas simulações de impactos artificiais e classificar os sinais, extraíndo assim conhecimento acerca dos dados a serem analisados.

Em um primeiro momento, este trabalho apresenta uma introdução a respeito de temas pertinentes e cita alguns trabalhos relacionados, de acordo com as tecnologias empregadas e as semelhanças quanto aos objetivos gerais.

Logo após, uma fundamentação teórica sobre inteligência artificial é apresentada,

com foco em técnicas e tecnologias que são diretamente relacionadas aos processos executados neste trabalho.

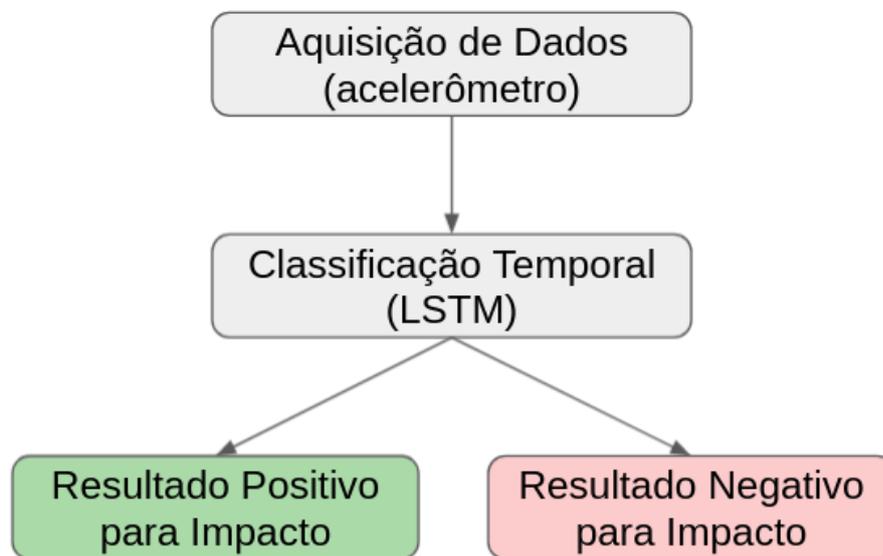
As próximas etapas a serem apresentadas fazem parte da execução do trabalho em si e são divididas em dois experimentos:

- Experimento inicial;
- Experimentação principal.

Para as etapas de aquisição de dados, as Seções 3.1 e 3.2 apresentam o desenvolvimento de ferramentas que permitem leitura e armazenamento dos sinais, as quais são utilizadas nos dois experimentos.

A experimentação inicial tem seus processos apresentados na Figura 1, onde pode-se observar que a partir da aquisição de dados, tem-se uma etapa de classificação temporal utilizando uma rede *Long-Short Term Memory* (LSTM), que é alimentada pelos sinais de forma crua.

Figura 1 – Diagrama de execução da experimentação inicial.



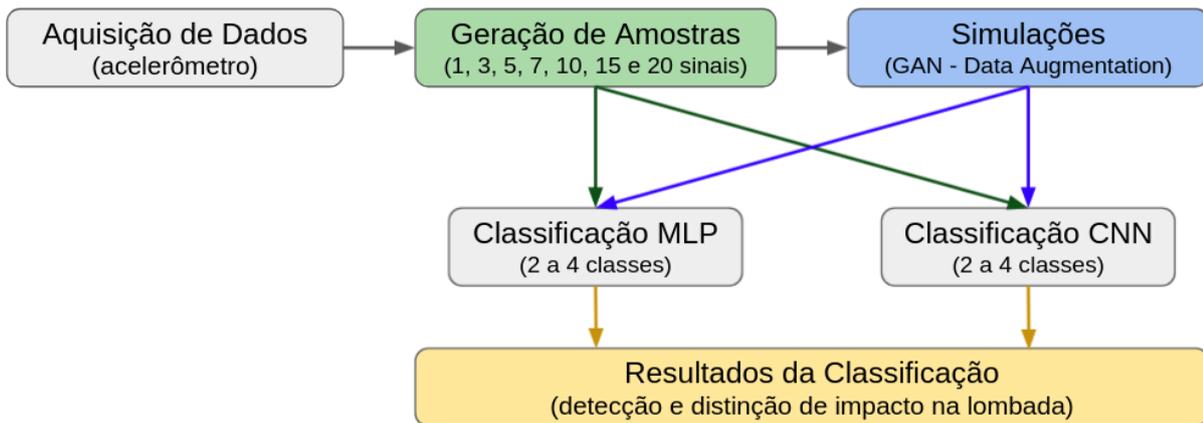
Fonte: Autor.

Esta experimentação é utilizada como processo de validação da aquisição de dados e portanto, apenas detecta impactos, não os classifica em tipos.

Os detalhes da aquisição de dados da experimentação inicial são apresentados na Seção 3.3.1, do processamento para classificação dos dados na Seção 4.1 e os resultados obtidos na Seção 5.1.

Na experimentação principal, existe um número maior de processos e estes são mais complexos, pois a classificação também engloba o tipo de veículo causador do impacto, além de possuir processo de aumento de dados (*data augmentation*) para gerar simulações de impactos artificiais (dados sintéticos).

Figura 2 – Diagrama de execução da experimentação principal.



Fonte: Autor.

Como pode ser observado na Figura 2, há uma geração de amostras entre os processos de aquisição e de classificação, o que demonstra que nesta experimentação os dados inseridos na rede neural não são 100% crus, embora haja um processo simples de amostragem.

É possível destacar que a complexidade da experimentação principal é aumentada vide o maior número de etapas, a comparação com o uso de dados sintéticos e a possibilidade de se distinguir o tipo de veículo causador do impacto.

A respeito da experimentação principal, maiores detalhes sobre o processo de aquisição dos dados são apresentados na Seção 3.3.2, sobre o processamento (geração de amostras, simulação e classificação) na Seção 4.2 e os resultados alcançados com os processos na Seção 5.2.

Por fim, as conclusões acerca do trabalho desenvolvido, assim como propostas e sugestões para extensão e estudos futuros são apresentados no Capítulo 6.

Os estudos acerca de trabalhos relacionados são apresentados na seção seguinte e, posteriormente, são pontuados os objetivos alcançados com o desenvolvimento deste trabalho.

1.1 Trabalhos Relacionados

Nesta seção são apresentadas breves citações de trabalhos que possuem relação direta com a tese desenvolvida quanto à semelhança na utilização de tecnologias e objetivos.

A utilização de *data augmentation* é difundida e pode ser encontrada em diversos tipos de dados. O trabalho de (WANG *et al.*, 2018) aplica redes GAN na geração de amostras de sinais UHF advindos de descargas parciais em isolação elétrica de alta tensão.

É concluído por Wang *et al.* (2018) que os classificadores utilizados no trabalho se beneficiaram das amostras sintéticas e a técnica aplicada pode ser de grande potencial industrial, principalmente em se tratando de dados esparsos.

Seguindo o mesmo objetivo (classificação de descargas parciais), Yijiang *et al.* (2018) aplica GAN para o aumento de dados e o compara com a aplicação de *Variable Noise Superposition* (VNS) e obtém resultados de melhora em seus classificadores para ambas as técnicas.

No trabalho de Eerdeken *et al.* (2020) o comportamento de equinos é classificado baseado em dados dos três eixos de acelerômetro (adquiridos com dispositivo vestível), primeiramente os dados crús são introduzidos no classificador e métodos de rotação das amostras são atribuídos aos dados fazendo assim com que os resultados de acurácia se mantenham acima dos 99% e tenham ganhos na generalização de amostras de teste (em torno de 0,3%).

Visando melhoria na segurança no uso de dispositivos móveis, Li, Hu e Zhou (2019) aplica em seu trabalho *data augmentation* (adição de ruído e escala) com o objetivo de melhorar a performance de um sistema de autenticação de usuário que é produzido (SensorAuth). Dotado de acelerômetro e sinais de movimento e vibração, a técnica alcança erro médio de 4,66% e a autenticação pode ocorrer em curto espaço de tempo, aproximadamente 5 segundos.

Um interessante trabalho que utiliza redes neurais convolucionais (CNN), LSTM, *data augmentation*, dados de acelerômetro e giroscópio é desenvolvido por Tufek *et al.* (2020), onde são analisados dados para que se possa fazer o reconhecimento de ações humanas baseadas em movimento (como correr, subir escadas, caminhar, entre outras). Com este trabalho alcança-se 99,0% de acurácia na classificação das ações, resultado promissor se comparado ao estado da arte.

No que diz respeito à veículos, a maioria dos trabalhos que utilizam dados de acelerômetro buscam classificar:

- A condição de pavimentos, como Dey *et al.* (2019) que utiliza acelerômetro de *smartphone* e IA para a classificação dos dados, entre outros (Gueta; Sato, 2017), (Gunawan; Yanfi; Soewito, 2015), (SALAU *et al.*, 2019);
- O comportamento de motoristas (VAITKUS; LENGVENIS; ŽYLIUS, 2014), (SERTTAS; GEREK; HOCAOGLU, 2019); e
- O tipo de veículo, baseado em imagens (CHEN *et al.*, 2017).

Outros métodos muito utilizados para a classificação de veículos e que utilizam acelerômetros em seu processo de aquisição de dados, acoplam os dispositivos no próprio veículo, como no trabalho de Tran, Luu e Nguyen (2010), que tem como objetivo fazer a leitura dos dados de aceleração e vibração e posterior classificação, visando monitoramento das condições de rodovias.

Este tipo de abordagem traz uma limitação, pois para cada veículo é necessário que haja um acelerômetro pronto para adquirir os dados, diferentemente do trabalho apresentado nesta tese onde apenas um acelerômetro é utilizado e consegue capturar, mesmo que em menor quantidade, dados de diversos veículos.

Embora a grande maioria dos trabalhos de classificação de tipos de veículos seja baseado em imagens (fotos ou vídeos) dos próprios veículos, o trabalho desenvolvido por Obertov e Andrievsky (2014) utiliza uma série de acelerômetros posicionados na superfície de uma rodovia e, a partir da leitura das vibrações, classifica veículos entre: carros, carros com trailer, caminhões e caminhões com trailer.

Apesar do artigo publicado por Obertov e Andrievsky (2014) ser um trabalho intimamente relacionado ao desenvolvido nesta tese, o uso de várias etapas de pré-processamento e engenharia de atributos, além de obter classificação utilizando-se de limiares calculados sem o uso de inteligência artificial, os resultados alcançados foram de 89% de acurácia média.

Apesar de haver uma gama de trabalhos relacionados que serviram como base para o direcionamento desta tese, há de se destacar algumas originalidades:

- A busca por soluções utilizando apenas um acelerômetro e um obstáculo (neste caso, a lombada);
- O estudo acerca do uso de redes neurais, inclusive fazendo-se uso de arquiteturas de *deep learning*, para analisar o comportamento das vibrações causadas pelos impactos dos veículos no obstáculo;
- O estudo sobre a geração de simulações aplicadas ao problema específico de vibrações de impacto de veículos no obstáculo;

Para melhor pontuar o objetivo geral e os específicos deste trabalho, a próxima seção traz uma descrição dos itens a serem desenvolvidos e posteriormente debatidos nas conclusões finais.

1.2 Objetivos

De maneira geral, o objetivo deste trabalho é fazer uma análise aprofundada do comportamento da vibração causada pelo impacto de veículos em um obstáculo utilizando

inteligência artificial.

Os objetivos específicos deste trabalho são:

- Desenvolver um sistema de aquisição de dados que possa ser utilizado para armazenar informações de vibração geradas a partir do impacto de veículos em um obstáculo;
- Detectar as vibrações causadas no obstáculo e classificá-las de acordo com o tipo de veículo causador do impacto;
- Gerar simulações de impactos, a fim de aumentar a quantidade de dados a serem analisados;
- Apresentar classificação de impactos com e sem a utilização de dados sintéticos (gerados de forma artificial);
- Comparar os resultados obtidos com e sem a utilização de dados sintéticos;
- Concluir sobre a qualidade da classificação com e sem a utilização de dados sintéticos;
- Avaliar o impacto da adição dos dados sintéticos nos dados reais de vibrações adquiridos pelo sistema de aquisição desenvolvido; e
- Proporcionar resultados promissores das análises executadas a fim de permitir que novas aplicações possam se beneficiar destes.

2 APRENDIZAGEM DE MÁQUINA

Neste capítulo é apresentada uma revisão bibliográfica das técnicas de aprendizagem de máquina que são necessárias para o entendimento do restante deste trabalho.

Primeiramente, uma introdução à aprendizagem de máquinas é apresentada, detalhando algumas técnicas e a evolução da área, com foco na proposta deste trabalho: classificação de padrões.

Na segunda seção, são apresentadas as arquiteturas de redes neurais artificiais clássicas, que possuem limitação na utilização de profundidade de camadas e processamento de dados sem a utilização de engenharia de *features*.

Na terceira seção são apresentadas algumas técnicas avançadas de aprendizagem de máquina que podem aprender padrões e classificá-los sem a utilização (ou com esta reduzida) da etapa de extração de características. Estas arquiteturas possuem a habilidade de aprender sozinhas quais as características do sinal são mais relevantes para a classificação de padrões pretendida.

2.1 Introdução

O termo Aprendizagem de Máquina faz referência à aprendizagem com experiências passadas, para melhorar o desempenho futuro. Esta área de estudo busca desenvolver métodos e algoritmos que, por meio de dados existentes, possam mapear as ações futuras ou generalizar o conhecimento adquirido para novos dados (DAS; BEHERA, 2017).

No campo das aplicações, são inúmeras as contribuições das técnicas de aprendizagem de máquina. De acordo com LeCun, Bengio e Hinton (2015), Silva, Spatti e Flauzino (2010) e Hadad (2019), entre muitas outras, destacam-se:

- Visão computacional;
- Processamento de língua natural;
- Melhorias em sistemas elétricos;
- Diagnósticos na área médica;
- Entretenimento;
- Processamento de sinais.

Com a evolução do hardware dos computadores, as arquiteturas e os algoritmos de aprendizado de máquina têm ficado cada vez mais complexos, alcançando assim melhores resultados nas tarefas em que são aplicados (LECUN; BENGIO; HINTON, 2015).

É comum a divisão dos algoritmos de aprendizagem de máquina aplicados à classificação de padrões em três tipos:

- Supervisionado;
- Não-supervisionado; e
- Semi-supervisionado.

No aprendizado supervisionado, há uma etapa de treinamento para ajuste do classificador. Esta etapa consiste em criar um modelo matemático que possa ser aplicado em problemas similares àqueles aprendidos pela máquina (SILVA; SPATTI; FLAUZINO, 2010).

No aprendizado não-supervisionado, não há conhecimento dos rótulos. Em outras palavras, não se tem um conjunto de treinamento. Sem esta informação, os objetos a serem classificados precisam aprender as relações entre si e se separar em grupos (JAIN; DUBES, 1988), (ULLMAN *et al.*, 2014).

Os algoritmos de agrupamento de dados (em inglês: *clustering*) são os mais comuns em se tratando de aprendizado não-supervisionado (WITTEN; FRANK; HALL, 2011).

Especialistas de domínio e algumas técnicas de validação de grupos podem ser utilizadas para auxílio na interpretação do resultado do agrupamento. Vendramin, Campello e Hruschka (2010) apresentaram em seu trabalho alguns critérios de validação de grupos, assim como um comparativo entre os mesmos.

A relação entre os objetos geralmente é a medida de similaridade ou dissimilaridade entre os mesmos, fazendo assim com que os algoritmos ajam agrupando aqueles mais próximos (similares) e mantendo os mais distantes (dissimilares) em grupos distintos (JAIN; DUBES, 1988).

Uma técnica de aprendizado não-supervisionado atual e muito utilizada é chamada de *autoencoder*. Esta técnica tem como objetivo codificar dados em uma menor dimensão, para que estes possam ser utilizados em classificadores ou outros tipos de algoritmos. Esta técnica é melhor descrita posteriormente neste capítulo.

O aprendizado semi-supervisionado acontece quando se tem uma grande quantidade de dados para serem classificados e apenas uma porção destes possui rótulos. Primeiramente o aprendizado não-supervisionado é aplicado a todos os dados para posteriormente utilizar os rótulos para fazer os ajustes finais (*fine-tuning*) do modelo de classificador (GOODFELLOW; BENGIO; COURVILLE, 2016).

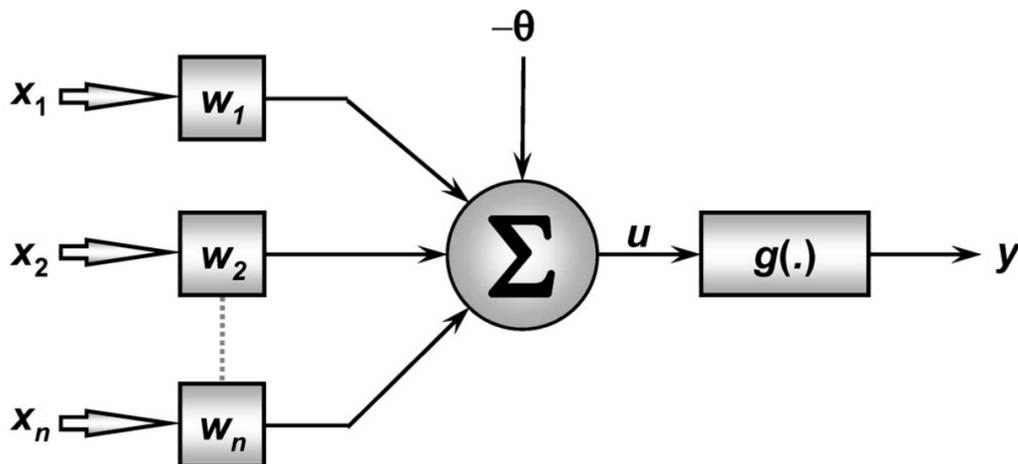
2.2 Redes Neurais Artificiais

Redes neurais artificiais são modelos computacionais baseados no comportamento e estrutura do cérebro humano. Composta por um ou mais neurônios artificiais, dispostos em uma ou mais camadas, que trocam ou não sinais entre si, as redes são dotadas de inteligência artificial, ou seja, aprendem com um conjunto de sinais de entrada (amostras de treinamento) e procuram generalizar este aprendizado para todo o universo de sinais de entrada (SILVA; SPATTI; FLAUZINO, 2010).

As redes neurais artificiais datam de mais de 50 anos desde a primeira publicação a seu respeito, a arquitetura do *Perceptron* simples. Porém, só a partir dos anos 1990 foram amplamente estudadas, tornando-se uma poderosa ferramenta para o desenvolvimento de sistemas inteligentes e resolução de problemas utilizando inteligência artificial (HAYKIN, 2009), (SILVA; SPATTI; FLAUZINO, 2010).

A composição de um neurônio artificial é apresentada na Figura 3. Este neurônio é baseado na arquitetura do *Perceptron* simples, rede neural que possui apenas um neurônio e só tem a capacidade de classificar padrões em problemas linearmente separáveis.

Figura 3 – Exemplo de neurônio artificial (*perceptron* simples).



Fonte: (SILVA; SPATTI; FLAUZINO, 2010).

Analisando o neurônio exibido na Figura 3, tem-se:

- $X_{1..n}$ - sinais de entrada;
- $W_{1..n}$ - pesos sinápticos;
- θ - limiar de ativação;
- Σ - combinador linear;

- u - potencial de ativação;
- g - função de ativação;
- y - sinal de saída.

Para se fazer uma analogia com este trabalho, os sinais de entrada seriam os valores das vibrações lidas dos acelerômetros, ou alguma característica que fora extraída desses sinais.

Os pesos sinápticos e o limiar de ativação são ajustados durante o treinamento da rede neural, são estes parâmetros que farão a rede "aprender" durante o processo de leitura das entradas.

O combinador linear soma todos os sinais que são provenientes da multiplicação da entrada X com os pesos W . Por fim, o resultado desta combinação é aplicada a função de ativação, que traz o valor tido como saída da rede, o qual será comparado com a saída desejada (na etapa de treinamento) e o valor do erro é utilizado para reajustar os pesos, fazendo assim com que a função de erro aplicada à saída da rede perante a saída desejada seja minimizada (SILVA; SPATTI; FLAUZINO, 2010).

Entre as décadas de 1970 e 1980, muitas arquiteturas de redes neurais foram criadas, como:

- LVQ (*Learning Vector Quantization*);
- Hopfield;
- Kohonen;
- ART (*Adaptive Resonance Theory*).

Após uma estagnação de alguns anos, a criação destas arquiteturas incentivaram o desenvolvimento e a publicação dos estudos e pesquisas sobre redes neurais artificiais, assim como a aplicação das mesmas em diferentes cenários (SILVA; SPATTI; FLAUZINO, 2010).

Após a descoberta do algoritmo *backpropagation*, as redes neurais voltaram à tona. Este algoritmo é utilizado para treinamento de redes neurais que possuem mais de uma camada, como a RBF (*Radial Basis Function*) e a MLP (*Multi-Layer Perceptron*) (HAYKIN, 2009).

De acordo com Silva, Spatti e Flauzino (2010) a MLP é a arquitetura de rede neural mais utilizada. É bastante difundida para a resolução de uma vasta série de problemas de engenharia entre eles:

- Classificação de padrões;
- Aproximação de funções;
- Otimização;
- Previsão de séries temporais.

Esta versatilidade trouxe para esta arquitetura de rede *status* de arquitetura de rede neural padrão (WITTEN; FRANK; HALL, 2011). É comum ver trabalhos que utilizam a MLP para validação de dados.

2.2.1 Redes Perceptron Multicamadas

Como exibido anteriormente, a rede *perceptron* simples possui apenas um neurônio. Deste modo, em um problema de classificação de padrões, observa-se que com apenas uma saída (binária, 0 ou 1), esta arquitetura só tem a capacidade de fazer a distinção entre dois diferentes padrões, e de forma linear (SILVA; SPATTI; FLAUZINO, 2010).

A rede MLP é derivada da rede *perceptron* simples, porém, pode conter diversos neurônios espalhados em diversas camadas (HAYKIN, 2009).

Uma das características da rede MLP é conter ao menos uma camada escondida de neurônios (camada intermediária)(HAYKIN, 2009). Um exemplo da topologia da MLP é exibido na Figura 4.

Pensando em classificação de padrões, na camada de entrada são introduzidos os sinais que a rede neural deverá classificar. Os neurônios das camadas internas mapeiam o espaço amostral, maximizando a saída (camada de saída) da qual o padrão pertence ou será classificado (SILVA; SPATTI; FLAUZINO, 2010).

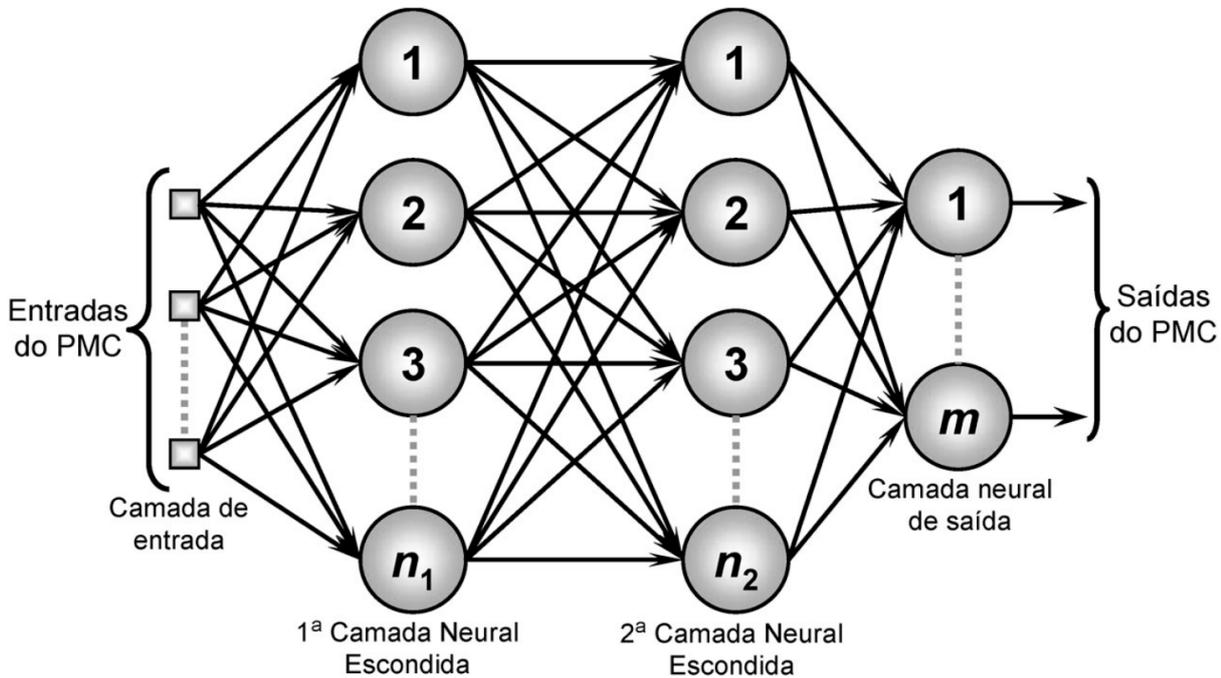
Se o objetivo for classificação de sinais de acelerômetro, em uma das abordagens pode-se ter como sinais de entrada os parâmetros extraídos por meio do processamento dos sinais. Na abordagem proposta por este trabalho, as entradas seriam a saída de um *autoencoder* ou uma rede convolucional, para então executar a classificação dos padrões.

A camada de saída deve conter neurônios suficientes para representar a quantidade de classes necessárias para a classificação. Geralmente, cada neurônio pode representar dois valores (zero ou um), e a combinação deles pode fazer com que alguns neurônios representem diversas classes (SILVA; SPATTI; FLAUZINO, 2010).

A abordagem mais comum é utilizar um neurônio para cada classe de saída. Dentre outras vantagens, há a de se poder utilizar funções de probabilidade nos valores de saída dos neurônios, como a *softmax* (BENGIO, 2009).

Uma discussão que é válida é a utilização de redes neurais com muitas camadas. Como a Figura 5 apresenta o aumento do número de camadas e seus efeitos na classificação

Figura 4 – Arquitetura de rede neural PMC (*Perceptron Multicamadas*), em inglês MLP (*Multi-Layer Perceptron*).



Fonte: (SILVA; SPATTI; FLAUZINO, 2010).

de padrões, pode-se observar que o aumento no número de camadas pode possibilitar o mapeamento de espaços de amostras mais complexos (SILVA; SPATTI; FLAUZINO, 2010).

O problema com o aumento de camadas é que o gradiente do erro que ajusta os pesos da rede neural é diminuído conforme é propagado para as primeiras camadas. Deste modo, o treinamento de redes com muitas camadas fica inviável (BENGIO, 2009).

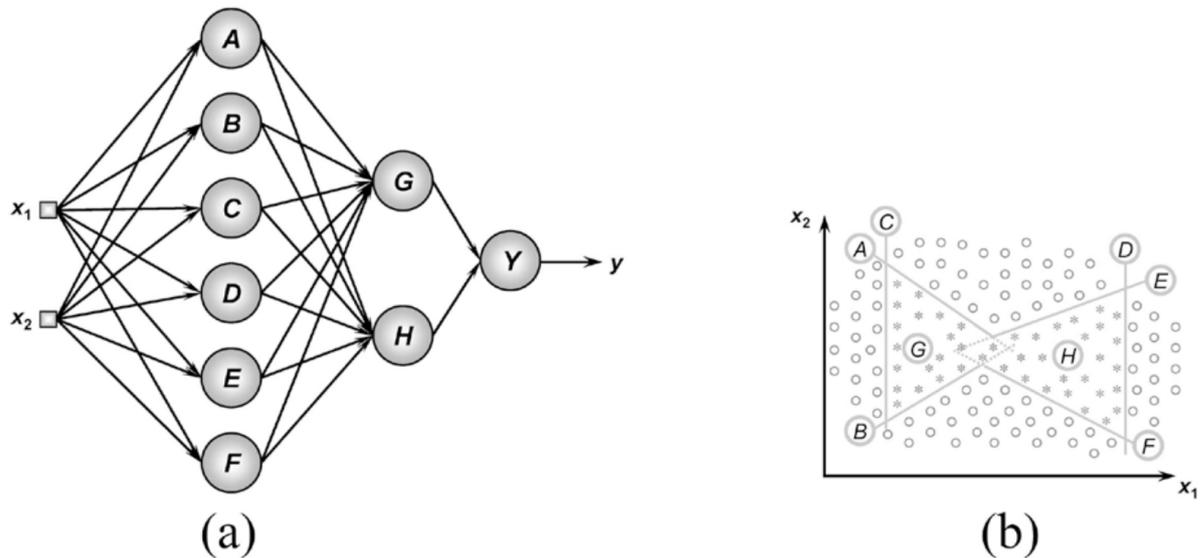
Alternativas a este problema foram desenvolvidas e são apresentadas na próxima seção.

2.3 *Deep Learning*

Após a primeira arquitetura de *deep learning* ser desenvolvida por Hinton, Osindero e Teh (2006), a possibilidade de se fazer abstrações de dados e construir máquinas de aprendizado mais poderosas levou alguns pesquisadores a procurar por maneiras de se mapear espaços amostrais mais complexos. Desta forma, a inclusão de mais camadas em modelos de aprendizagem de máquina baseados em redes neurais artificiais foi o caminho encontrado. (BUDUMA, 2017).

O aumento do poder computacional também foi fator relevante para que se pudesse efetuar o treinamento de modelos mais complexos, inclusive com o início da era de

Figura 5 – Comportamento das camadas em relação ao espaço amostral para problemas de classificação de padrões. Em (a) é exibida a topologia da rede neural e (b) exhibe o espaço amostral e a capacidade de separação fornecida pela topologia.



Fonte: (SILVA; SPATTI; FLAUZINO, 2010).

processamento de inteligência artificial utilizando GPU (*Graphics Processing Unit*).

As GPUs são preparadas para processamento gráfico (onde geralmente se tem muitas multiplicações de matrizes) e por este motivo são mais adequadas para o treinamento de redes neurais artificiais (NVIDIA, 2020).

Outro fator muito relevante para o desenvolvimento e utilização das arquiteturas de *deep learning* foram os *frameworks* de desenvolvimento de soluções, que são bibliotecas de código (geralmente *open source*) para facilitar o desenvolvimento de soluções em *deep learning*. Algumas delas são:

- Tensorflow (TENSORFLOW, 2020);
- Theano (THEANO, 2020);
- Keras (KERAS, 2020);
- DeepLearning4J (DEEPLARNING4J, 2020);
- Caffe (CAFFE, 2020)

Com a junção de hardware e software ideais para o desenvolvimento de modelos de aprendizado profundo, popularizou-se o *deep learning* como uma das mais procuradas técnicas para o desenvolvimento de soluções em inteligência artificial, inclusive retomando

debates sobre a ética e a segurança na utilização desta tecnologia. Este debate trouxe à tona uma carta aberta, que já fora assinada por diversos nome importantes da ciência (LETTER, 2017).

Dentre muitas arquiteturas desenvolvidas baseadas em *deep learning*, três são diretamente relacionadas ao desenvolvimento deste trabalho:

- LSTM (*Long-Short Term Memory*);
- CNN (*Convolutional Neural Networks*); e
- GAN (*Generative Adversarial Networks*).

Também é apresentada uma introdução sobre *Autoencoders* na seção 2.3.2, pois se trata de uma arquitetura promissora para a execução de procedimentos em trabalhos futuros.

2.3.1 *Long-Short Term Memory*

Para a análise e classificação de dados temporais utilizando redes neurais artificiais, a arquitetura mais indicada são as redes recorrentes (SILVA; SPATTI; FLAUZINO, 2010).

As redes neurais recorrentes (em inglês *Recurrent Neural Networks*, RNN) são arquiteturas que possuem, com exceção de sua primeira entrada, um processo de retroalimentação (CALIN, 2020).

Por ser uma arquitetura difundida e muito aplicada à problemas avaliando séries temporais (KANG, 2017), a arquitetura de RNN escolhida para a análise dos dados referentes ao primeiro experimento é a LSTM (*Long-Short Term Memory*).

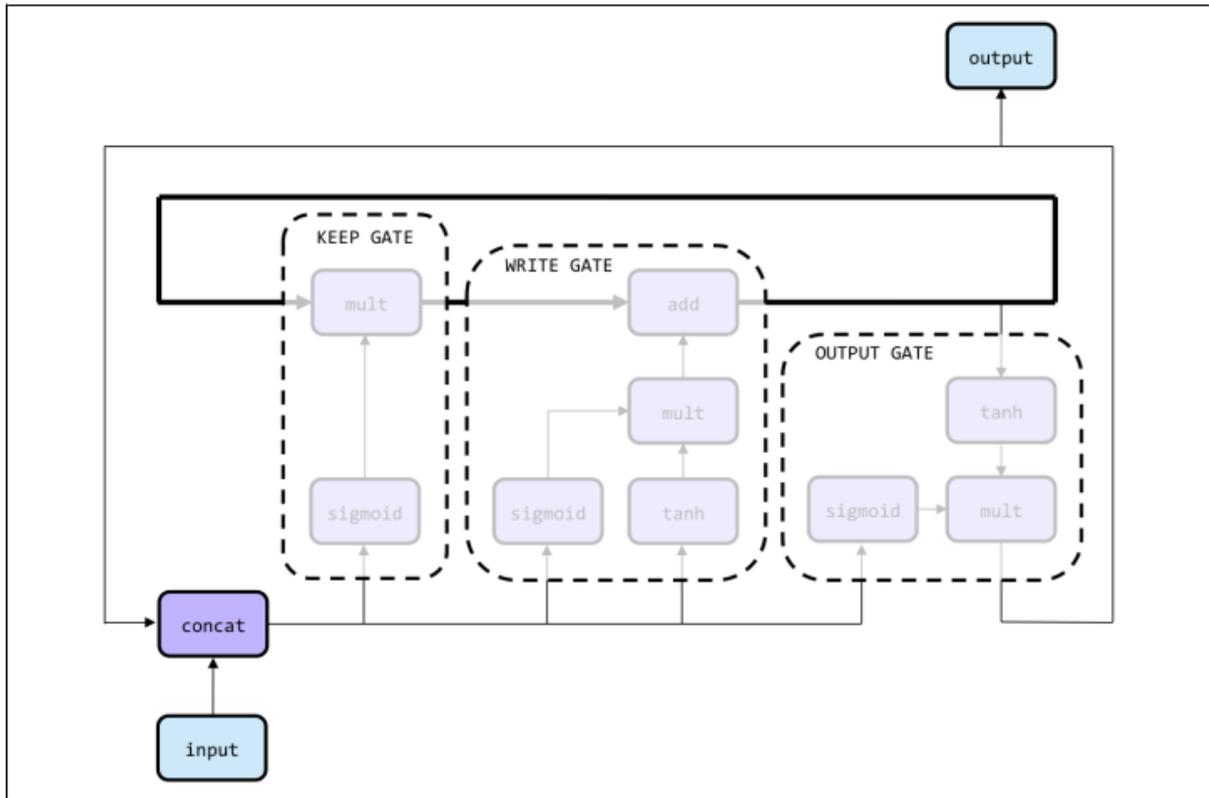
Esta arquitetura tem como objetivo ser capaz de armazenar, além de uma memória de curto prazo, comum em arquiteturas neurais, dados de longo prazo. Em outras palavras, a rede é capaz de manter valores importantes aprendidos em etapas da análise temporal por uma maior quantidade de etapas futuras (KANG, 2017).

Seu funcionamento se dá por execução das células que armazenam dados em memória, capazes de mapear séries temporais de maior dimensão. A Figura 6 traz um exemplo simplificado do funcionamento interno de uma célula LSTM.

Observa-se na Figura 6 que, de forma menos detalhada, cada célula possui 3 *gates* e cada um tem uma função bem definida.

O *keep gate* é o responsável por manter dados de memória de longo prazo, fazendo assim com que a rede seja capaz de perdurar padrões da série temporal que possam ter sido assimilados há mais tempo. Seu funcionamento analisa o próximo passo da rede e

Figura 6 – Arquitetura simplificada de uma célula da rede neural LSTM.



Fonte: (BUDUMA, 2017).

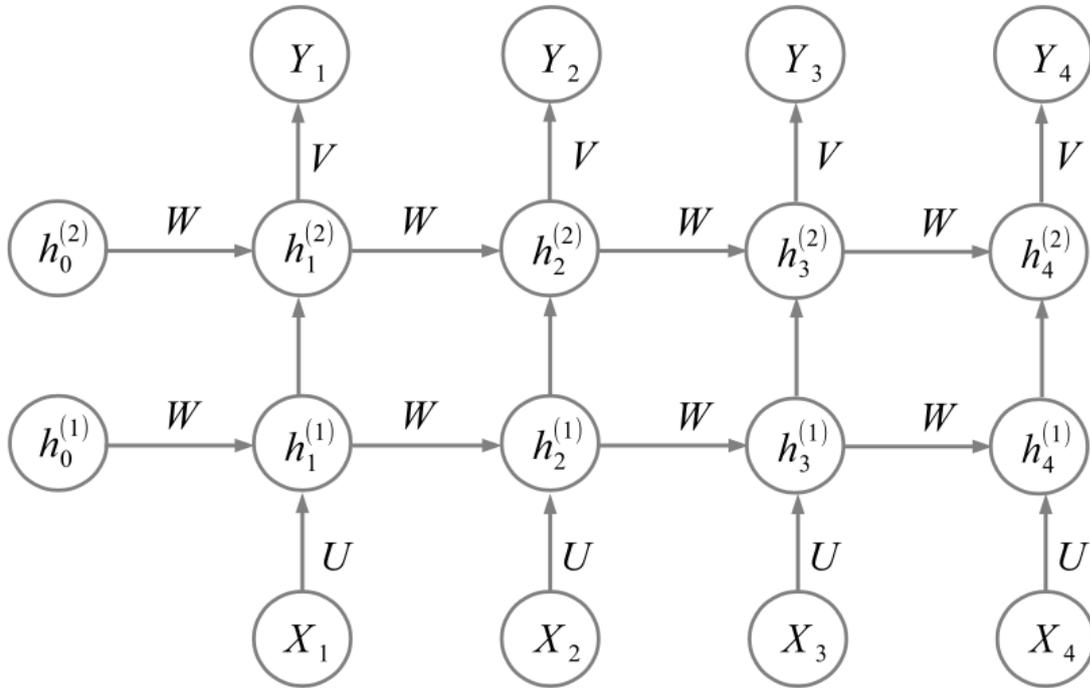
calcula uma matriz binária, que é multiplicada pelos dados já armazenados. Onde a matriz for 1, o valor é mantido e onde for 0, é apagado (BUDUMA, 2017).

O papel do *write gate* é calcular os novos valores e fazer alterações naqueles espaços de memória liberados pelo *keep gate*, gerando assim melhoria no processo de aprendizagem sem danificar dados de longo prazo que possam ser importantes. No caso do *output gate*, uma saída do estado atual da rede é calculada, a fim de representar os dados contidos na memória e sua atuação na série temporal (BUDUMA, 2017).

Por se tratar de uma arquitetura *deep learning*, é possível empilhar células LSTM, fazendo com que a rede neural seja processada de forma profunda, como exibido na Figura 7.

No esquema apresentado na Figura 7 pode-se observar as células LSTM recebendo os valores X_t e posteriormente estes sendo retroalimentados em novas células, como h_t entre as camadas (neste caso são duas). Por final, cada tempo produz uma saída Y_t referente ao estado da rede em cada etapa.

Figura 7 – Utilização de LSTM em camadas.



Fonte: (CALIN, 2020).

2.3.2 Autoencoders

Os *autoencoders* são uma classe específica de rede neural artificial que, por sua natureza, possuem a mesma dimensionalidade de entrada e saída (DENG; YU, 2014).

A dimensionalidade é a mesma, pois o objetivo desta rede neural, enquanto treinada, é fazer a cópia de sua entrada para sua saída (GOODFELLOW; BENGIO; COURVILLE, 2016).

A entrada dos dados são os sinais puros (crus) e a saída é a reconstrução deste sinal pela rede neural. Na camada interna da rede é onde o mapeamento da amostra de entrada acontece (GOODFELLOW; BENGIO; COURVILLE, 2016).

O mapeamento deve ser ótimo tal que, com uma dimensão muito menor que a entrada pura, possa-se reconstruí-la com acurácia suficiente para que a codificação mantenha os aspectos importantes do sinal de entrada (DENG; YU, 2014).

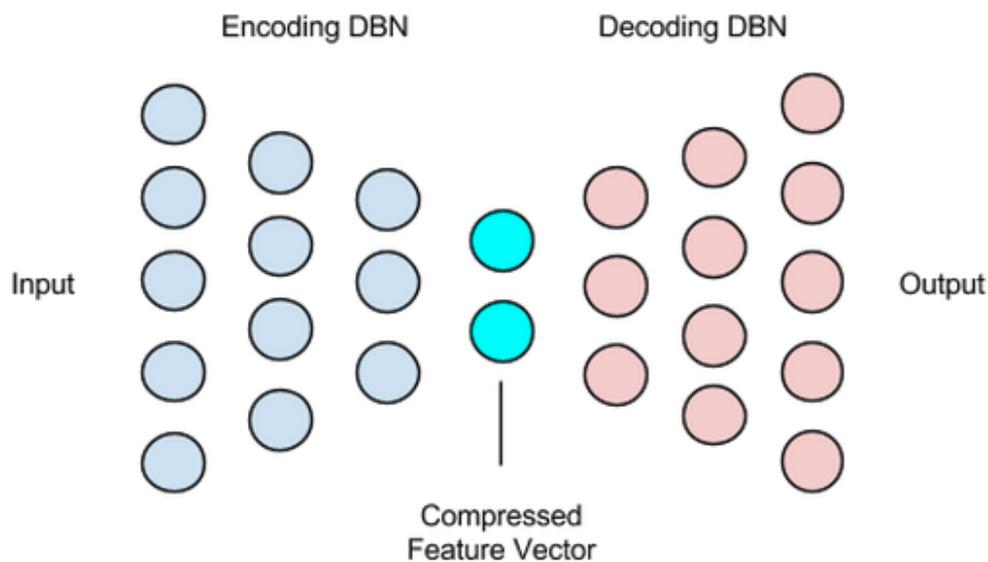
A Figura 8 exibe um exemplo de modelo de *autoencoder*, onde pode-se observar duas fases distintas e que tem funções específicas:

- *Encoding*; e
- *Decoding*.

Na etapa de codificação (*encoding*), a amostra crua é transformada em um vetor de características de menor dimensão (*compressed feature vector*) que a entrada original (*input*).

Na segunda etapa (*decoding*), partindo do vetor de características, a rede neural provê uma saída tal que o erro seja minimizado. A minimização do erro se dá com a comparação entre as entradas (*input*) e a saída (*output*). Porém, as entradas e as saídas são os mesmos sinais. Portanto, a saída da rede procura minimizar a diferença entre a entrada original e a reconstruída (GOODFELLOW; BENGIO; COURVILLE, 2016).

Figura 8 – Exemplo de arquitetura padrão para *autoencoders*.



Fonte: (DEEPLARNING4J, 2020).

O vetor de características criado por esta arquitetura de rede neural tende a conter as informações que são mais relevantes para a recriação do sinal de entrada (BUDUMA, 2017).

São duas as principais aplicações dos *autoencoders*:

- Compressão de dados; e
- Codificador de dados.

Outra técnica de *deep learning* e que sua utilização faz relação com o trabalho a ser desenvolvido são as Redes Neurais Convolucionais (CNN, *Convolutional Neural Networks*), explicadas a seguir.

2.3.3 Convolutional Neural Networks

As Redes Neurais Convolucionais (CNN, *Convolutional Neural Networks*) são arquiteturas que se utilizam de camadas de convolução dos dados para a extração de características de forma não-supervisionada (GOODFELLOW; BENGIO; COURVILLE, 2016).

Ainda segundo Goodfellow, Bengio e Courville (2016), basta a utilização da operação de convolução nos dados a serem analisados ou em uma camada da rede neural onde seja possível aplicar uma máscara convolucional para que essa seja considerada uma CNN.

A aplicação mais comum de CNNs é em processamento de imagens. Seu poder de extração de características pode ser bem explorado por matrizes de pontos, que é o caso de imagens em duas dimensões (LECUN; BENGIO; HINTON, 2015).

A utilização de redes profundas em conjunto com camadas convolucionais mostrou-se útil na extração de características e classificação de imagens, como pode ser observado com maiores detalhes no trabalho desenvolvido por Krizhevsky, Sutskever e Hinton (2012).

Em seu trabalho, Krizhevsky, Sutskever e Hinton (2012) utilizaram um total de 1,2 milhões de imagens, que foram classificadas em 1.000 diferentes classes, aplicando extração automática de características por meio de CNNs.

Geralmente, as CNNs são utilizadas para o processamento de sinais multidimensionais. A Figura 9 exibe um exemplo de operação de convolução em uma matriz de duas dimensões.

Analisando a Figura 9, é possível destacar os seguintes atributos:

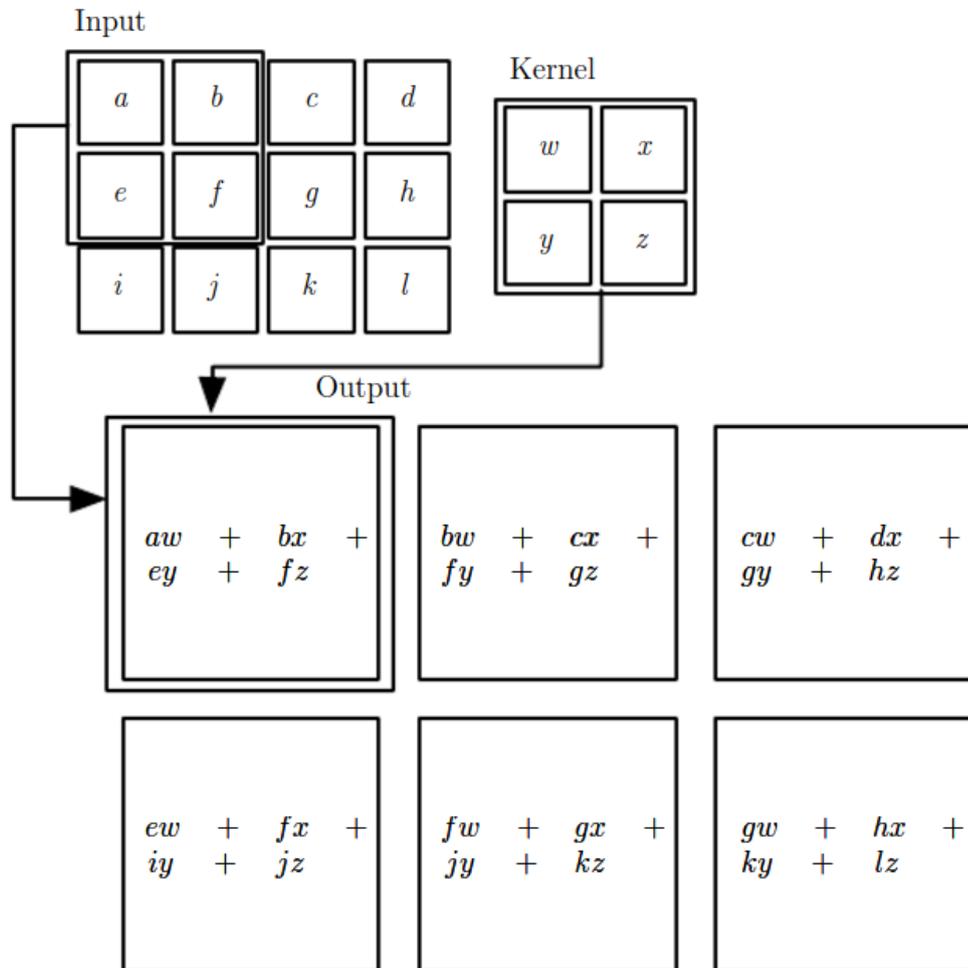
- A matriz de entrada (*input*) possui dimensão de 4×3 ;
- A máscara de convolução (*kernel*) possui dimensão de 2×2 ; e
- A saída da convolução (*output*) possui dimensão de 3×2 .

Sendo assim, observa-se que o resultado da convolução diminuiu a dimensão da matriz original. Esta diminuição se dá pelo tamanho da máscara que, quanto maior, menor a saída da camada de convolução.

É comum que sejam aplicadas várias máscaras de convolução nas entradas de uma rede. Os valores das células das máscaras é que são ajustados pela etapa de treinamento. Portanto, quanto mais máscaras, mais parâmetros para serem ajustados (LECUN; BENGIO; HINTON, 2015), (GOODFELLOW; BENGIO; COURVILLE, 2016), (KERAS, 2020).

As saídas da camada de convolução passam por uma função de ativação, assim como a saída de camadas de neurônios em uma rede MLP (GOODFELLOW; BENGIO; COURVILLE, 2016).

Figura 9 – Exemplo de operação de convolução em uma matriz 2D, com máscara de 2x2.



Fonte: (GOODFELLOW; BENGIO; COURVILLE, 2016).

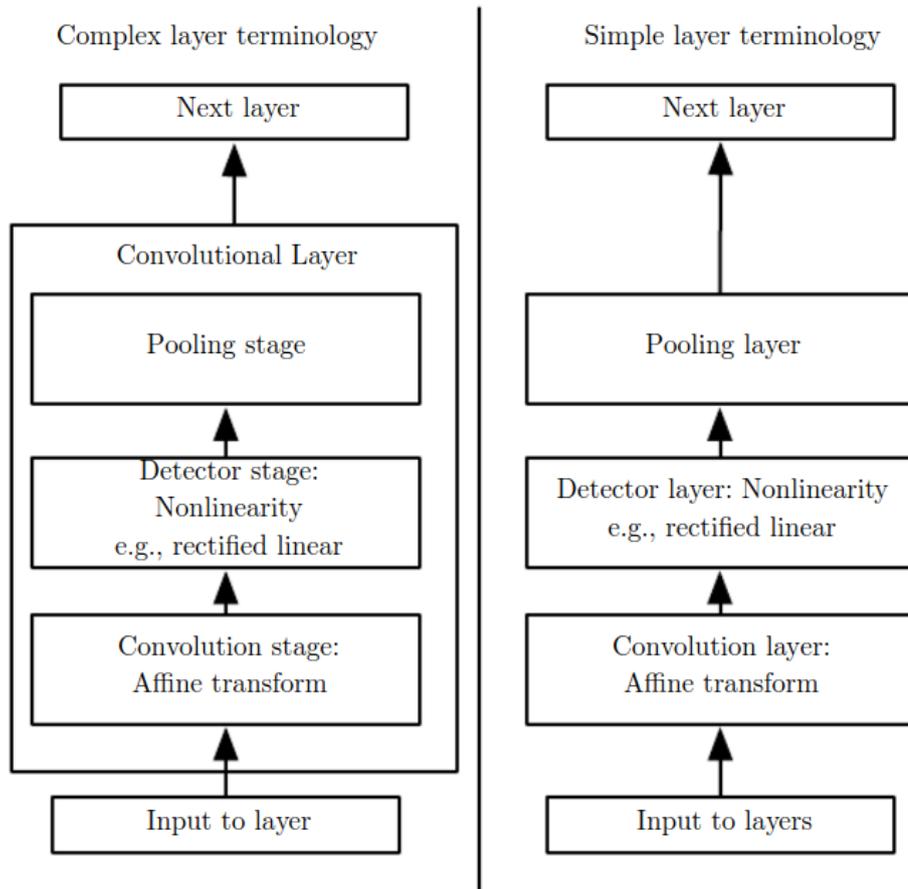
Uma segunda etapa da camada de convolução é chamada de *pooling*. Nesta etapa, apenas alguns parâmetros extraídos das máscaras de convolução são selecionados e posteriormente passados para as próximas camadas da rede neural (NIELSEN, 2015).

É comum utilizar na camada de *pooling* a função *max-pooling*. Esta seleciona, dentro de uma janela, apenas o valor máximo que foi extraído da entrada utilizando as máscaras convolucionais e, apenas este valor máximo é repassado para a próxima camada da rede (NIELSEN, 2015).

A Figura 10 exibe uma camada de CNN completa. As camadas convolucionais podem ser empilhadas, fazendo assim com que as redes se tornem profundas (GOODFELLOW; BENGIO; COURVILLE, 2016), (NIELSEN, 2015), (KERAS, 2020).

Para o processamento de áudio utilizando camadas convolucionais em uma rede neural (CNN), pode-se:

Figura 10 – Camada convolucional em uma rede neural artificial.



Fonte: (GOODFELLOW; BENGIO; COURVILLE, 2016).

- Utilizar o sinal acústico e performar a convolução em uma dimensão; ou
- Imprimir o espectro do sinal de áudio em uma imagem de alta resolução e então efetuar a convolução na imagem de 2 ou 3 dimensões (depende se estiver utilizando a terceira dimensão da matriz para representar canais de cores) (DENG; YU, 2014).

A abordagem utilizada neste trabalho, que é descrita na Seção 4.2.3, se assemelha com o processamento e classificação de imagens em duas dimensões.

Apesar dos sinais adquiridos do acelerômetro possuírem três dimensões, as amostras geradas ao longo do tempo podem ser interpretadas como imagens 2D, como exibido de forma detalhada na Figura 21.

2.3.4 Generative Adversarial Networks (GAN)

As redes GAN foram introduzidas por Goodfellow *et al.* (2014) e consistem em uma arquitetura de rede neural artificial generativa (geradora) baseada em redes que trabalham

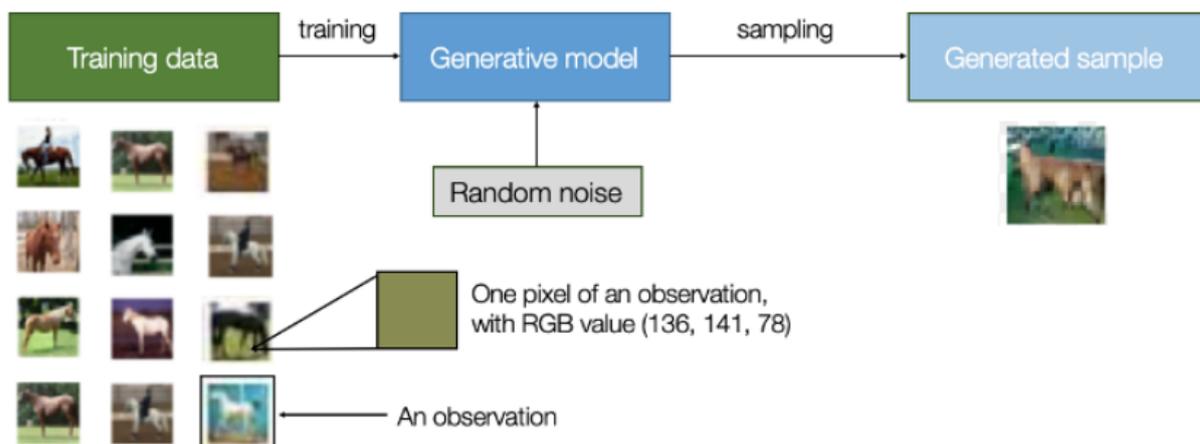
como adversárias.

A ideia básica é se criar um par de redes neurais com objetivos definidos:

- Rede geradora: essa rede tem como objetivo mapear dados de um determinado problema, fazendo com que uma entrada aleatória possa ser transformada nos dados reais a serem analisados;
- Rede discriminadora: seu objetivo é definir os dados sintéticos gerados pela rede geradora como dados que não são reais.

Para melhor explicar o funcionamento de cada uma das redes que fazem parte desta arquitetura, a Figura 11 apresenta o processo de uma rede geradora.

Figura 11 – Processo de execução de uma rede geradora (GAN).



Fonte: (FOSTER, 2019).

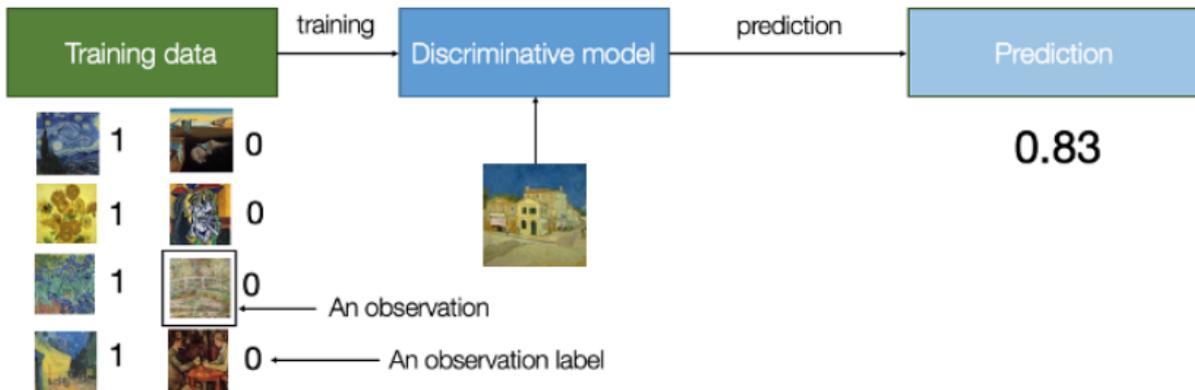
Observa-se que, além dos valores de dados reais, existe também um ruído (*noise*) que é entrada da rede. Este ruído é, geralmente, um conjunto de números aleatórios, os quais devem ser mapeados para dados que sejam semelhantes aos dados reais.

Em contrapartida, há a segunda rede (discriminadora) que avalia o resultado da rede geradora, trabalhando como um classificador binário, onde se oferece duas possíveis respostas:

- Números próximos de 0: a amostra gerada não se assemelha o suficiente com as amostras reais, portanto, é considerada uma amostra sintética; e
- números próximos de 1: a amostra se assemelha às originais (reais), portanto a rede discriminadora está considerando esta como uma amostra pertencente aos dados reais.

O processo de execução da rede discriminadora é exibido na Figura 12.

Figura 12 – Processo de execução de uma rede discriminadora (GAN).



Fonte: (FOSTER, 2019).

No exemplo apresentado na Figura 12, uma imagem é dada como entrada na rede (esta imagem foi previamente gerada pela rede geradora). A rede discriminadora afere a precisão da geração da amostra e dá como resultado o valor 0,83. Este valor é utilizado como erro para a rede geradora, que se ajusta e aprimora seu processo de geração de novos dados, visando maximizar a saída (enganar) da rede discriminadora.

Uma das principais aplicações para as redes GAN é o aumento de dados (em inglês *data augmentation*). Isto se dá pois é possível mapear dados reais gerando, simulando novas iterações e amostras sintéticas de forma inteligente (WANG *et al.*, 2018).

Neste trabalho, a arquitetura GAN é utilizada no processo de aumento de dados (*data augmentation*). Os detalhes deste processo são apresentados na Seção 4.2.1.

É possível também visualizar os resultados obtidos com a geração de amostras sintéticas, apresentados em imagens na Seção 5.2.1.

3 AQUISIÇÃO DE DADOS

Neste capítulo são apresentados, de forma detalhada, os métodos empregados para a aquisição dos dados utilizados no decorrer do trabalho.

Para a aquisição dos dados de vibração advindos do impacto de veículos em um obstáculo, foi necessário a confecção de uma lombada. De maneira artesanal, essa é desenvolvida com o objetivo de transferir a vibração do impacto para um acelerômetro, que se encontra fixado à mesma.

O desenvolvimento, medidas e parâmetros desta lombada são apresentados na seção 3.1.

Como segunda etapa para a aquisição dos dados, também é desenvolvido um aplicativo para *smartphone* Android. Esse é responsável por captar e armazenar, de maneira conveniente, as vibrações que ocorrem no acelerômetro acoplado ao *smartphone*, para posterior processamento e análise.

Sobre o aplicativo e suas funcionalidades, a seção 3.2 apresenta seus detalhes de desenvolvimento e funcionamento, assim como sua disponibilização para utilização pela comunidade.

Durante o processo de definição das tecnologias e estudos para otimização dos processos de aquisição de dados, foram testados dois tipos de acelerômetro:

- Acelerômetro de piezoeletreto;
- Acelerômetro acoplado a *smartphone*.

Após breve análise das vibrações capturadas por estes dois tipos de acelerômetro, definiu-se que o modelo acoplado ao *smartphone* é mais efetivo para a definição dos impactos.

O acelerômetro de piezoeletreto (exibido na Figura 13) foi desenvolvido por (ALVES *et al.*, 2020). Os testes são executados utilizando um osciloscópio digital (possível de se observar na Figura 16), mas não resultaram em aquisição da vibração causada pelo impacto do veículo.

A aquisição dos dados é executada em duas etapas. A primeira utilizando os dois tipos de acelerômetro (onde é possível observar a ineficiência do acelerômetro de piezoeletreto para esta finalidade) e a segunda apenas utilizando o acelerômetro acoplado ao *smartphone*.

Detalhes da metodologia e dos objetivos destas aquisições são exibidos na seção 3.3.

Figura 13 – Foto real do Acelerômetro de Piezoelétrico.



Fonte: Autor.

3.1 Lombada

A confecção da lombada se deu pela necessidade de padronizar um obstáculo capaz de gerar impacto na passagem de veículos. Desse impacto, surge a vibração que, quando medida, pode ser utilizada nas análises propostas. A Figura 14 apresenta uma imagem real da lombada confeccionada.

Totalmente feita em madeira e de forma artesanal, a lombada é pensada de forma a proporcionar vibração e transmiti-la ao sensor (acelerômetro). O material escolhido se deve à sua resistência a impactos, facilidade e leveza para transporte e por permitir a transmissão da vibração.

A lombada é formada por uma chapa superior reta e uma série de tacos para sustentá-la de forma semi-suspensa (Figura 14). Sem estes tacos, a vibração sairia prejudicada pois o movimento pós impacto seria absorvido pelo solo.

Figura 14 – Foto real da lombada que foi desenvolvida.



Fonte: Casati, Altafim e Altafim (2020)

Figura 15 – Foto aproximada do posicionamento dos tacos na parte inferior da lombada.



Fonte: Casati, Altafim e Altafim (2020).

A maneira com que os tacos são posicionados na parte inferior da lombada pode ser melhor observada na Figura 15. Essa imagem ilustra, de maneira aproximada, o posicionamento perpendicular que os tacos possuem em relação à chapa superior e o espaçamento entre eles.

No total são 11 tacos igualmente espaçados e de igual dimensão. As medidas da lombada podem ser melhor observadas no Quadro 1.

Ao analisar as medidas, pode-se observar que trata-se de um obstáculo de fácil transposição para veículos motores, bicicletas e pedestres. Outra observação pertinente é pontuar que o comprimento dos tacos ocupa na totalidade a largura da chapa principal, evitando que a lombada possa virar dependendo da posição, aceleração, velocidade e peso do veículo.

Após sua confecção, a lombada foi pintada de amarelo, para que se tornasse mais

Quadro 1 – Medidas da lombada (em centímetros).

Medida	Tamanho (cm)
Comprimento da chapa principal	283,0
Largura da chapa principal	18,8
Altura da chapa principal	1,5
Comprimento do taco	18,8
Largura do taco	3,0
Altura do taco	1,5
Espaçamento entre os tacos	25,0

Fonte: Casati, Altafim e Altafim (2020).

visível ao olho dos motoristas. A Figura 16 exibe a lombada pintada e posicionada para a aquisição de dados.

Na seção seguinte são apresentados detalhes do desenvolvimento e das funcionalidades do aplicativo *KaptAccel*, desenvolvido exclusivamente para a aquisição de sinais nos experimentos executados neste trabalho.

Figura 16 – A lombada confeccionada pintada e posicionada para fazer a aquisição dos dados.



Fonte: Casati, Altafim e Altafim (2020).

3.2 Aplicativo Móvel

Ao se utilizar um *smartphone* para adquirir dados advindos do seu acelerômetro, é necessário que um software possa interpretar seus sinais. Outro objetivo pertinente é o armazenamento das medidas de vibração e movimento para posterior utilização.

Por se tratar de um problema de classificação de padrões, algumas métricas, como as classes de seus sinais, também deveriam estar presentes nos dados. Por este motivo foi desenvolvido um aplicativo para *smartphone* Android.

A aplicação foi desenvolvida de forma nativa em linguagem Kotlin (KOTLIN, 2020). Dentre suas funcionalidades, pode-se citar:

- Leitura dos três eixos do acelerômetro (X, Y e Z);
- Armazenamento dos valores dos eixos e o momento em que a leitura é executada (*timestamp*);
- Seleção da frequência com que os dados são adquiridos;
- Exibição ou não em tempo real dos dados que estão sendo lidos e da quantidade de registros;
- Pause e Start na aquisição dos dados;
- Reinício na contagem dos registros lidos, apagando-se os que estiverem em memória;
- Seleção em tempo real da classe atual (tipo de vibração) para qual os registros estão sendo lidos;
- Persistência dos dados em formato CSV.

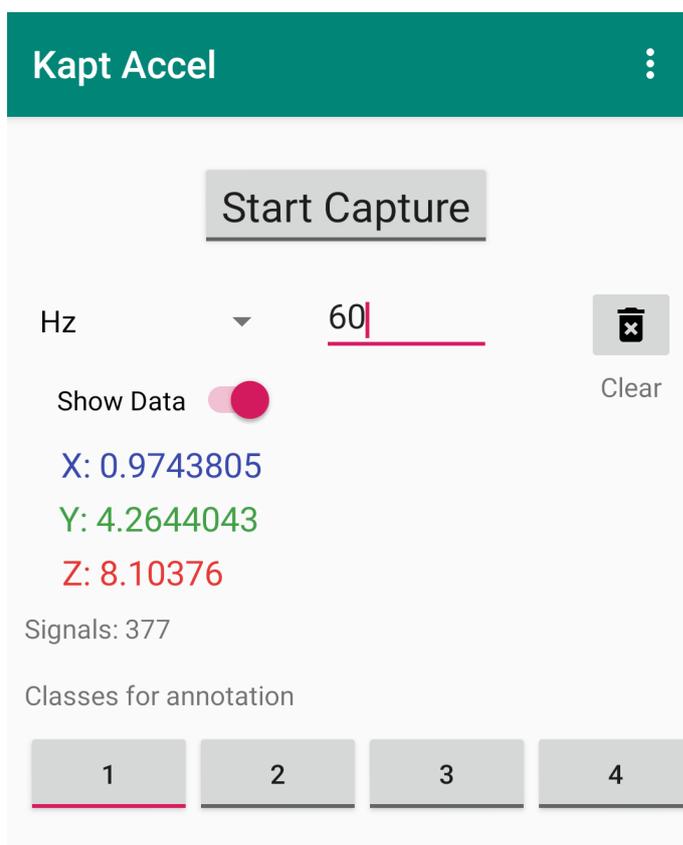
A Figura 17 apresenta a tela inicial do aplicativo desenvolvido. Nela é possível observar, com a distribuição dos componentes, algumas de suas funcionalidades.

É importante pontuar que, enquanto o aplicativo coleta os dados das vibrações, é possível utilizar os botões das classes (de 1 a 4), sinalizando que tipo de vibração está ocorrendo. Com este mecanismo, é possível distinguir entre até cinco diferentes classes (0, 1, 2, 3 e 4).

A disponibilização do aplicativo é feita na loja de aplicativos Google Play (GOOGLE, 2021). A Figura 18 representa o link de acesso para a instalação.

O arquivo CSV que é gerado pelo aplicativo possui cinco colunas (representando os tipos de dados) e cada linha representa um registro. Na primeira coluna é armazenado o momento (valor do *timestamp* em milissegundos), na segunda coluna o valor lido do eixo X do acelerômetro, na terceira e quarta colunas, os valores dos eixos Y e Z respectivamente e na quinta e última coluna, a classe (valor que pode variar de 0 a 4).

Na próxima seção são apresentados maiores detalhes da aquisição dos dados utilizando o aplicativo e suas funcionalidades.

Figura 17 – Tela principal do aplicativo *KaptAccel*.

Fonte: Casati, Altafim e Altafim (2020).

3.3 Experimentos de Aquisição

Neste trabalho, duas experimentações distintas foram executadas:

- Experimentação Inicial;
- Experimentação Principal.

A primeira teve como objetivo a validação dos dados advindos dos acelerômetros. Portanto, foram adquiridos dados de ambos os acelerômetros (piezoelétrico e *smartphone*).

Na segunda experimentação, a principal, apenas o acelerômetro do *smartphone* foi utilizado para a aquisição dos sinais de vibração. Seu objetivo consiste em uma análise mais profunda dos sinais.

A aquisição dos dados em ambas as experimentações é melhor detalhada nas seções seguintes.

Figura 18 – QR Code utilizado para acesso ao aplicativo KaptAccel.



Fonte: Casati, Altafim e Altafim (2020).

3.3.1 Experimentação Inicial

Publicado por Casati, Altafim e Altafim (2020), este experimento tem como principal objetivo a validação dos dados obtidos por meio dos acelerômetros utilizados.

O *smartphone* utilizado para a coleta dos dados foi um Motorola G8 *Plus*, dotado de sensor acelerômetro e o software KaptAccel.

Desenvolvido por (ALVES *et al.*, 2020), o acelerômetro de piezoeletreto utilizado no experimento foi preso à lombada e ligado a um osciloscópio digital *Agilent DSO-X 2002A*.

Os dados adquiridos via acelerômetro de piezoeletreto foram descartados do experimento por não possuírem variação suficiente para a distinção dos impactos, fato aferido já nas análises iniciais.

Considerando as informações coletadas pelo *smartphone* e armazenadas via KaptAccel, pôde-se criar 36.060 amostras de três dimensões. A frequência de coleta utilizada foi nativa do sistema operacional Android, que sofre variação dependendo da demanda de hardware. Em média, de 20 a 25 amostras por segundo foram armazenadas.

Por se tratar de um experimento de validação, apenas duas classes foram consideradas:

- 0: ausência de impacto de veículo;

- 1: presença de impacto de veículo.

Para melhor exemplificar o comportamento das amostras, o gráfico exibido na Figura 19 demonstra o comportamento de 500 amostras coletada sequencialmente durante o experimento.

Os três eixos (X, Y e Z) do acelerômetro são representados em diferentes cores e os degraus representam a classe (ausência ou presença de impacto).

Figura 19 – Gráfico que representa o comportamento de 500 amostras no tempo.



Fonte: Casati, Altafim e Altafim (2020).

Fazem parte das amostras com presença de impacto, passagens de veículos como motocicletas e carros de variados tamanhos e pesos. Esses fazem parte da classe 1, pois neste experimento não há diferenciação da categoria de veículo.

A continuação deste primeiro experimento é apresentada na seção 4.1.

No segundo experimento é executada uma coleta de dados mais robusta. Seus detalhes são apresentados e discutidos na seção seguinte.

3.3.2 Experimentação Principal

Com o objetivo de coletar amostras em maior quantidade e fazer a diferenciação entre tipos de veículos que produziram as vibrações, uma segunda etapa de aquisição dos sinais foi necessária.

Assim como na experimentação inicial, a lombada de madeira e o aplicativo KaptAccel foram utilizados, pois se mostraram efetivos na aquisição dos sinais de vibração, comprovados por meio de validação (apresentada em maiores detalhes nas seções 4.1 e 5.1).

Para que a segunda experimentação pudesse conter análises de maior complexidade, os sinais armazenados na aquisição de dados foram divididos em quatro classes. São elas:

- 0: ausência de impacto de veículo;
- 1: presença de impacto de veículo de médio porte.
- 2: presença de impacto de veículo de pequeno porte.
- 3: presença de impacto de veículo de grande porte.

Para melhor exemplificação das classes dos tipos de sinais que foram adquiridos, pode-se observar no Quadro 2 uma breve explicação de cada categoria.

Quadro 2 – Explicação detalhada das diferentes categorias de veículos analisadas na experimentação principal.

Tipo de veículo	Detalhamento
Veículo de pequeno porte	Motocicletas de qualquer tamanho, que possuam apenas duas rodas.
Veículo de médio porte	Carros de passeio e caminhonetes de pequeno porte, em sua maioria comportam de 2 a 5 passageiros.
Veículo de grande porte	Caminhonetes de grande porte, super utilitários, e pequenos caminhões de carga (2 eixos).

O procedimento utilizado para a aquisição dos dados foi manual. A cada passagem de veículo, era alterada a classe em vigência no aplicativo. Desta forma, foi criado um banco de dados supervisionado possuindo os três eixos do acelerômetro, o tempo em milissegundos e a classe referente à vibração em determinado momento.

Com a presença de mais classes (neste caso são 4), é possível se executar análises de maior complexidade, diferenciando não apenas a ausência e presença de impacto, como também o tipo de veículo que ocasionou a vibração.

A frequência de aquisição dos dados neste experimento foi definida para 25Hz, por se tratar de um número possível para o smartphone captar sem problemas de lentidão e falhas, além dos testes iniciais demonstrarem que, nesta frequência, é possível detectar os impactos por meio das vibrações.

O Quadro 3 permite observar números importantes sobre o processo de aquisição dos dados referentes ao experimento principal.

Os dados adquiridos podem ser processados posteriormente sem a necessidade de pré-processamento. Porém, é executado um procedimento de geração de amostras que

Quadro 3 – Números importantes sobre a etapa de aquisição de dados para a experimentação principal.

Número total de sinais adquiridos	152.193
Frequência de aquisição dos sinais (Hz)	25
Distância de tempo entre os sinais (milissegundos)	40
Número de eixos captados por vibração	3
Número de classes definidas	4

possuem maior quantidade de sinais. Com isto, pode-se avaliar a vibração do impacto em um maior período de tempo.

Maiores detalhes do pré-processamento e das técnicas utilizadas para as análises dos dados adquiridos são apresentadas no Capítulo 4.

4 PROCESSAMENTO DOS DADOS

Após a etapa de aquisição dos dados, estes são processados e analisados a fim de que os objetivos do trabalho sejam alcançados.

Este capítulo apresenta a metodologia utilizada para o processamento dos dados em ambos os experimentos (o inicial e o principal).

Na Seção 4.1 são apresentados detalhes do processamento dos dados que são utilizados como forma de validação do processo e na Seção 4.2 a metodologia empregada para a classificação e simulação de amostras.

4.1 Experimentação Inicial

O sistema desenvolvido para a aquisição dos dados utilizados nesta análise consiste em uma lombada com um smartphone dotado de acelerômetro acoplado. Esse sistema provê dados de entrada e saída, onde:

- Entrada: amostra com os 3 eixos do acelerômetro;
- Saída: Classe da amostra.

Neste experimento de validação dos dados, o objetivo é distinguir quando há (ou não) impacto de veículos na lombada. Para isto, foi utilizado um classificador inteligente.

Com o objetivo de analisar os dados de forma temporal, a arquitetura de rede neural LSTM foi escolhida por diversos motivos. De acordo com Kang (2017), ela provê capacidade de armazenamento de eventos a longo prazo e classificação de séries temporais *online*. Outra vantagem para uso da LSTM é trabalhar de forma robusta com amostras de dados sem pré-processamento (crús, em inglês *raw data*), assim como outras arquiteturas *deep learning* (BROWNLEE, 2018).

Além das vantagens da arquitetura em si, há também a disponibilidade de bibliotecas de código que implementam e proveem funções de otimização para o emprego desta tecnologia (KERAS, 2020).

A configuração utilizada para esta classificação é uma camada de rede neural LSTM com 32 células, suficientes para memorizar a quantidade de dados que será processada e variações, dada sua complexidade.

A camada de saída da rede desenvolvida possui apenas um neurônio, suficiente para distinguir entre as duas classes que estão sendo ponderadas (valores próximos de zero para ausência de impacto e valores próximos a um para presença de impacto).

A análise temporal utiliza diferentes tamanhos de entrada, que são a quantidade de amostras sendo analisadas num mesmo momento. Os testes aplicados utilizaram de 10, 20, 30, 40 e 50 amostras de entrada, para uma única saída.

Em outras palavras, para entradas de 10 amostras, seria necessário menos de 0,5 segundo para que o sistema indicasse se há ou não presença de impacto na lombada, enquanto com a utilização de 50 amostras como entrada, seriam necessários quase 2 segundos.

A LSTM é uma rede neural que trabalha de forma supervisionada. Neste caso, ela necessita de treinamento com amostras de entrada e suas respectivas saídas desejadas.

Para que haja amostras de treinamento e de teste (para aferição dos resultados), a divisão dos dados coletados é apresentada no Quadro 4.

Quadro 4 – Divisão das amostras para o experimento inicial de validação.

Tipo de Amostra	Porcentagem	Quantidade
Treinamento	70%	25.242
Teste	30%	10.818
Total de amostras	100%	36.060

Neste tipo de problema, a função de erro que desempenha melhor papel é o Erro Quadrático Médio (EQM), portanto, foi utilizada para o treinamento da rede.

A execução da etapa de treinamento se deu por 100 épocas. Buscando melhores resultados de classificação, o modelo treinado utilizado no teste foi resultante da época que desempenhou menor valor da função de erro.

Os dados utilizados na etapa de teste possuem a saída desejada, ou seja, as classes das amostras são conhecidas. Porém, são desconsideradas para a execução do teste na rede neural treinada.

O resultado de saída da rede neural provê a informação de que, se na entrada utilizada (contendo de 10 a 50 amostras) há ou não a presença de impacto de veículo. Este então é comparado com a classe original da amostra.

4.2 Experimentação Principal

Esta é a etapa mais importante do desenvolvimento do trabalho, onde são aplicadas diversas técnicas de inteligência artificial para explorar o potencial de classificação e análise dos dados obtidos (detalhes da aquisição são apresentados na Seção 3.3.2).

Os tamanhos de amostras selecionados são:

- 1 sinal;

- 3 sinais;
- 5 sinais;
- 7 sinais;
- 10 sinais;
- 15 sinais;
- 20 sinais.

Cada sinal é adquirido a cada 40 milissegundos. Assim, cada amostra corresponde a análise de um determinado tempo onde, pode-se classificar o tipo de impacto ocorrido dentro deste intervalo.

O Quadro 5 exibe detalhes informativos sobre os diferentes tamanhos de amostra criados a partir dos dados crus.

Quadro 5 – Divisão das amostras para o experimento inicial de validação.

Tamanho da amostra (quantidade de sinais)	Intervalo de tempo (milissegundos)	Quantidade de amostras
1	40	152.192
3	120	152.190
5	200	152.188
7	280	152.186
10	400	152.183
15	600	152.178
20	800	152.173

A redução na quantidade de amostras se dá devido à necessidade de se utilizar uma quantidade de sinais iniciais para a geração da primeira amostra. A partir desta, as demais são criadas utilizando o passo de 1 sinal, até que o algoritmo percorra todos os sinais da base de dados adquirida.

Por se tratar de dados supervisionados, para cada amostra é necessário que se atribua um valor de saída desejado. Este valor corresponde à classe da vibração representada pela amostra em questão.

O algoritmo utilizado atribui o valor mediano da classe dos sinais considerados para cada amostra, como pode ser observado com mais detalhes na Figura 20.

Para uma situação onde o número de sinais é ímpar, a classe desejada resultante é sempre a mediana, porém, em caso de amostras com número par de sinais, a classe atribuída é a posterior à mediana. Como exemplo, em uma amostra com 10 sinais, a classe

a ser atribuída como saída desejada desta amostra seria aquela correspondente ao sexto sinal.

Figura 20 – Atribuição de classe para saída desejada em amostras com mais de 1 sinal. Exemplo utilizando 5 sinais por amostra.

SINAIS	1	2	3	4	5
X	0.6	0.4	0.1	0.2	0.4
Y	0.9	0.7	0.5	0.6	0.8
Z	0.2	0.2	0.3	0.1	0.2
CLASSES	0	1	1	1	0

Fonte: Autor.

Após este pré-processamento dos dados, foi detectada uma grande quantidade de amostras de classe zero (amostras que correspondem à ausência de vibração). Como o intuito do trabalho é fazer a análise dos dados com pouca interferência de engenharia de *features* (deixar os dados o mais crú possível), os dados deveriam ser mantidos na base em grande quantidade.

Esta disparidade na quantidade de amostras de cada classe resultou em um problema: as classes que possuíam menos amostras eram muitas vezes tratadas como ruídos e, caso a complexidade do modelo utilizado para aprendizagem e classificação, o resultado era *overfitting*, proporcionando um valor de erro muito pequeno na etapa de treinamento, mas grande na etapa de teste.

Por este motivo, optou-se por gerar novas amostras de forma inteligente, simulando uma maior quantidade de veículos provocando vibrações. Sendo assim, o número de amostras de cada classe pode ser melhor balanceado, buscando mitigar o problema descrito anteriormente, evitando *overfitting*. Detalhes do processo de simulação de novas amostras são exibidos na Seção 4.2.1.

Na Seção 4.2.2, são utilizadas redes neurais Perceptron Multicamadas (MLP) como classificadores das diferentes amostras. Os procedimentos são executados com e sem a adição das amostras simuladas sinteticamente.

Por fim, ao se tratar de amostras de mais de uma dimensão, optou-se por efetuar uma análise utilizando Redes Neurais Convolucionais, processo detalhado na Seção 4.2.3.

4.2.1 Geração de Simulações

Com o objetivo de melhorar o processo de treinamento e classificação das amostras de impacto dos veículos na lombada, foi necessário fazer alterações na quantidade adquirida de cada tipo amostra.

O uso de *Data Augmentation* (em português: Aumento de Dados) possibilita simular novas amostras de dados sintéticas baseando-se na amostragem real (MIKOŁAJCZYK; GROCHOWSKI, 2018).

Neste trabalho, a arquitetura inteligente utilizada para gerar simulações é chamada de *Generative Adversarial Networks* (GAN), por se tratar de uma técnica reconhecida e vastamente utilizada para este fim (TRAN *et al.*, 2020).

Informações mais detalhadas sobre o funcionamento e aplicações das redes neurais GAN são exibidas na seção 2.3.4.

Por se tratar de uma arquitetura que possui duas redes neurais que fazem o papel de adversárias, duas redes distintas devem ser construídas. A primeira, chamada de gerador, mapeia o universo das amostras reais e a segunda verifica a qualidade deste mapeamento.

Cada tamanho de amostra (1 a 20 sinais) possui seu universo amostral independente, portanto, para cada um existe uma rede neural treinada para geração de amostras já com a dimensão correta para uso nos demais processos.

Optou-se por simular amostras já pré-processadas com seus tamanhos fixos e não sinais temporais contínuos, a fim de diminuir a complexidade das redes adversárias, mantendo-se apenas a arquitetura MLP.

Baseado no experimento desenvolvido por (BHATTIPROLU, 2020), que promove a geração de imagens simuladas em duas dimensões, foi desenvolvido o conjunto das duas arquiteturas adversárias utilizado neste trabalho.

Mesmo utilizando arquiteturas baseadas em projeto de simulação de novas imagens, optou-se por não utilizar redes convolucionais nesta etapa, por possuir algumas amostras na qual a dimensão não permite a utilização de máscaras convolucionais.

Com a necessidade de se utilizar uma entrada (ruído) que é transformada, durante o processo de geração de simulações, em novas amostras, foi utilizado neste trabalho um vetor de números aleatórios com 100 posições.

O objetivo do gerador é então mapear o vetor de entrada da rede em amostras de saída, com o intuito de se aproximar o máximo possível das amostras reais.

Após este mapeamento ser executado, a segunda arquitetura de rede neural (adversária do gerador) chamada discriminador, define se a amostra gerada pode se passar por real. Sendo assim, quanto melhor o mapeamento, mais dentro do espectro de amostras

reais as simulações se encaixam.

A aplicação de convolução nas amostras de maior dimensão, com o objetivo de classificação, é apresentada na seção 4.2.3.

As topologias de gerador e discriminador não necessitam ser as mesmas. Detalhes da topologia do gerador são apresentados no Quadro 6.

Quadro 6 – Topologia da rede neural geradora de simulações.

Camada da rede	Quantidade de neurônios
Entrada	Tamanho do ruído (100)
Camada 1	256
Camada 2	512
Camada 3	1024
Saída	Tamanho da amostra (número de sinais x 3)

Como pode ser observado no Quadro 6, a saída da rede geradora de amostras é sempre da dimensão da amostra que está sendo treinada. Portanto, para os diferentes tamanhos de amostras, são criadas topologias de treinamento distintas.

Uma das informações que se pode utilizar para medir a complexidade de uma rede neural é a quantidade de parâmetros treináveis que ela possui.

Esses parâmetros são os valores numéricos que podem ser ajustados durante a etapa de treinamento com o objetivo de ajustar a função, aproximando a saída da rede neural com a saída desejada (para dados supervisionados).

Por possuir alterações apenas nas extremidades da topologia da rede, a quantidade de parâmetros treináveis não sofre grandes variações com a mudança no tamanho da amostra. Considerando as amostras compostas por 20 sinais de acelerômetro, a quantidade de parâmetros treináveis é:

- Rede geradora: 747.836 valores;
- Rede discriminadora: 162.817 valores.

No caso dos discriminadores, foi também necessário criar uma topologia diferente para cada tamanho de amostra. O Quadro 7 apresenta a topologia utilizada neste tipo de rede onde pode-se observar que a entrada da rede neural depende também do tamanho da amostra que será avaliada.

A tarefa da rede neural discriminadora é simplesmente dar um parecer sobre cada amostra, por isso apenas um neurônio de saída. É um classificador binário onde:

- Saída próxima de 0: provavelmente a amostra é sintética (foi criada por um simulador);

Quadro 7 – Topologia da rede neural discriminadora de simulações.

Camada da rede	Quantidade de neurônios
Entrada	Tamanho da amostra (número de sinais x 3)
Camada 1	512
Camada 2	256
Saída	1

- Saída próxima de 1: provavelmente a amostra é verdadeira.

Para que as redes convergissem e a rede geradora pudesse criar amostras sintéticas que se passassem por amostras reais, foram necessárias 20.000 épocas de treinamento. Já a rede discriminadora foi treinada em duas etapas:

- Treinamento com entradas reais e saídas positivas;
- Treinamento com entradas sintéticas e saídas negativas;

Assim, é possível ensinar à rede discriminadora quais amostras que devem ser consideradas sintéticas e quais devem ser consideradas reais, enquanto a rede geradora melhora sua performance em enganar o discriminador.

Cada época consiste em uma etapa de treinamento para cada uma das redes. Detalhes e maiores informações sobre os parâmetros de treinamento das redes são exibidos no Quadro 8.

Quadro 8 – Parâmetros de treinamento da rede neural GAN.

Rede Geradora	
Épocas	20.000
<i>Momentum</i>	0,8
Taxa de aprendizagem	0,0002
<i>Batch</i>	128
Rede Discriminadora	
Épocas	20.000
<i>Momentum</i>	0,8
Taxa de aprendizagem	0,0002
<i>Batch</i>	64

A quantidade de amostras positivas (com classe de vibração definida) gerada é de 100.000 para cada uma. Desse modo, para cada uma das classes 1, 2 e 3 (impacto de diferentes tipos de veículos), existem 100.000 amostras sintéticas que podem ser utilizadas nos processos de classificação.

Na seção 4.2.2 são utilizadas primeiro apenas as amostras reais para a classificação das mesmas e em um segundo momento, é feita utilizando também as amostras sintéticas.

4.2.2 Classificação MLP

Nesta primeira etapa de classificação, considerando a segunda experimentação, foi utilizada uma rede neural Perceptron Multicamadas (*Multi-Layer Perceptron*). Por se tratar da arquitetura de rede neural mais difundida (SILVA; SPATTI; FLAUZINO, 2010), optou-se por utilizá-la para avaliar a utilização das amostras sintéticas geradas anteriormente.

Ao observar a grande disparidade na quantidade de amostras reais adquiridas (descrito na seção 3.3.2) entre as classes que correspondem à impactos e a classe que corresponde a não impacto, é pontuado que a possibilidade de ocorrência de *oversampling* é real.

Para constatar esta ocorrência, na etapa de classificação utilizando MLP são feitos dois experimentos distintos:

- Classificação das amostras utilizando dados reais (detalhado na seção 4.2.2.1);
- Classificação das amostras utilizando dados reais acrescidos dos dados sintéticos simulados na seção 4.2.1 (detalhado na seção 4.2.2.2).

As abordagens utilizadas na classificação das amostras para este experimento, utilizando ou não as amostras sintéticas, são:

- Dividindo as amostras em duas classes, onde a classe 0 é a ausência de impacto de veículo na lombada e a classe 1 é a presença de impacto de veículo de qualquer tipo;
- Dividindo as amostras em três classes, onde a classe 0 é a ausência de impacto de veículo na lombada, classe 1 é a presença de impacto de veículo de médio porte e classe 2 é a presença de impacto de veículo de pequeno porte;
- Dividindo as amostras em quatro classes, onde a classe 0 é a ausência de impacto de veículo na lombada, classe 1 é a presença de impacto de veículo de médio porte, classe 2 é a presença de impacto de veículo de pequeno porte e classe 3 é a presença de impacto de veículo de grande porte;

4.2.2.1 Dados Reais

Nesta etapa da experimentação principal, são utilizados apenas os dados coletados na etapa de aquisição descrita na seção 3.3.2, transformados em amostras de diferentes tamanhos (1 a 20 sinais).

Diversas topologias de rede neural foram testadas e, como trata-se de um processo que geralmente é executado de forma empírica, definiu-se como parâmetro de definição da melhor topologia o erro gerado durante o treinamento (*loss function*) combinado com a acurácia alcançada utilizando as amostras de teste.

A arquitetura de rede neural utilizada nesta etapa é a MLP e sua topologia é apresentada no Quadro 9. Pode-se observar variações:

- No tamanho da entrada: dependente do tamanho da amostra que está sendo analisada para a classificação;
- No tamanho da saída: dependente da quantidade de classes que estão sendo analisadas;

A quantidade de classes varia de acordo com os experimentos executados, considerando o método de aquisição apresentado na seção 3.3.2. As variações são:

- Considerando 2 classes:
 - 0: sem impacto;
 - 1: impacto de veículos das classes 2, 3 e 4;
- Considerando 3 classes:
 - 0: sem impacto;
 - 1: impacto de veículos da classe 1;
 - 2: impacto de veículos da classe 2;
- Considerando 3 classes:
 - 0: sem impacto;
 - 1: impacto de veículos da classe 1;
 - 2: impacto de veículos da classe 2;
 - 2: impacto de veículos da classe 3;

Quadro 9 – Topologia da rede neural MLP utilizada para classificação dos dados reais.

Camada da rede	Quantidade de neurônios
Entrada	Tamanho da amostra (número de sinais x 3)
Camada 1	20
Camada 2	20
Saída	Número de classes (2 a 4)

Os parâmetros de treinamento das redes neurais são apresentados no Quadro 10. A definição dos parâmetros ocorreu de forma empírica, mantendo como prioridade a minimização de valores da função de erro e maximização a acurácia.

A função de erro Entropia Cruzada Categórica (em inglês *Categorical Cross-Entropy*), ou CCE, é utilizada na etapa de treinamento para medir a qualidade dos pesos ajustados da rede neural para os dados de treinamento em questão.

Quadro 10 – Parâmetros utilizados para treinamento das redes neurais de classificação das amostras utilizando a arquitetura MLP.

Épocas	500
<i>Momentum</i>	não utilizado
Taxa de aprendizagem	0,01
<i>Batch</i>	20% das amostras

Na próxima seção são apresentados detalhes da etapa de classificação utilizando MLP com adição das amostras sintéticas.

4.2.2.2 Dados Simulados

Esta seção apresenta o método utilizado para a classificação de amostras de vibração dos impactos dos veículos na lombada utilizando as simulações geradas de forma inteligente (apresentada na Seção 4.2.1).

Para cada um dos experimentos já descritos na Seção 4.2.2.1 foram executados testes utilizando 100.000 amostras sintéticas adicionadas às amostras reais. Este processo tem como objetivos:

- Validar o processo de simulação inteligente (*data augmentation*) executado na Seção 4.2.1;
- Mitigar possível problema de *oversampling* dado à disparidade de volume das amostras de cada classe.

Com o objetivo de isolar os problemas que são alvo desta etapa do experimento, a topologia, parâmetros e demais variáveis (exclua-se o número de amostras) são mantidos como executados na Seção 4.2.2.1.

Na próxima Seção são descritos detalhes do processo de classificação utilizando CNNs, que também utiliza-se de amostras sintéticas adicionadas às reais.

4.2.3 Classificação Convolutacional

Com o objetivo de aprofundar a análise acerca dos dados coletados (Capítulo 3), este trabalho propõe, como uma de suas abordagens, a classificação das amostras utilizando redes neurais convolucionais.

O uso de redes neurais convolucionais para a classificação de padrões é vastamente utilizado em imagens e amostras que possuem muitas dimensões. Porém, sua aplicação pode-se dar em diversos tipos de amostras, como áudio, vetores de dados numéricos entre outros.

É comum a transformação de amostras em "imagens" de duas dimensões para que se possa executar convoluções e assim poder se utilizar desta arquitetura. No caso deste trabalho, as amostras geradas já possuem duas dimensões, como pode ser observado na 21.

Figura 21 – Exemplo de amostra real de tamanho 10, onde as colunas representam os eixos do acelerômetro e as linhas as capturas ao longo do tempo.

Sinais -> Tempo	X	Y	Z
T1	0.42495	0.47407	0.67661
T2	0.43196	0.49181	0.68706
T3	0.41591	0.49394	0.62335
T4	0.40642	0.50716	0.54941
T5	0.42541	0.50205	0.69387
T6	0.45886	0.49096	0.64782
T7	0.43739	0.46682	0.65047
T8	0.43400	0.48175	0.67036
T9	0.41501	0.50972	0.63892
T10	0.43264	0.47774	0.66226

Fonte: Autor.

Deste modo, as máscaras de convolução conseguem tratar a amostra como um espaço onde se pode navegar, extraíndo informações importantes acerca de seus valores.

Para que a topologia utilizada nesta etapa possa ter uma profundidade (número

de camadas) que permita treinamento e classificação eficazes, as amostras utilizadas são apenas as maiores (tamanhos 10, 15 e 20).

No Quadro 11 são apresentadas as informações sobre a topologia da rede neural convolucional utilizada.

Quadro 11 – Topologia da rede neural CNN utilizada para classificação.

Camada da rede	Detalhes
Convolução 1	Máscara de 3x3 e profundidade 16
<i>Max Pooling</i> 1	Máscara de 2x2
Convolução 2	Máscara de 2x2 e profundidade 16
<i>Max Pooling</i> 2	Máscara de 2x2
Saída	MLP: Número de neurônios é igual ao número de classes

A escolha da topologia descrita no Quadro 11 considera o valores de:

- Função de erro no treinamento: CCE;
- Acurácia no conjunto de testes.

A diminuição na complexidade da topologia aumentou os valores de erro e diminuiu a acurácia, enquanto a adoção de maior quantidade de camadas, profundidade ou diferentes máscaras não diminuiu o erro do treinamento e não aumentou a acurácia no conjunto de testes.

Para as camadas de convolução, as máscaras bidimensionais são de mesmo tamanho para as 16 convoluções. O *pooling* não sofre alteração quanto ao seu tamanho e a saída é dependente da quantidade de classes que se está analisando.

Nesta etapa, assim como na Seção 4.2.2, são utilizadas abordagens com 2, 3 e 4 classes, fazendo assim com que os resultados possam ser comparados entre as diferentes arquiteturas de redes neurais.

Para que se possa também testar a efetividade das amostras sintéticas na classificação dos tipos de impacto, os testes utilizando a rede convolucional também são executados em duas etapas:

- Utilização das amostras reais;
- Utilização das amostras reais com adição das amostras sintéticas.

Apesar do processamento ser efetuado com ambas as configurações de amostras, a ênfase desta etapa se dá com a utilização da amostragem sintética, gerada na Seção 4.2.1.

Assim como demonstrado na Seção 4.2.2.2, a quantidade de amostras sintéticas adicionadas é de 100.000 para cada uma das classes positivas.

Os resultados obtidos com o processamento das abordagens descritas neste Capítulo são apresentados e discutidos a seguir (Capítulo 5).

5 RESULTADOS E DISCUSSÃO

Após executados os procedimentos e análises apresentados no Capítulo 4, neste são exibidos e discutidos os resultados alcançados em cada experimento.

Para a experimentação inicial, busca-se o resultado de validação das amostras adquiridas para este fim.

Na experimentação principal, mais métodos são utilizados e há maior complexidade na interpretação dos resultados por se tratar de uma análise profunda, considerando maior quantidade de dados, número de classes, e fatores em geral.

5.1 Experimentação Inicial

Na Seção 4.1 foram exibidos os detalhes do processamento das amostras coletadas (Seção 3.3.1), sua metodologia e técnicas utilizadas. Nesta seção, são apresentados os resultados obtidos a partir da execução destas etapas anteriores.

O primeiro resultado a ser exibido é o valor da função de erro (*loss function*) utilizada para o treinamento da rede neural LSTM, que utilizou 70% do total de amostras selecionadas aleatoriamente.

Os valores desta função de erro para os 5 diferentes tipos de entrada (de 10 a 50 amostras) são exibidos na Tabela 1. Estes valores foram obtidos após a execução de 100 épocas de treinamento da rede.

Tabela 1 – Resultados do Erro Quadrático Médio no treinamento

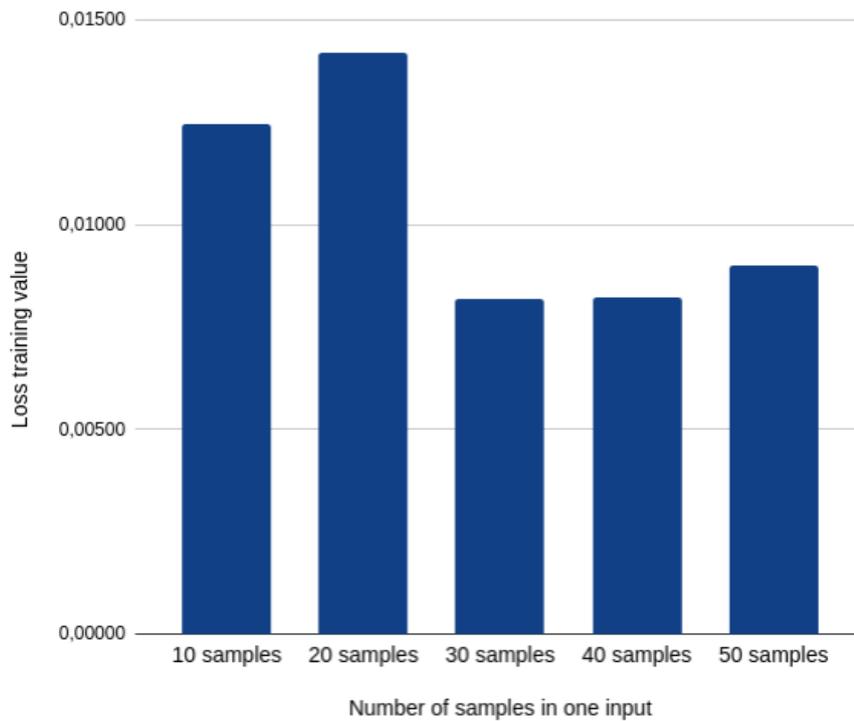
Tamanho de Entrada da Rede	Valor do EQM
10 amostras	0,01206
20 amostras	0,01421
30 amostras	0,00820
40 amostras	0,00824
50 amostras	0,00901

Fonte: (CASATI; ALTA FIM; ALTA FIM, 2020).

É possível observar nos dados apresentados na Tabela 1 que ao incrementar o tamanho da entrada da rede, o valor do EQM no treinamento tende a diminuir. Porém, após o tamanho de 30 amostras, este EQM volta a subir, demonstrando que para o treinamento, entradas com tamanho próximo a 30 amostras são suficientes para a determinação do

melhor treinamento para este modelo. A Figura 22 exibe a comparação visual destes valores.

Figura 22 – Gráfico que representa a variação dos valores de EQM para os treinamentos dos diferentes tamanhos de entrada.



Fonte: Casati, Altafim e Altafim (2020).

Ao analisar o gráfico da Figura 22 pode-se observar a queda do valor do EQM quando se altera a quantidade de amostras da entrada de 20 para 30. Após, um leve aumento no EQM, acompanhando o crescimento do tamanho das entradas da rede.

Tabela 2 – Resultados com entrada de 10 amostras

Medidas	Valores (%)
Verdadeiro Positivo	5,26
Verdadeiro Negativo	92,76
Falso Positivo	0,61
Falso Negativo	1,36
Acurácia	98,03

Fonte: (CASATI; ALTAFIM; ALTAFIM, 2020).

Após a etapa de treinamento, o teste é executado. Considerando 30% do total de amostras, os resultados do teste para as entradas contendo 10 amostras são exibidos na Tabela 2.

A Tabela 2 possui valores de verdadeiro positivo, verdadeiro negativo, falso positivo, falso negativo e acurácia para a execução utilizando entrada possuindo 10 amostras cada. Pode-se observar o baixo valor de falsos positivos, o que indica uma estabilidade do sistema para que não considere que haja veículo passando e gerando vibração na lombada, quando na verdade não há.

Tabela 3 – Resultados com entrada de 20 amostras

Medidas	Valores (%)
Verdadeiro Positivo	5,61
Verdadeiro Negativo	92,23
Falso Positivo	0,80
Falso Negativo	1,35
Acurácia	97,85

Fonte: (CASATI; ALTAFIM; ALTA-FIM, 2020).

Na Tabela 3 são apresentados os resultados considerando as entradas que possuem 20 amostras cada. Nela é possível observar que, se comparados aos resultados com entrada de 10 amostras, os valores são um pouco inferiores. A acurácia diminuiu e, ao analisar mais profundamente a tabela, pode-se concluir que foi influência direta do aumento de falsos positivos.

Tabela 4 – Resultados com entrada de 30 amostras

Medidas	Valores (%)
Verdadeiro Positivo	6,00
Verdadeiro Negativo	92,60
Falso Positivo	0,53
Falso Negativo	0,88
Acurácia	98,59

Fonte: (CASATI; ALTAFIM; ALTA-FIM, 2020).

Ao considerar as entradas com 30 amostras, que possui o menor valor de EQM na

etapa de treinamento, observa-se que os resultados obtidos na etapa de teste são melhores que os valores aferidos anteriormente (10 e 20 amostras para cada entrada).

A performance melhorou em todos os aspectos, com ênfase na importância da diminuição de falsos positivos e aumento na acurácia, alcançado 98,59%, como pode ser observado na Tabela 4.

Tabela 5 – Resultados com entrada de 40 amostras

Medidas	Valores (%)
Verdadeiro Positivo	6,05
Verdadeiro Negativo	92,71
Falso Positivo	0,35
Falso Negativo	0,90
Acurácia	98,76

Fonte: (CASATI; ALTA-FIM; ALTA-FIM, 2020).

Na Tabela 5 são apresentados os resultados para a execução com entradas de 40 amostras. Considerando que o EQM na etapa de treinamento teve um valor muito próximo ao alcançado na etapa utilizando 30 amostras por entrada, era esperado que os valores do resultado, principalmente a acurácia, fossem muito próximos.

Tabela 6 – Resultados com entrada de 50 amostras

Medidas	Valores (%)
Verdadeiro Positivo	6,03
Verdadeiro Negativo	92,57
Falso Positivo	0,57
Falso Negativo	0,82
Acurácia	98,61

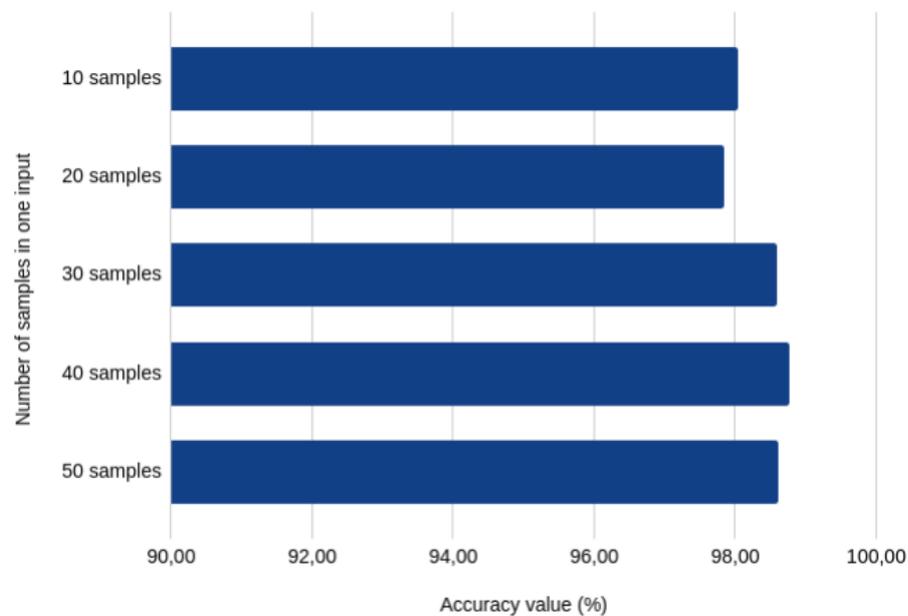
Fonte: (CASATI; ALTA-FIM; ALTA-FIM, 2020).

Apesar da proximidade dos valores, observa-se que houve melhoria nos resultados obtidos com esta etapa (40 amostras), contrariando o fato do EQM ter maior valor na etapa de treinamento. Isto demonstra um maior poder de generalização da rede neural trabalhando com 40 amostras por entrada.

Outro fator a ser notado é a diminuição considerável do valor de falsos positivos, apenas 0,35%, sendo este o melhor resultado obtido dentre todos os testes. Portanto, com este modelo é raro que a rede neural interprete os sinais considerando um veículo passando pela lombada, quando não houver.

Por fim, a Tabela 6 apresenta os resultados para o teste utilizando 50 amostras por entrada da rede. Os resultados obtidos são muito semelhantes aos demais.

Figura 23 – Gráfico que apresenta um comparativo do percentual de acurácia dos testes para os diferentes tamanhos de entrada.



Fonte: Casati, Altafim e Altafim (2020)

É possível concluir que a acurácia obtida pelo teste com 50 amostras por entrada pode ser alcançada com entradas de menor tamanho, que possuem menor custo computacional e tempo de resposta.

Uma comparação da performance geral considerando a acurácia dos 5 testes executados é apresentada na Figura 23 (Seção 5.1).

Ao observar a variação da acurácia pode-se considerar que o valor do EQM do treinamento da rede influencia em sua performance em um teste, mas não é perfeitamente proporcional.

Outra conclusão que pode ser obtida é que considerando os valores de resultados e o custo computacional (que não é o objetivo deste trabalho), a performance mais interessante seria a do teste utilizando 30 amostras por entrada.

Observando apenas a acurácia geral, o melhor resultado é obtido pela rede que utiliza entradas com 40 amostras.

Por se tratar de um experimento mais simples, para validação dos dados, os resultados obtidos se mostraram muito próximos e corroboram a teoria de que as vibrações causadas na lombada por impacto de veículos podem ser detectadas ao analisar os sinais de um acelerômetro.

Na próxima seção são apresentados resultados do segundo experimento, denominado aqui como Experimentação Principal, que considera maior volume de dados e maior complexidade nas análises.

5.2 Experimentação Principal

Nesta seção os resultados do segundo experimento são apresentados e discutidos. A avaliação se dá de diferentes formas, dependendo do objetivo de cada etapa.

Primeiramente, os resultados obtidos com a geração das amostras simulações são apresentados. Com o objetivo de aumentar e balancear o banco de dados real, são exibidos também exemplos visuais das amostras sintéticas.

Em um segundo momento, após os resultados das simulações, as etapas de classificação são avaliadas.

Estas etapas englobam a execução de testes utilizando MLP e CNN, ambas fazendo uso das amostras reais isoladas e também com acréscimo das amostras sintéticas para fins de comparação das métricas de resultado obtidas.

5.2.1 Geração de Simulações

A geração das simulações se deu utilizando redes neurais GAN. Detalhes sobre o processamento desta etapa foram exibidos na Seção 4.2.1.

A avaliação objetiva dos resultados obtidos com as amostras sintéticas geradas são melhor observados quando comparados os resultados obtidos com as etapas de classificação, pois é possível comparar o uso de amostras reais com o uso destes com o acréscimo dos sintéticos.

Esta seção se limita a exibir os resultados visuais obtidos das simulações performadas pelo algoritmo inteligente aplicado neste trabalho.

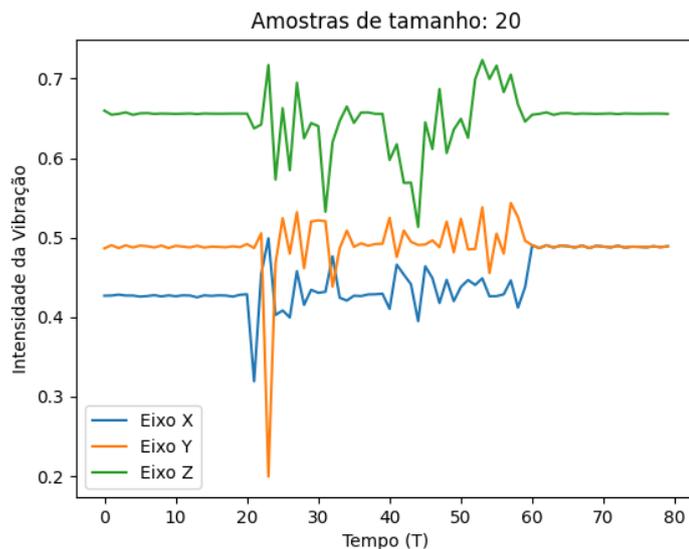
Para cada um dos tamanhos de amostra (1, 3, 5, 7, 10, 15 e 20), foram geradas simulações de impacto de veículos, considerando as três diferentes classes abordadas na etapa de aquisição de dados (3.3.2).

A Figura 24 exibe exemplos de amostras de tamanho 20. O gráfico pode ser dividido em 4 partes:

- No primeiro quarto (valores de 0 a 19 do eixo T), é exibida uma amostra real de

- classe 0 (sem impacto);
- No segundo quarto (valores de 20 a 39 do eixo T), é exibida uma amostra real de classe 1 (impacto de veículo);
 - No terceiro quarto (valores de 40 a 59 do eixo T), é exibida uma amostra sintética de classe 1 (impacto de veículo);
 - No último quarto (valores de 60 a 79 do eixo T), é exibida uma amostra real de classe 0 (sem impacto);

Figura 24 – Gráfico que apresenta a comparação entre duas amostras de tamanho 20, para a classe 1. No segundo quarto do Eixo T uma amostra real e no terceiro quarto uma amostra sintética.



Fonte: Autor.

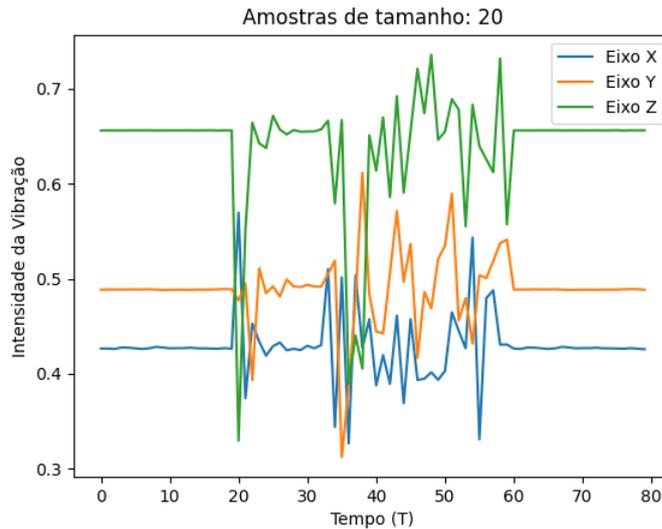
Esta configuração de dividir os gráficos em quatro quartos é utilizada ao longo desta seção, a fim de destacar as vibrações de impacto e também facilitar a comparação entre as amostras reais e sintéticas.

Um segundo exemplo comparativo entre as amostras real e sintética é exibido na Figura 25. É importante afirmar que os exemplos são selecionados de forma aleatória e sua comparação pode ser prejudicada em um universo de milhares de amostras.

Considerando a classe 2 de impacto (veículos de menor porte), a Figura 26 exibe a comparação entre duas destas amostras (sintética e real).

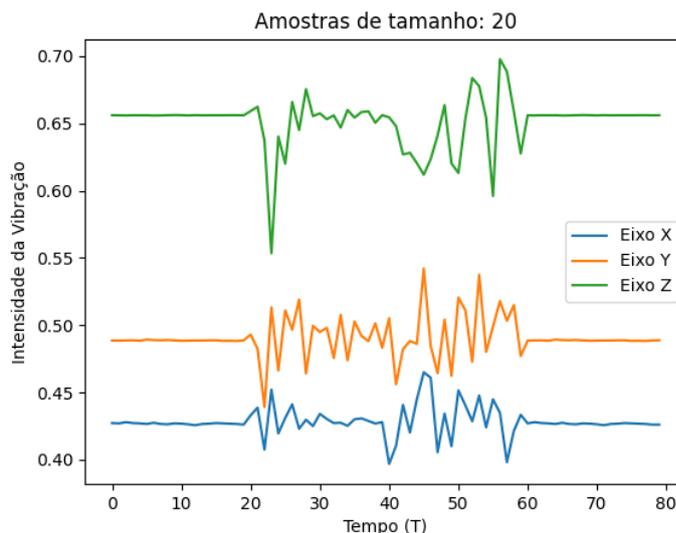
As amostras de classe 3 possuem os impactos dos veículos de maior porte. Um exemplo de amostras reais e sintéticas desta classe podem ser observadas na Figura 27.

Figura 25 – Gráfico que apresenta uma segunda comparação entre duas amostras de tamanho 20, para a classe 1. No segundo quarto do Eixo T uma amostra real e no terceiro quarto uma amostra sintética.



Fonte: Autor.

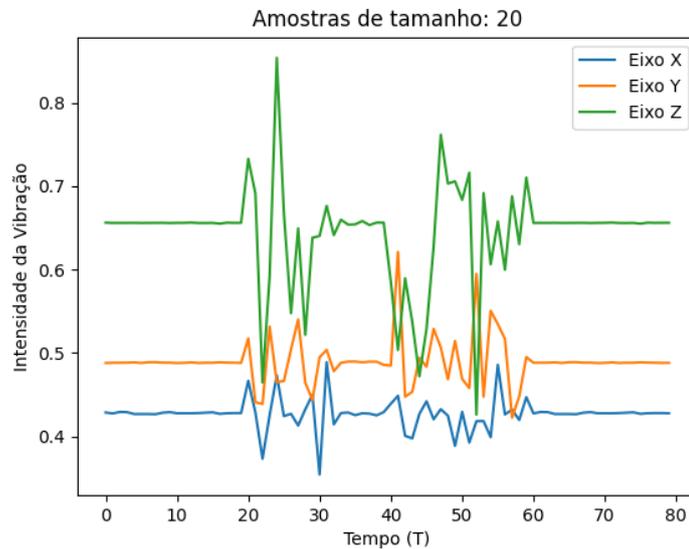
Figura 26 – Gráfico que apresenta uma comparação entre duas amostras de tamanho 20, para a classe 2. No segundo quarto do Eixo T uma amostra real e no terceiro quarto uma amostra sintética.



Fonte: Autor.

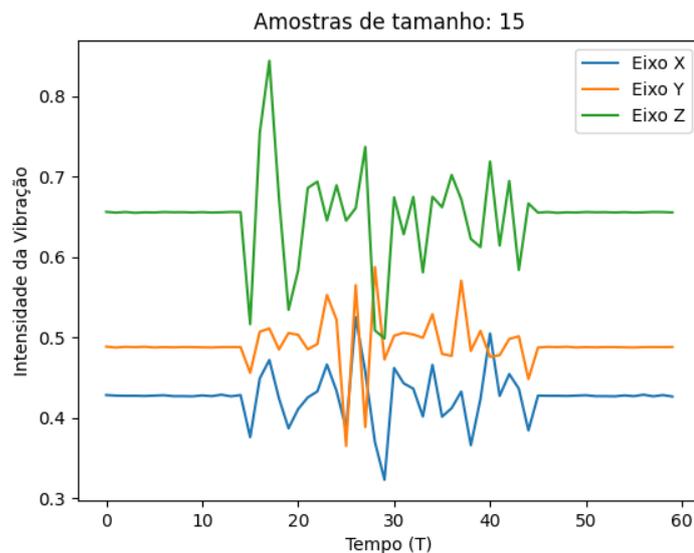
Observando-se os casos apresentados nas Figuras 24, 25, 26 e 27 quanto à variação da intensidade de vibração nos 3 eixos e o formato das amostras geradas a partir das simulações inteligentes, pode-se destacar a similaridade das mesmas com as amostras reais.

Figura 27 – Gráfico que apresenta uma comparação entre duas amostras de tamanho 20, para a classe 3. No segundo quarto do Eixo T uma amostra real e no terceiro quarto uma amostra sintética.



Fonte: Autor.

Figura 28 – Gráfico que apresenta uma comparação entre duas amostras de tamanho 15, para a classe 1. No segundo quarto do Eixo T uma amostra real e no terceiro quarto uma amostra sintética.

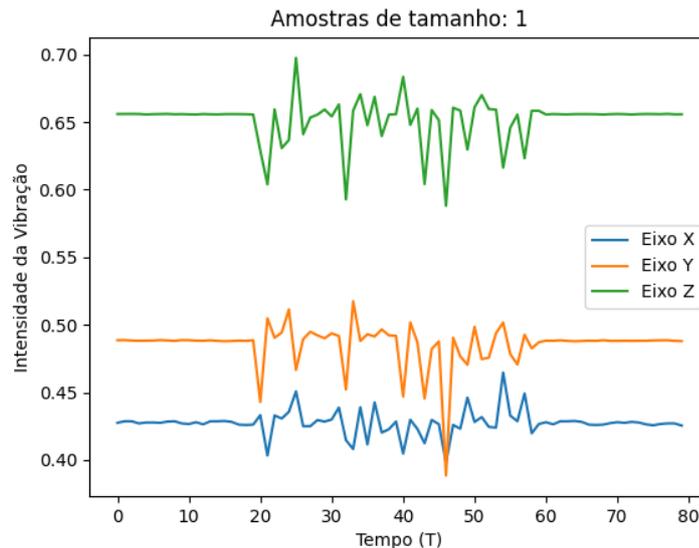


Fonte: Autor.

Em amostras de menor tamanho, observa-se também similaridade entre dados reais e simulações. A Figura 28 exibe um exemplo utilizando amostras de tamanho 15 para a

classe 1.

Figura 29 – Gráfico que apresenta uma segunda comparação entre amostras de tamanho 1, para a classe 2. O segundo quarto do Eixo T possui 20 amostras reais o terceiro quarto 20 amostras sintéticas.



Fonte: Autor.

Para que seja possível analisar de forma visual amostras muito pequenas, é necessário que estas sejam exibidas em maior quantidade. O gráfico apresentado na Figura 29 traz um exemplo de simulação de amostras com tamanho 1.

Também dividida em quatro partes e seguindo o mesmo padrão anterior, são alocadas 20 amostras de tamanho 1 para cada um dos quartos do gráfico.

Após observação do gráfico da Figura 29 pode-se afirmar que a rede neural GAN segue um padrão para geração das amostras sintéticas que é repetido independente do tamanho da amostra analisada.

Outra pertinente observação é que a simulação dos sinais de acelerômetro advindos de impacto dos veículos na lombada pode ser executado considerando amostras de tamanho fixo e que as mesmas sugerem, visualmente, seguir um mapeamento dos dados reais, transformando um vetor numérico aleatório em amostras que se assemelham às naturais.

As análises numéricas dos resultados das simulações são analisadas nas seções seguintes e avaliados de acordo com o desempenho das mesmas nas etapas de classificação dos padrões de vibrações.

5.2.2 Classificação MLP

Como um dos principais objetivos deste trabalho, a classificação das amostras de vibração se dá utilizando duas diferentes arquiteturas: MLP e CNN.

Os resultados obtidos com os treinamentos e testes executados nos dados são apresentados e discutidos para duas diferentes abordagens: utilizando apenas os dados reais e também os dados sintéticos.

Ao se utilizar a arquitetura MLP como classificador, tem-se como, por meio dos resultados obtidos, avaliar também o desempenho das simulações realizadas, que geraram amostras sintéticas.

Na Seção 5.2.2.1 são apresentados os resultados de classificação que, tanto no treinamento quanto nos testes, possuem apenas dados reais.

Posteriormente, na Seção 5.2.2.2 os resultados são apresentados e discutidos para a abordagem que se utiliza dos dados sintéticos gerados pelas simulações.

5.2.2.1 Dados Reais

A disparidade na quantidade de dados (*oversampling*) considerados negativos (amostras sem incidência de impacto de veículo) e os positivos (amostras com incidência de impacto de veículos) é mantida nesta etapa.

O objetivo de manutenção da quantidade de amostras é evitar interferência e engenharia de requisitos, passando esta responsabilidade para as arquiteturas de redes neurais.

Como descrito anteriormente (Capítulo 4), os testes são executados considerando três abordagens:

- Utilizando 2 classes;
- Utilizando 3 classes;
- Utilizando 4 classes.

Detalhes do funcionamento de cada abordagem foram descritos na Seção 4.2.2.1, onde foram apresentados também detalhes da arquitetura e topologia de rede neural utilizadas.

Os resultados das etapas de treinamento são aferidos utilizando a função de erro (*loss function*) que é aplicada. A minimização desta função ocorre durante as épocas de treinamento.

Os valores resultantes do treinamento podem ser observados na Tabela 7. Esta tabela possui os valores mínimos de erro alcançados para cada execução (tamanhos de amostra e quantidade de classes).

Tabela 7 – Resultados de treinamento da MLP utilizando apenas amostras reais.

Tam. da Amostra	Loss (2 classes)	Loss (3 classes)	Loss (4 classes)
1	0,2171	0,2867	0,3203
3	0,1862	0,2330	0,2558
5	0,2104	0,2635	0,2971
7	0,1627	0,1912	0,2112
10	0,1382	0,1743	0,1976
15	0,1161	0,1553	0,1830
20	0,1288	0,1897	0,2514

Ao analisar a Tabela 7 pode-se observar que há uma tendência na obtenção de melhores resultados quando se aumenta o tamanho da amostra.

Este aumento no desempenho encontra um limitante quando os resultados para as amostras de tamanho 20 são analisadas, pois estas produziram maior valor de erro se comparado com os tamanhos 10 e 15.

O gráfico contido na Figura 30 exibe a tendência do erro nos diferentes treinamentos das redes neurais e seus tamanhos de amostras.

Outras importantes observações analisando o gráfico apresentado na Figura 30:

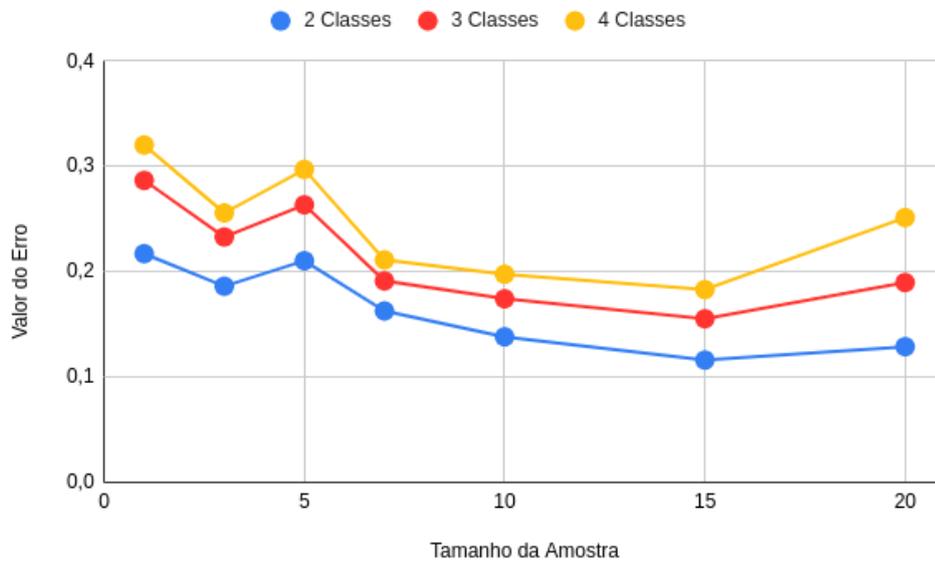
- As amostras de tamanho 15 obtiveram a melhor performance durante o processo de treinamento da rede;
- Quanto maior o número de classes, maior o valor do erro;
- Independente do número de classes de cada análise, a tendência sobre o valor do erro nos treinamentos é a mesma.

Apenas 30% das amostras são utilizadas na etapa de testes, que é executada após o treinamento da rede. Uma das métricas mais utilizadas para a avaliação dos resultados é a acurácia da classificação nas amostras de teste.

A Tabela 8 exibe os valores de acurácia para a etapa de classificação das amostras reais utilizando MLP.

Com os valores de acurácia da classificação obtidos, pode-se afirmar que há relação direta com os valores aferidos do erro do treinamento. A Figura 31 exibe um gráfico para demonstração visual da acurácia aferida nesta etapa de classificação.

Figura 30 – Gráfico que apresenta um comparativo entre os valores de erro alcançados pela rede MLP utilizando as amostras reais.



Fonte: Autor.

Tabela 8 – Resultados da acurácia da classificação do teste da MLP utilizando apenas amostras reais.

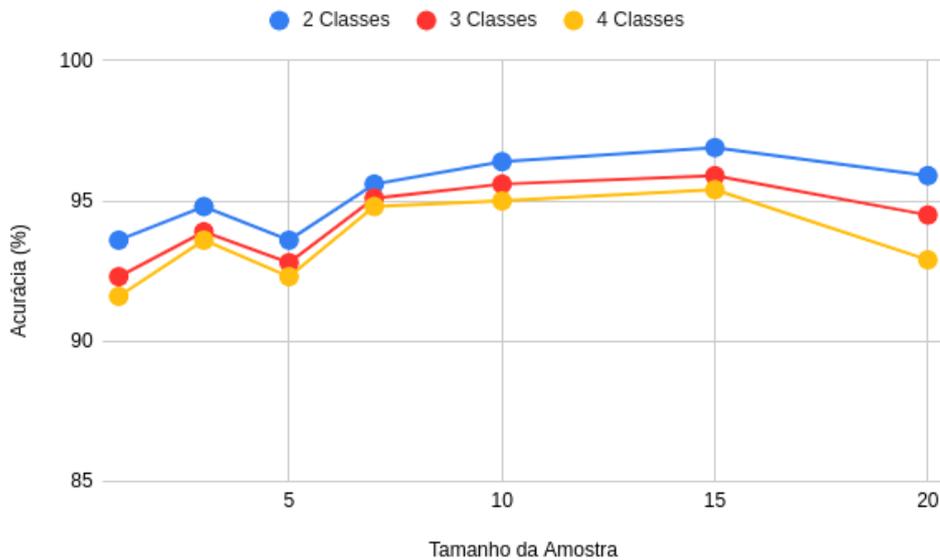
Tamanho da Amostra	Acurácia (2 classes)	Acurácia (3 classes)	Acurácia (4 classes)
1	93,6%	92,3%	91,6%
3	94,8%	93,9%	93,6%
5	93,6%	92,8%	92,3%
7	95,6%	95,1%	94,8%
10	96,4%	95,6%	95,0%
15	96,9%	95,9%	95,4%
20	95,9%	94,5%	92,9%

Uma tendência inversa a do erro é detectada ao analisar os gráficos apresentados, indicando que quanto maior o erro na etapa de classificação, menor a acurácia na etapa de testes.

Porém, apenas a acurácia não é suficiente para definir um bom resultado. A fim de investigar a performance da classificação utilizando as amostras reais, foram geradas matrizes de confusão de todos os resultados.

Ao analisar as matrizes, observou-se que a grande quantidade de amostras negativas (classe 0) tendenciou o algoritmo à classificação errônea de muitas amostras de outras

Figura 31 – Gráfico que apresenta um comparativo entre os valores de acurácia alcançados pela rede MLP utilizando as amostras reais.



Fonte: Autor.

classes, chegando a obter resultado nulo em determinados casos.

Para exemplificar a questão discutida, a Figura 32 apresenta a matriz de confusão do treinamento e teste utilizando 4 classes e amostras de tamanho 15.

Por atingir um valor muito alto na acurácia da classe 0 (99,56% de acertos) e por esta classe possuir grande quantidade de amostras, mesmo que a acurácia tenha sido baixa nas demais classes, o valor geral tende a ser alto.

A Figura 32 exibe os baixos valores de acurácia referentes às demais classes, inclusive salientando que não há incidência de classificação para as classes 2 e 3. No caso da classe 1, a acurácia é de 68,34%, valor significativamente abaixo da média geral.

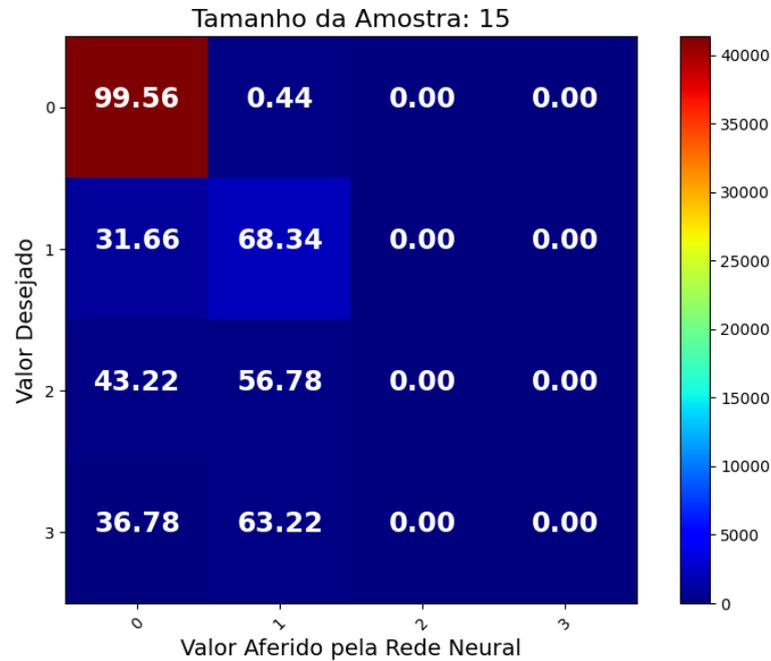
Este comportamento foi detectado em todas as aplicações utilizando apenas as amostras reais, o que permite concluir que trata-se de um exemplo da ocorrência de *oversampling*.

Com o objetivo de mitigar este problema e validar as simulações executadas, a próxima seção apresenta uma etapa de classificação semelhante a essa, porém, com a adição das amostras sintéticas geradas pela rede GAN.

5.2.2.2 Dados Simulados

Com o objetivo de superar os problemas encontrados nos resultados apresentados na Seção 5.2.2.1, neste experimento, a quantidade de amostras reais utilizada é a mesma

Figura 32 – Matriz de confusão do resultado do teste de classificação com amostras reais utilizando MLP, amostras de tamanho 15 e 4 classes (valores em %). Legenda: quantidade de amostras.



Fonte: Autor.

da anterior, porém, são adicionadas amostras sintéticas, como as exibidas na Seção 5.2.1.

A etapa de treinamento, guiada pelo valor da função de erro, apresentou diminuição em seus valores, exibidos na Tabela 9.

Tabela 9 – Resultados de treinamento da MLP e adição de amostras sintéticas.

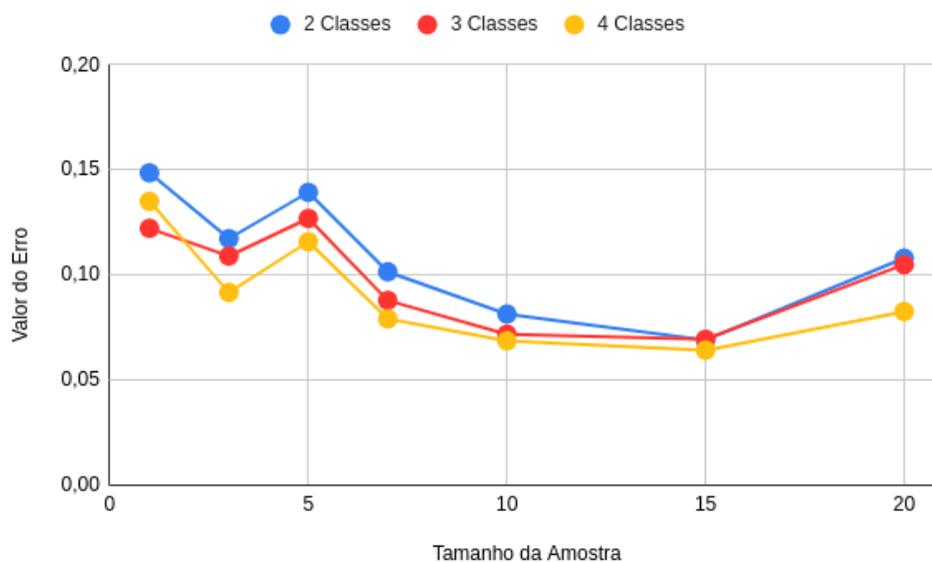
Tam. da Amostra	Loss (2 classes)	Loss (3 classes)	Loss (4 classes)
1	0.1486	0.1222	0.1350
3	0.1173	0.1090	0.0917
5	0.1393	0.1269	0.1158
7	0.1015	0.0879	0.0791
10	0.0814	0.0718	0.0687
15	0.0690	0.0694	0.0642
20	0.1080	0.1049	0.0826

É significativa a redução no erro na etapa de treinamento. o menor valor apresentado na Seção 5.2.2.1 é de 0,1161 utilizando as amostras de tamanho 15 e considerando 2 classes. Nesse experimento, o melhor valor alcançado é 0,0690, com a mesma configuração.

Apenas considerando os melhores resultados obtidos para esta métrica, a melhora é de aproximadamente 40%, demonstrando que há maior convergência dos parâmetros ajustados da rede neural em relação às amostras apresentadas.

Com o objetivo de melhorar o entendimento dos treinamentos executados, a Figura 33 exibe o comportamento dos valores de erro variando-se o tamanho da amostra.

Figura 33 – Gráfico que apresenta um comparativo entre os valores de erro alcançados pela rede MLP utilizando amostras sintéticas.



Fonte: Autor.

Muito semelhante à tendência apresentada na Figura 30, onde são utilizadas apenas amostras reais, neste caso os valores de erro são significativamente menores.

Manteve-se o tamanho de amostra 15 como a configuração que apresenta o menor erro no treinamento da rede neural, mas com pouca diferença para as amostras de tamanho 10.

Uma questão relevante é a diferença do valor de erro quando há variação no número de classes consideradas no treinamento. Anteriormente, quanto maior o número de classes, maior o valor de erro. No caso desse experimento (utilizando amostras sintéticas), excluindo-se o treinamento das amostras de tamanho 3, há uma inversão do resultado.

Observa-se na Figura 30 com clareza que a performance do treinamento é melhor quando se aumenta o número de classes, o que indica que as amostras sintéticas podem ser mais distintas. Há também um indício que o problema de *oversampling* pode estar sendo diminuído.

Para melhor análise comparativa, a Tabela 10 traz os valores de acurácia alcançados

na etapa de teste da rede neural, que utiliza 30% do total de amostras, selecionados aleatoriamente.

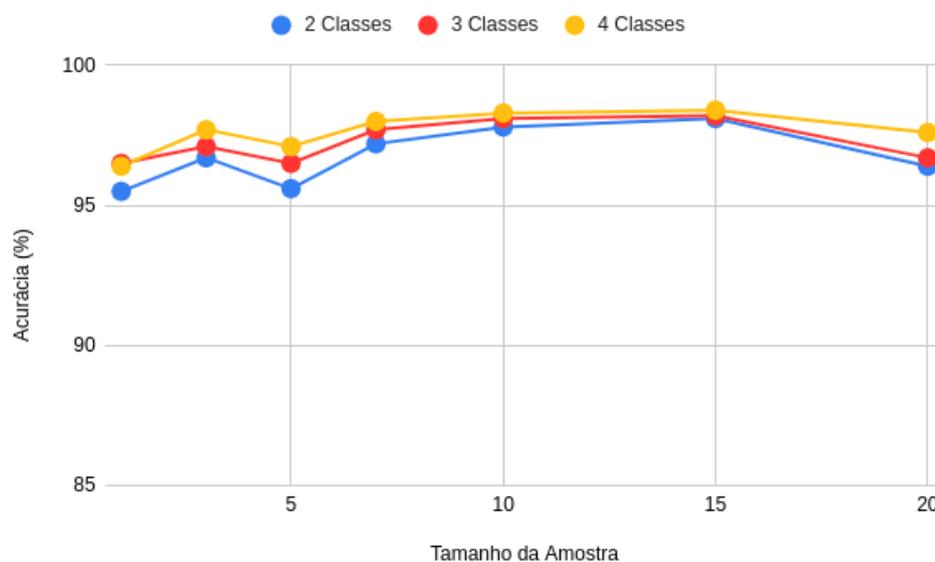
Tabela 10 – Resultados de acurácia da classificação do teste da MLP utilizando amostras sintéticas adicionadas às reais.

Tamanho da Amostra	Acurácia (2 classes)	Acurácia (3 classes)	Acurácia (4 classes)
1	95,5%	96,5%	96,4%
3	96,7%	97,1%	97,7%
5	95,6%	96,5%	97,1%
7	97,2%	97,7%	98,0%
10	97,8%	98,1%	98,3%
15	98,1%	98,2%	98,4%
20	96,4%	96,7%	97,6%

Ao analisar os valores obtidos e apresentados na Tabela 10, pode-se observar um ganho nos valores em comparação à abordagem que não utilizada amostras sintéticas.

Por não haver tanta disparidade na quantidade de amostras, visto que são adicionadas 100.000 amostras para cada classe a ser analisada, o aumento nos valores de acurácia geral podem ser interpretados como uma melhora na performance.

Figura 34 – Gráfico que apresenta um comparativo entre os valores de acurácia alcançados pela rede MLP utilizando as amostras sintéticas.

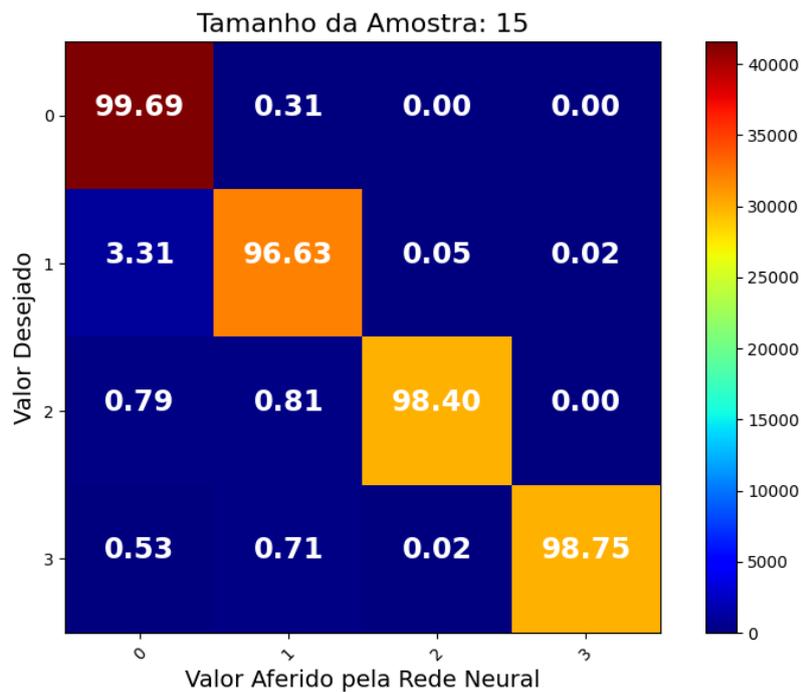


Fonte: Autor.

O gráfico apresentado na Figura 34 exibe o comportamento da acurácia em relação à quantidade de classes considerada e aos diferentes tamanhos de amostra.

Com isto, é possível avaliar que, em relação à utilização apenas de amostras reais, houve melhora na performance da classificação.

Figura 35 – Matriz de confusão do resultado do teste de classificação com amostras sintéticas utilizando MLP, amostras de tamanho 15 e 4 classes (valores em %).
Legenda: quantidade de amostras.



Fonte: Autor.

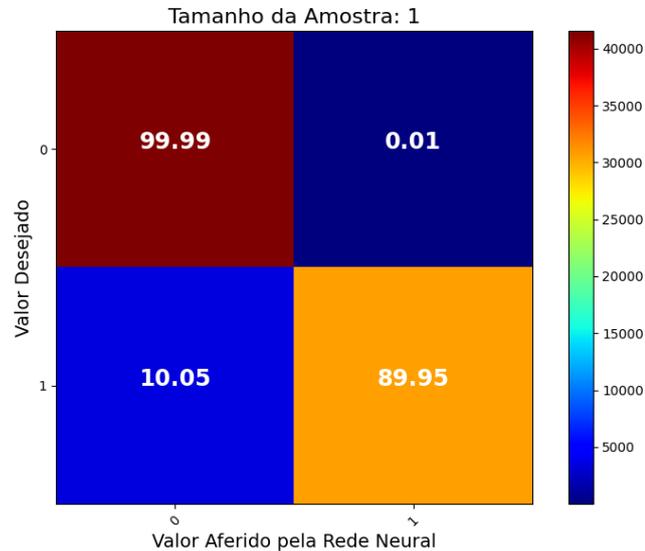
Se comparado o maior valor alcançado na Seção 5.2.2.1 com o maior valor apresentado na Tabela 10, observa-se um ganho aproximado de apenas 1,5%. Porém, é necessário que seja feita a análise considerando a quantidade de acertos para cada classe, a fim de se discutir sobre a redução do efeito de *oversampling*.

A Figura 35 apresenta a matriz de confusão normalizada (em percentual de acertos) gerada a partir do teste que gerou os melhores resultados: tamanho de amostra 15 e 4 classes.

O importante a se analisar na Figura 35 são os valores de percentual de acerto, pois a escala de cores apresenta a quantidade absoluta de amostras em cada posição da matriz e, apesar do aumento na quantidade de amostras positivas, ainda assim continuam em menor volume que as amostras negativas.

Os resultados observados na matriz de confusão da Figura 35 apresentam significativa melhora se comparados com os apresentados na Figura 32, pois a acurácia é superior a 96%

Figura 36 – Matriz de confusão do resultado do teste de classificação com amostras sintéticas utilizando MLP, amostras de tamanho 1 e 2 classes (valores em %).
Legenda: quantidade de amostras.



Fonte: Autor.

em todas as classes analisadas pela MLP, apresentando maior confusão entre as classes 0 e 1 (menos de 4%).

É possível avaliar, em relação ao *oversampling*, que os resultados obtidos com a adição das amostras sintéticas promoveram melhora na classificação do total de amostras, para todas as classes analisadas.

Com o objetivo de avaliar também o que seria um resultado em uma configuração de menor desempenho, a Figura 36 apresenta a matriz de confusão gerada a partir dos resultados obtidos com amostras de tamanho 1 e considerando apenas 2 classes.

Ao observar os resultados apresentados na Figura 36, foi necessário descobrir se os 10% de amostras confundidas pela rede neural se tratavam apenas de amostras reais.

A investigação conclui que há amostras reais e sintéticas participantes do montante que se encontram em confusão e que a quantidade absoluta de amostras reais é menor que o mesmo experimento aplicado sem o uso de amostras sintéticas.

Deste modo, as amostras sintéticas foram capazes de induzir a rede neural a classificar corretamente um maior número de amostras reais, permitindo concluir que o problema de *oversampling* foi mitigado com a aplicação do método proposto neste trabalho.

Os demais resultados (tamanhos de amostra e número de classes consideradas) seguem o mesmo padrão dos apresentados aqui em detalhes, com altas taxas de acurácia para todas as classes avaliadas.

Na seção seguinte são apresentados e discutidos os resultados de classificação das amostras utilizando redes neurais CNN.

5.2.3 Classificação Convolutacional

O objetivo desta etapa do experimento é verificar se as redes CNN são capazes de alcançar acurácia relevante quanto ao uso de amostras reais e, em caso de *oversampling*, se é capaz de minimizar o problema utilizando-se das amostras sintéticas.

Nesta etapa de classificação são utilizadas apenas amostras de maior dimensão, visto que é necessário espaço para que a convolução seja executada nas amostras. Portanto, as que possuem menos de 10 sinais são descartadas para estes testes.

Mantendo-se a abordagem realizada com a rede MLP, os procedimentos são executados com os dados reais e posteriormente com os dados acrescidos de amostras sintéticas.

Com o objetivo de isolar a arquitetura da rede para comparação com a MLP, os 70% de amostras utilizadas no treinamento da CNN são os mesmo que, na etapa de classificação MLP, foram selecionadas aleatoriamente. Consequentemente, os 30% de amostras selecionadas para teste também são as mesmas.

5.2.3.1 Dados Reais

Nesta primeira etapa de uso da CNN, apenas os dados reais são considerados. Como apresentado anteriormente, há um problema de *oversampling* já detectado. O objetivo desta seção é avaliar se há melhorias nos resultados se comparados aos apresentados na Seção 5.2.2.1.

Tabela 11 – Resultados de treinamento da CNN utilizando apenas amostras reais.

Tam. da Amostra	Loss (2 classes)	Loss (3 classes)	Loss (4 classes)
10	0,1464	0,1814	0,2016
15	0,1263	0,1595	0,1858
20	0,1292	0,2436	0,2345

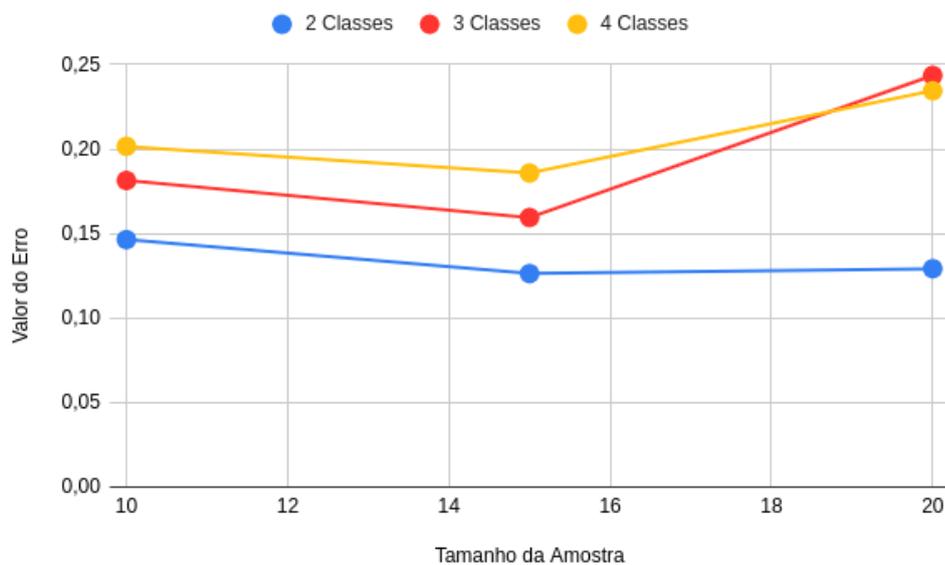
Os resultados referentes ao treinamento da rede neural CNN utilizando apenas os dados reais são apresentados na Tabela 11, apresentando o resultado da função de erro (*loss function*) considerando as variações de tamanho de amostra e quantidade de classes.

Apesar dos resultados do treinamento da rede neural CNN sofrerem variação em relação aos resultados da MLP, pode-se afirmar que esta não apresentou diferenças significativas nos valores de erro. Desse modo, pode-se inferir que há grande possibilidade do problema de *oversampling* identificado anteriormente ter se mantido na mesma proporção.

O gráfico exibido na Figura 37 demonstra grande semelhança com os resultados obtidos e apresentados na Seção 5.2.2.1.

Apesar do erro apresentado nas amostras de tamanho 20 para 3 classes ter sido um pouco elevado, a tendência apresentada anteriormente na utilização da MLP não sofre grandes alterações em relação ao uso da CNN nos dados reais.

Figura 37 – Gráfico que apresenta um comparativo entre os valores de erro alcançados pela rede CNN utilizando as amostras reais.



Fonte: Autor.

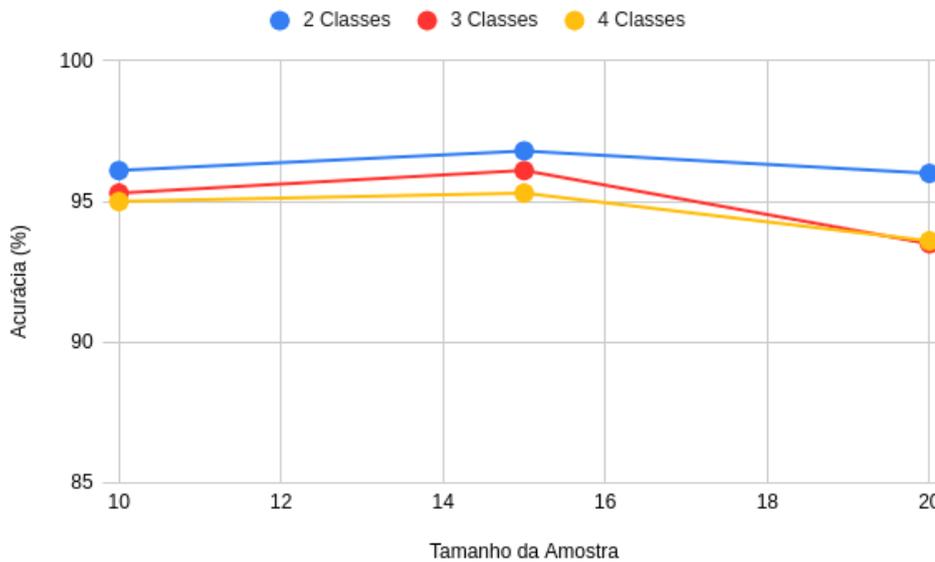
Para melhor analisar os resultados que utilizam apenas amostras reais, a Tabela 12 exhibe a acurácia total da classificação no conjunto de testes.

Tabela 12 – Resultados da acurácia da classificação do teste da CNN utilizando apenas amostras reais.

Tamanho da Amostra	Acurácia (2 classes)	Acurácia (3 classes)	Acurácia (4 classes)
10	96,1%	95,3%	95,0%
15	96,8%	96,1%	95,3%
20	96,0%	93,5%	93,6%

Também quanto à acurácia, pouco se observa de alterações que poderiam significar algum progresso relevante na classificação das amostras. Com o objetivo de propiciar uma análise visual do comportamento da acurácia, a Figura 38 apresenta o comparativo dentre os diferentes tamanhos de amostra e a quantidade de classes considerada nos testes.

Figura 38 – Gráfico que apresenta um comparativo entre os valores de acurácia alcançados pela rede CNN utilizando as amostras reais.



Fonte: Autor.

O gráfico exibido sugere a manutenção da tendência que é apresentada nos valores de erro da etapa de treinamento: quanto maior o erro, menor a acurácia e vice-versa.

Uma diferença a ser notada em relação à MLP é que nas amostras de tamanho 20, considerando-se 4 classes, a tendência de performance é superior ao teste de amostras de tamanho 15, diferente do que ocorre na abordagem apresentada na Seção 5.2.2.1.

A Figura 39 traz a matriz de confusão referente a este resultado, o que permite uma melhor observação do ocorrido nos testes.

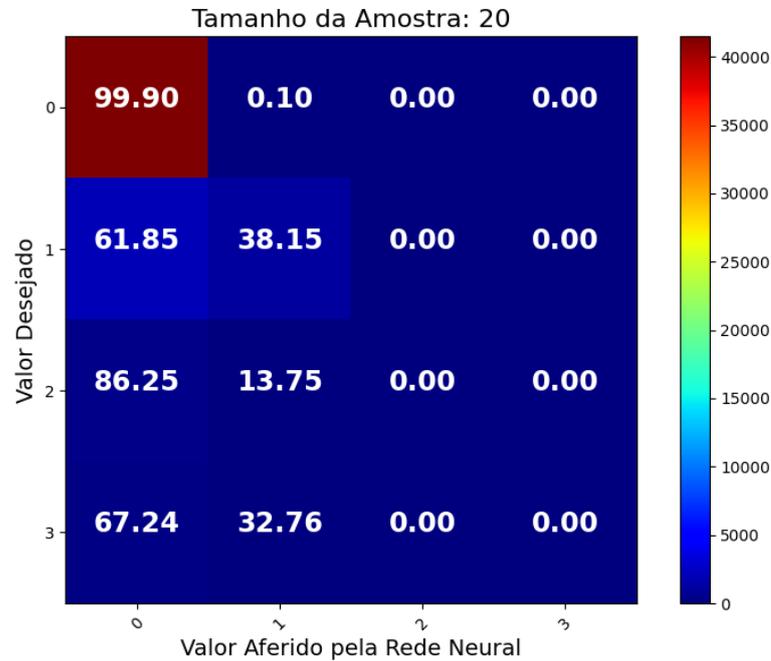
Observa-se então que os resultados sugerem que o problema de *oversampling* se manteve em proporções semelhantes aos encontrados utilizando-se MLP. Pode-se concluir que o ganho de performance quanto à acurácia alcançada se deve apenas à melhora na distinção entre as classes 0 e 1.

Deste modo, é necessário que se execute os testes utilizando também as amostras sintéticas e que os resultados sejam comparados a fim de verificar se a CNN também é capaz de minimizar significativamente o problema de *oversampling* e proporcionar altos valores de acurácia para as demais classes.

5.2.3.2 Dados Simulados

Aqui são apresentados os resultados da utilização da CNN com a adição dos dados sintéticos aos reais, já analisados e discutidos na seção anterior.

Figura 39 – Matriz de confusão do resultado do teste de classificação com amostras reais utilizando CNN, amostras de tamanho 20 e 4 classes (valores em %). Legenda: quantidade de amostras.



Fonte: Autor.

Na etapa de treinamento, a avaliação da *loss function* é apresentada para cada uma das execuções. Seus resultados são apresentados na Tabela 13.

Tabela 13 – Resultados de treinamento da CNN e adição de amostras sintéticas.

Tam. da Amostra	<i>Loss</i> (2 classes)	<i>Loss</i> (3 classes)	<i>Loss</i> (4 classes)
10	0,0837	0,0780	0.0677
15	0,0692	0,0686	0.0640
20	0,1811	0,0910	0.0814

Os valores dos resultados de treinamento se assemelham com os apresentados pela rede MLP, o que sugere uma melhoria com a adição das amostras sintéticas. Para melhor analisar o comportamento destes resultados, a Figura 40 apresenta um gráfico comparativo.

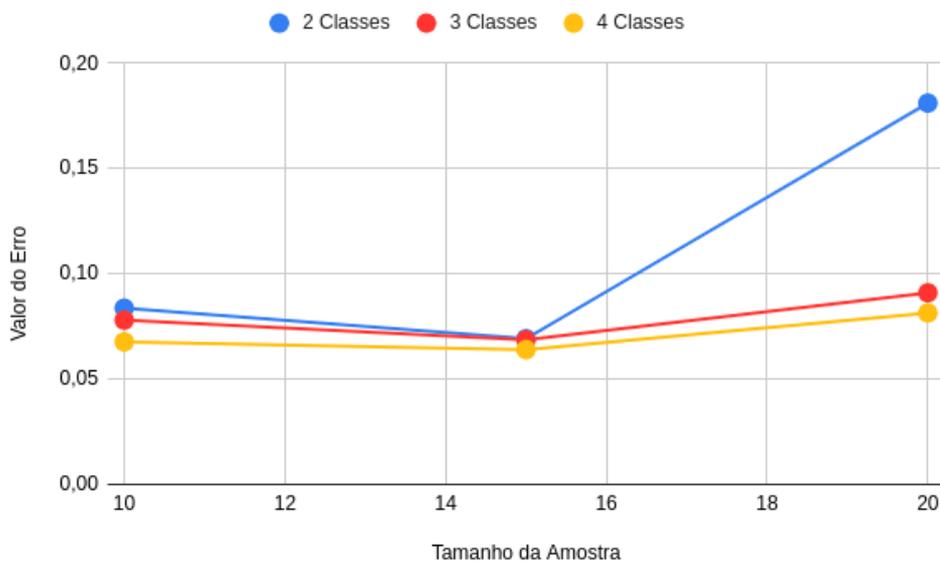
Destaca-se neste gráfico (Figura 40) que o valor de erro no treinamento da rede para a amostra de tamanho 20 e considerando 2 classes é nitidamente maior que o restante e não segue a tendência alcançada nos testes executados com a rede MLP (Figura 33).

O restante dos resultados estão dentro de uma normalidade já alcançada pela rede MLP, o que sugere avanços na questão de minimização de *oversampling*.

Para melhor avaliar estas tendências, a acurácia alcançada com cada teste é apresentada na Tabela 14.

Observa-se então, nos valores apresentados na Tabela 14, que a tendência apresentada nos resultados de treinamento (Tabela 13) se mantém, configurando queda na acurácia geral dos resultados obtidos utilizando amostras de tamanho 20 e 2 classes.

Figura 40 – Gráfico que apresenta um comparativo entre os valores de erro alcançados pela rede CNN utilizando amostras sintéticas.



Fonte: Autor.

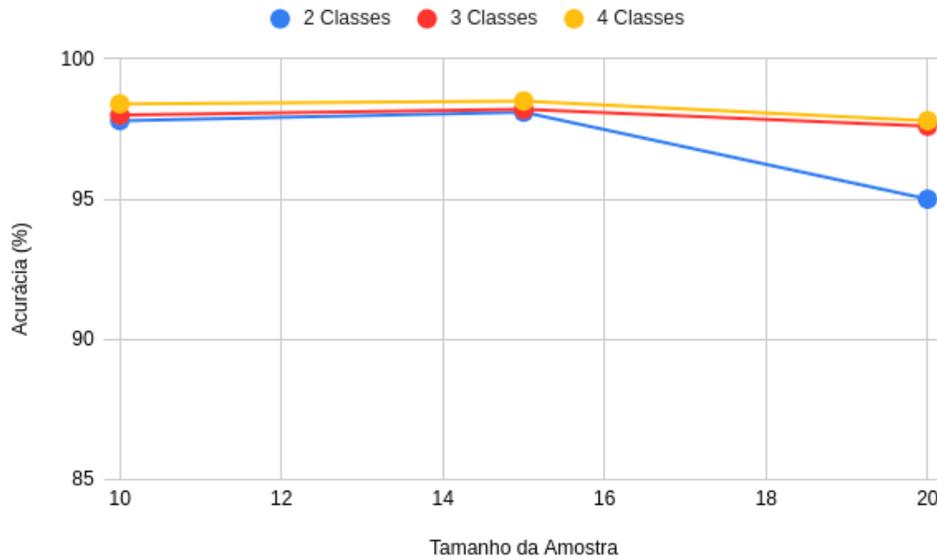
Tabela 14 – Resultados de acurácia da classificação do teste da CNN utilizando amostras sintéticas adicionadas às reais.

Tamanho da Amostra	Acurácia (2 classes)	Acurácia (3 classes)	Acurácia (4 classes)
10	97,8%	98,0%	98,4%
15	98,1%	98,2%	98,5%
20	95,0%	96,7%	97,8%

A Figura 38 exibe um gráfico comparativo com a acurácia geral alcançada utilizando redes CNN e adição de amostras sintéticas, com o objetivo de visualizar a tendência dos resultados alcançados.

As linhas exibidas no gráfico apresentado na Figura 41 confirmam a manutenção da tendência entre valores de erro de treinamento e acurácia dos testes, pois a queda de performance das amostras de tamanho 20 e 2 classes se manteve.

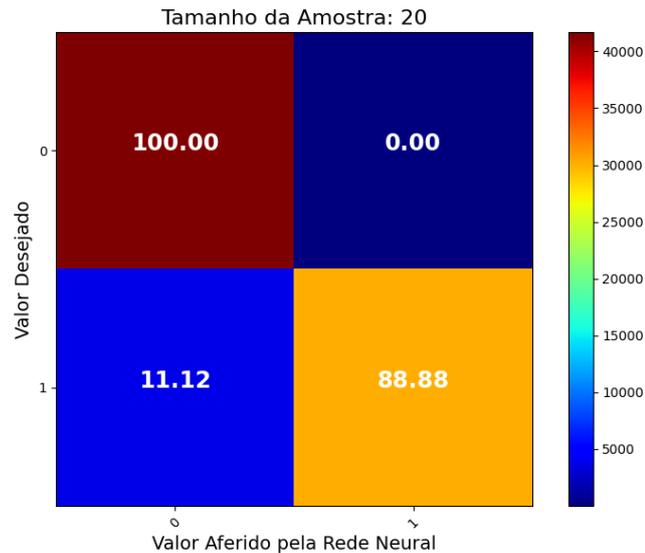
Figura 41 – Gráfico que apresenta um comparativo entre os valores de acurácia alcançados pela rede CNN utilizando as amostras sintéticas.



Fonte: Autor.

A Figura 42 apresenta a matriz de confusão para o teste executado utilizando as amostras de tamanho 20 e 2 classes, com o objetivo de analisar a queda de performance.

Figura 42 – Matriz de confusão do resultado do teste de classificação com amostras sintéticas utilizando CNN, amostras de tamanho 20 e 2 classes (valores em %).
Legenda: quantidade de amostras.



Fonte: Autor.

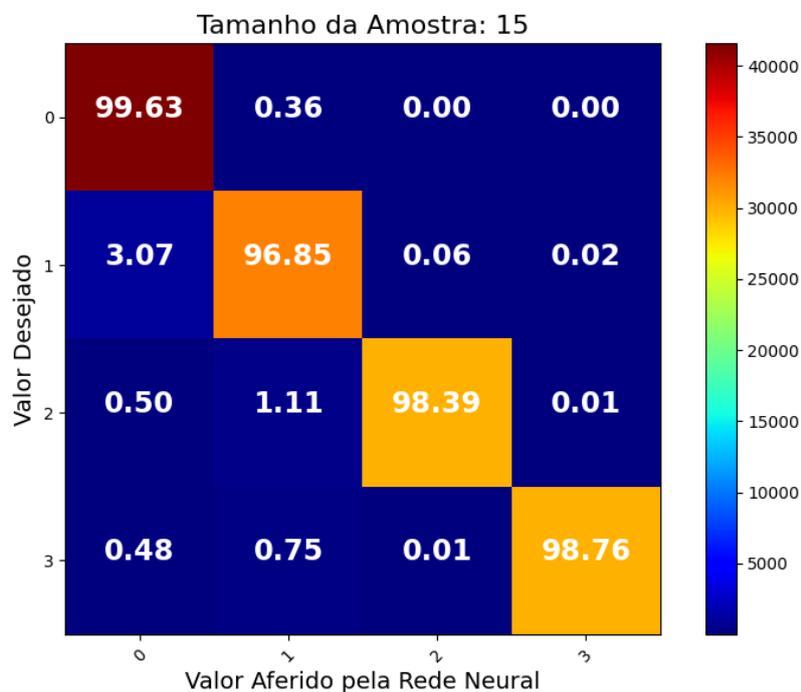
Ao analisar a matriz apresentada na Figura 42 é possível concluir que há uma maior quantidade de amostras de classe 1 que ainda são confundidas com amostras de classe 0, sugerindo que o problema de *oversampling* foi mitigado, porém, não o suficiente para que se produzisse um resultado mais otimizado.

Com o aprofundamento da investigação deste resultado, observou-se que a proporção de amostras sendo confundidas é praticamente a mesma entre reais e sintéticas, o que corrobora a tese de que as amostras sintéticas são suficientemente semelhantes às reais, a ponto de poderem ser utilizadas em treinamentos e testes de classificadores.

É possível observar também, nos resultados apresentados nessa seção, que houve sensível melhora na acurácia (0,1%) alcançada no melhor dos resultados (amostras de tamanho 15 e 4 classes), se comparado aos resultados apresentados pela rede MLP.

A matriz de confusão apresentada na Figura 43 exibe uma análise mais aprofundada dos valores obtidos no resultado de melhor performance, considerando a acurácia geral.

Figura 43 – Matriz de confusão do resultado do teste de classificação com amostras sintéticas utilizando CNN, amostras de tamanho 15 e 4 classes (valores em %).
Legenda: quantidade de amostras.



Fonte: Autor.

A diminuição da confusão entre as amostras e o conseqüente aumento na acurácia da classificação observado na Figura 43 proporcionam eficácia na utilização desta configuração para a classificação proposta pelo experimento.

A análise investigativa das amostras que apresentaram confusão considerando a classificação apresentada na Figura 43 concluiu que ambos os tipos de amostras (reais e sintéticas) fazem parte do grupo onde ocorre confusão e amostras reais que anteriormente eram confundidas (sem o uso de amostras sintéticas), passam a ser corretamente classificadas.

Para os demais testes que não são apresentados de forma detalhada (matriz de confusão), a proporção das amostras confundidas e a acurácia alcançada em cada classe se mantém semelhante à apresentada em detalhes (tamanho de amostra 15 e 4 classes), seguindo os valores de acurácia geral.

Apresentou-se nessa seção os resultados obtidos utilizando a rede CNN. Esta se demonstrou capaz de mitigar o problema de *oversampling*, assim como a MLP. O melhor resultado obtido pela MLP sofreu sensível melhora com o uso da CNN (vide Figuras 41 e 43 e a Tabela 14).

Concluindo assim que, com a adição das amostras sintéticas, a performance da classificação das vibrações causadas pelos impactos de veículos em lombadas é significativamente aumentada.

No próximo capítulo é apresentada a conclusão do trabalho, pontuando as principais contribuições e guiando possíveis trabalhos futuros e novos experimentos que podem ser executados a fim de aprofundar ainda mais a análise das vibrações.

6 CONCLUSÃO

Neste trabalho é apresentada uma análise das vibrações causadas por impacto de veículos em uma lombada, com o objetivo de detectar e classificar estes dados.

Em um primeiro momento, um sistema de aquisição dos dados de vibração é construído. Fazem parte desse sistema uma lombada, que foi confeccionada de forma artesanal em madeira e um aplicativo móvel desenvolvido para o sistema operacional Android.

Dois experimentos foram executados no decorrer do trabalho. O primeiro, mais simples, analisou a vibração de uma forma recorrente e alcançou acurácia relevante.

A partir deste, houve publicação em congresso: (CASATI; ALTA FIM; ALTA FIM, 2020). Apesar dos altos valores de acurácia alcançados (mais de 98%), não se analisou profundamente neste primeiro experimento, questões como amostragem, classificação e possíveis problemas de *oversampling*.

Os experimentos de aquisição foram executados, porém, a quantidade de dados obtidos se mostrou insuficiente para que, com o uso de inteligência artificial, fosse possível efetuar a classificação dos sinais adquiridos.

Neste sentido, redes neurais GAN foram criadas para que se pudesse gerar simulações (*data augmentation*) dos dados reais, fazendo assim com que o banco de dados fosse equalizado e os testes com a classificação dos padrões de vibração pudessem performar corretamente.

Os resultados obtidos com o segundo experimento permitem concluir que a classificação dos padrões de vibração puderam se beneficiar do processo de simulação, fazendo assim com que altos valores de acurácia (acima de 90%) fossem alcançados para todas as classes de veículos consideradas.

Embora os resultados das simulações utilizando IA tenham sido promissores, não se descarta como trabalho futuro a implantação de novos dados reais com o objetivo de prover maior segurança na geração das simulações.

Embora a técnica de *Data Augmentation* tenha sido utilizada por outros trabalhos, este traz como contribuição sua utilização em sinais de acelerômetros gerados a partir de impacto de veículos em lombada.

Com esta aplicação, abre-se a possibilidade de estudos futuros, como a aproximação de funções que relacionam o impacto ao peso e à danos em pavimentação.

Também como uma das propostas para trabalhos futuros, propõe-se a utilização

de *Autoencoders* (definido na Seção 2.3.2), que podem proporcionar uma nova leitura e otimização nas amostras reais e sintéticas da aplicação.

Outras contribuições deste trabalho são:

- O desenvolvimento de um sistema de aquisição de dados (vibração gerada a partir de impacto de veículos em lombada);
- O desenvolvimento de aplicativo móvel apropriado para a coleta e armazenamento de dados de vibração e movimento, que pode ser utilizado para diversos tipos de experimentos;
- O desenvolvimento de sistema inteligente para simulação e geração de dados de vibração e movimento (acelerômetro);
- Um estudo aprofundado sobre a classificação de amostras de vibração, comparando-se o uso de amostras reais e amostras sintéticas;
- Um sistema que se mostrou eficiente para a redução significativa de *oversampling* das amostras reais adquiridas.

Os resultados apresentados neste trabalho permitem avanços em possíveis aplicações considerando tráfego de veículos, monitoramento e soluções em cidades inteligentes.

Conclui-se com este trabalho que a utilização de amostras sintéticas geradas de forma inteligente podem simular situações reais de impacto de veículos em lombada, proporcionando também melhor desempenho nas classificações mas, principalmente, apresentando considerável diminuição no problema de *oversampling* comumente encontrado em bancos de dados desbalanceados.

REFERÊNCIAS

- ALVES, J. F. *et al.* An accelerometer based on thermoformed piezoelectrets with open-tubular channels. *In: Conference on Electrical Insulation and Dielectric Phenomena (CEIDP 2020)*. East Rutherford, NJ, USA: [S.l.: s.n.], 2020.
- BENGIO, Y. Learning deep architectures for AI. **Foundations and Trends® in Machine Learning**, Now Publishers, v. 2, n. 1, p. 1–127, 2009. Disponível em: <https://doi.org/10.1561/22000000006>.
- BHATTIPROLU, S. **Generative Adversarial Networks (GAN) using keras in python**. Youtube, 2020. [Online; accessed 22 Nov 2020]. Disponível em: <https://www.youtube.com/watch?v=Mng57Tj18pc>.
- BROWNLEE, J. **How to Develop RNN Models for Human Activity Recognition Time Series Classification**. Machine Learning Mastery, 2018. [Online; accessed 22 April 2020]. Disponível em: <https://machinelearningmastery.com/how-to-develop-rnn-models-for-human-activity-recognition-time-series-classification/>.
- BUDUMA, N. **Fundamentals of Deep Learning: Designing next-generation machine intelligence algorithms**. 1. ed. [S.l.: s.n.]: O’Reilly Media, 2017.
- CAFFE. **Caffe**. 2020. <http://caffe.berkeleyvision.org/>. [Online; accessed 21 May 2020].
- CALIN, O. **Deep Learning Architectures: A Mathematical Approach**. Ypsilanti, MI, USA: Springer, 2020. ISBN 978-3-030-36720-6.
- CASATI, J. ao P. B.; ALTAFIM, R. A. P.; ALTAFIM, R. A. C. Vibration detection of vehicle impact using smartphone accelerometer data and long-short term memory neural network. *In: Anais do XXIII Congresso Brasileiro de Automática*. [S.l.: s.n.]: SBA: Sociedade Brasileira de Automática, 2020. v. 2, n. 1.
- CHEN, Y. *et al.* Vehicle type classification based on convolutional neural network. *In: 2017 Chinese Automation Congress (CAC)*. [S.l.: s.n.], 2017. p. 1898–1901.
- DAS, K.; BEHERA, R. N. A survey on machine learning: Concept, algorithms and applications. *In: International Journal of Innovative Research in Computer and Communication Engineering*. [S.l.: s.n.], 2017. v. 5, p. 1301–1309.
- DEEPLARNING4J. **Deep Learning for Java**. 2020. <https://deeplearning4j.org/>. [Online; accessed 19 May 2020].
- DENG, L.; YU, D. **Deep Learning: Methods and Applications**. [S.l.], 2014. Disponível em: <https://www.microsoft.com/en-us/research/publication/deep-learning-methods-and-applications/>.
- DEY, M. R. *et al.* Magtrack: Detecting road surface condition using smartphone sensors and machine learning. *In: TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*. [S.l.: s.n.], 2019. p. 2485–2489.

- DISTRITO. **Smart Cities: Brasil possui 166 startups focadas em cidades inteligentes**. 2020. https://distrito.me/smart-cities/?gclid=Cj0KCQjwna2FBhDPARIsACAEC_Uta4Xc4Yx6SnKG84fTg1VydaUXdBYVAJZzYjjpae5qmBONAmyLVB0aAtBoEALw_wcB. [Online; accessed 06 Jan 2021].
- EERDEKENS, A. *et al.* Resampling and data augmentation for equines' behaviour classification based on wearable sensor accelerometer data using a convolutional neural network. *In: 2020 International Conference on Omni-layer Intelligent Systems (COINS)*. [*S.l.: s.n.*], 2020. p. 1–6.
- FOSTER, D. **Generative Deep Learning**. 1. ed. [*S.l.: s.n.*]: O'Reilly Media, Inc., 2019. ISBN 9781492041948.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [*S.l.: s.n.*]: MIT Press, 2016. <http://www.deeplearningbook.org>.
- GOODFELLOW, I. J. *et al.* **Generative Adversarial Networks**. 2014.
- GOOGLE. **Google Play Store**. 2021. <https://play.google.com/store/apps/details?id=br.usp.eesc.sel.kaptaccel>. [Online; accessed 10 Apr 2021].
- Gueta, L. B.; Sato, A. Classifying road surface conditions using vibration signals. *In: 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. [*S.l.: s.n.*], 2017. p. 039–043.
- Gunawan, F. E.; Yanfi; Soewito, B. A vibratory-based method for road damage classification. *In: 2015 International Seminar on Intelligent Technology and Its Applications (ISITIA)*. [*S.l.: s.n.*], 2015. p. 1–4.
- HADAD, Y. **30 Amazing Applications of Deep Learning**. 2019. <http://www.yaronhadad.com/deep-learning-most-amazing-applications/>. [Online; acesso em 10 de abril de 2019].
- HAYKIN, S. **Neural Networks and Learning Machines**. 3. ed. [*S.l.: s.n.*]: Pearson Prentice Hall, 2009. ISBN 978-0-13-147139-9.
- HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. **Neural Comput.**, MIT Press, Cambridge, MA, USA, v. 18, n. 7, p. 1527–1554, jul. 2006. ISSN 0899-7667. Disponível em: <http://dx.doi.org/10.1162/neco.2006.18.7.1527>.
- JAIN, A. K.; DUBES, R. C. **Algorithms for Clustering Data**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988. ISBN 0-13-022278-X.
- KANG, E. **Long Short-Term Memory (LSTM): Concept**. Medium, 2017. [Online; accessed 19 April 2020]. Disponível em: <https://medium.com/@kangeugine/long-short-term-memory-lstm-concept-cb3283934359>.
- KERAS. **Keras Documentation**. 2020. <https://keras.io/>. [Online; accessed 21 May 2020].
- KOTLIN. **Kotlin Programming Language**. 2020. <https://kotlinlang.org/>. [Online; accessed 21 Dec 2020].

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *In: Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 1097–1105.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, Springer Nature, v. 521, n. 7553, p. 436–444, may 2015. Disponível em: <https://doi.org/10.1038/nature14539>.

LETTER. **Research Priorities for Robust and Beneficial Artificial Intelligence: An open letter**. 2017. <https://futureoflife.org/ai-open-letter/>. [Online; acesso em 05 de Julho de 2017].

LI, Y.; HU, H.; ZHOU, G. Using data augmentation in continuous authentication on smartphones. *IEEE Internet of Things Journal*, v. 6, n. 1, p. 628–640, 2019.

MIKOŁAJCZYK, A.; GROCHOWSKI, M. Data augmentation for improving deep learning in image classification problem. *In: .* [S.l.: s.n.], 2018. p. 117–122.

NIELSEN, M. A. **Neural Networks and Deep Learning**. [S.l.: s.n.]: Determination Press, 2015. <http://neuralnetworksanddeeplearning.com/>.

NVIDIA. **NVIDIA cuDNN GPU Accelerated Deep Learning**. 2020. <https://developer.nvidia.com/cudnn>. [Online; accessed 22 April 2020].

Obertov, D.; Andrievsky, B. Vehicle classification using measurements from accelerometers mounted on the road surface. *In: 2014 19th International Conference on Methods and Models in Automation and Robotics (MMAR)*. [S.l.: s.n.], 2014. p. 413–417.

POLIZZI, J.-P.; FAIN, B.; MASPERO, F. Chapter 45 - accelerometer. *In: TILLI, M. et al. (ed.). Handbook of Silicon Based MEMS Materials and Technologies (Third Edition)*. Third edition. Elsevier, 2020, (Micro and Nano Technologies). p. 879–898. ISBN 978-0-12-817786-0. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780128177860000451>.

SALAU, H. B. *et al.* A survey of accelerometer-based techniques for road anomalies detection and characterization. **INTERNATIONAL JOURNAL OF ENGINEERING SCIENCE AND APPLICATION**, v. 3, p. 8 – 20, 03 2019.

SERTTAS, T. N.; GEREK, O. N.; HOCAOGLU, F. O. Driver classification using k-means clustering of within-car accelerometer data. *In: 2019 27th Signal Processing and Communications Applications Conference (SIU)*. [S.l.: s.n.], 2019. p. 1–4.

SILVA, I.; SPATTI, D.; FLAUZINO, R. **Redes Neurais Artificiais para engenharia e ciências aplicadas: curso prático**. 1. ed. [S.l.: s.n.]: Artliber, 2010.

SÁNCHEZ-CORCUERA, R. *et al.* Smart cities survey: Technologies, application domains and challenges for the cities of the future. **International Journal of Distributed Sensor Networks**, v. 15, n. 6, p. 1550147719853984, 2019. Disponível em: <https://doi.org/10.1177/1550147719853984>.

TEFFÉ, C. de; BRANCO, S.; VICENTE, V. **Cidades inteligentes em perspectivas**. 1. ed. [S.l.: s.n.]: Obliq, 2019.

TENSORFLOW. **Tensorflow**: An open-source software library for machine intelligence. 2020. <https://www.tensorflow.org/>. [Online; accessed 11 April 2020].

- THEANO. **Theano**. 2020. <http://www.deeplearning.net/software/theano/>. [Online; accessed 14 May 2020].
- THOMAZINI, D.; ALBUQUERQUE, P. U. B. de. **Sensores Industriais: Fundamentos e Aplicações**. 9. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2020. ISBN 978-85-365-XXXX-X.
- TRAN, D.-T.; LUU, H.; NGUYEN, T. A. A real-time vibration monitoring for vehicle based on 3-dof mems accelerometer. *In: . [S.l.: s.n.]*, 2010.
- TRAN, N. *et al.* Towards good practices for data augmentation in GAN training. **CoRR**, abs/2006.05338, 2020. Disponível em: <https://arxiv.org/abs/2006.05338>.
- TUFEK, N. *et al.* Human action recognition using deep learning methods on limited sensory data. **IEEE Sensors Journal**, v. 20, n. 6, p. 3101–3112, 2020.
- ULLMAN, S. *et al.* **Unsupervised learning: Clustering**. 2014. <http://www.mit.edu/~9.54/fall14/slides/Class13.pdf>. [Online; acesso em 01 de Agosto de 2017].
- VAITKUS, V.; LENGVENIS, P.; ŽYLIUS, G. Driving style classification using long-term accelerometer information. *In: 2014 19th International Conference on Methods and Models in Automation and Robotics (MMAR)*. [S.l.: s.n.], 2014. p. 641–644.
- VENDRAMIN, L.; CAMPELLO, R. J. G. B.; HRUSCHKA, E. R. Relative clustering validity criteria: A comparative overview. **Stat. Anal. Data Min.**, John Wiley & Sons, Inc., New York, NY, USA, v. 3, n. 4, p. 209–235, ago. 2010. ISSN 1932-1864. Disponível em: <http://dx.doi.org/10.1002/sam.v3:4>.
- WANG, X. *et al.* Partial discharge pattern recognition with data augmentation based on generative adversarial networks. *In: 2018 Condition Monitoring and Diagnosis (CMD)*. [S.l.: s.n.], 2018. p. 1–4.
- WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data Mining: Practical Machine Learning Tools and Techniques**. 3. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN 0123748569, 9780123748560.
- YIJIANG, W. *et al.* Partial discharge data augmentation of high voltage cables based on the variable noise superposition and generative adversarial network. *In: 2018 International Conference on Power System Technology (POWERCON)*. [S.l.: s.n.], 2018. p. 3855–3859.