

ÁLVARO TEIXEIRA ESCOTTÁ

**RECONHECIMENTO DE EVENTOS DE
DIREÇÃO UTILIZANDO SENSORES INERCIAIS
E TÉCNICAS DE APRENDIZADO DE MÁQUINA**

São Paulo
2022

ÁLVARO TEIXEIRA ESCOTTÁ

**RECONHECIMENTO DE EVENTOS DE
DIREÇÃO UTILIZANDO SENSORES INERCIAIS
E TÉCNICAS DE APRENDIZADO DE MÁQUINA**

Versão Corrigida

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Ciências.

Área de Concentração:

Microeletrônica

Orientador:

Prof. Dr. Wesley Beccaro

São Paulo
2022

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

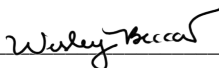
Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 28 de Agosto de 2022

Assinatura do autor:



Assinatura do orientador:



Catálogo-na-publicação

Escottá, Alvaro Teixeira

Reconhecimento de Eventos de Direção Utilizando Sensores Inerciais e Técnicas de Aprendizado de Máquina / A. T. Escottá -- versão corr. -- São Paulo, 2022.

113 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Sensores Inerciais 2.Processamento digital de sinais 3.Sistemas embarcados 4.Aprendizagem de Máquina 5.Aprendizagem Profunda I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

AGRADECIMENTOS

À Deus pelo dom da vida e por me conceder força e perseverança para desenvolver este trabalho. Ao Prof^o Wesley Beccaro, pela orientação, paciência, dedicação e ensinamentos que foram de grande importância. À minha mãe e meu pai, motivo de orgulho, por todas as lições de vida e afeto. Ao meu irmão Alessandro, companheiro de todas as horas. À Carolliny, mulher querida, por todo amor, apoio e compreensão. Aos demais familiares por todo o carinho e vivência. Aos amigos, pelo encorajamento e motivação. A todos os professores que de alguma forma contribuíram para meu aprendizado em mais uma jornada acadêmica. À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) e Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro para publicação do trabalho. À Universidade de São Paulo por todos os recursos fornecidos e serviços prestados.

RESUMO

O monitoramento do comportamento de motoristas tem sido amplamente explorado pelo crescente interesse em sistemas que permitem um maior controle sobre as atividades executadas. Através do reconhecimento de padrões ao volante é possível criar serviços personalizados que visam, dentre algumas aplicações, o controle da taxa de acidentes, precificação de seguros, detecção de sequestros, direção ecológica e gerenciamento de veículos compartilhados e locados. Os aspectos relacionados ao comportamento de motoristas podem ser monitorados a partir de métodos categorizados como recursos visuais e não visuais. No entanto, recursos visuais, baseados em técnicas de visão computacional, apresentam algumas limitações, tais como a violação de privacidade do usuário, a restrição na amplitude do movimento e as interferências do meio externo. Dessa forma, os sensores inerciais (acelerômetros e giroscópios) surgem como um dos meios mais atrativos e utilizados para obtenção de dados que caracterizam o comportamento de motoristas, através dos eventos de direção, que em geral, são descritos pelas ações de acelerar, frear e virar. Este trabalho explora algoritmos tradicionais de aprendizagem de máquina e aprendizagem profunda para realizar a classificação de comportamentos agressivos e não agressivos através dos eventos de direção. Para tanto, são utilizados sinais de aceleração linear e velocidade angular, processando somente as componentes do sinal que apresentam o maior potencial discriminatório para categorizar os eventos. Os modelos de classificação são baseados em *Support Vector Machine* (SVM), Redes Neurais *Multilayer Perceptron* (MLP) e Redes Neurais Convolucionais 1D e 2D. Os modelos baseados em SVM e MLP utilizam como entrada dados resultantes da engenharia de atributos e da implementação do algoritmo *Dynamic Time Warping* (DTW). Os modelos baseados em Redes Neurais Convolucionais 1D e 2D, aplicam a estrutura AlexNet, com pequenas modificações, nos dados pré-processados dos sensores inerciais e nas representações bidimensionais provenientes da conversão dos sinais temporais em gráficos de recorrência. Os resultados demonstram a potencialidade dos métodos abordados, obtendo-se bom desempenho na classificação de eventos de direção com os modelos SVM, MLP, 1D-AlexNet e 2D-AlexNet. Além disso, os resultados são consistentes para avaliação de dados coletados por meio de sensores inerciais e abrangem os eventos de direção mais recorrentes durante a condução veicular, caracterizando o comportamento agressivo dos motoristas.

Palavras-Chave – Sensores Inerciais. Processamento digital de sinais. Sistemas embarcados. Aprendizagem de Máquina. Aprendizagem Profunda.

ABSTRACT

The monitoring of driver behavior has been widely explored by the growing interest in systems that allow greater control over the activities performed. Through pattern recognition at the wheel it is possible to create personalized services that aim, among some applications, to control the accident rate, insurance pricing, kidnapping detection, eco-driving and management of shared and leased vehicles. Aspects related to driver behavior can be monitored using methods categorized as visual and non-visual resources. However, visual resources, based on computer vision techniques, have some limitations, such as violation of user privacy, restriction in the range of motion and interference from the external environment. In this way, inertial sensors (accelerometers and gyroscopes) emerge as one of the most attractive and used means to obtain data that characterize the behavior of drivers, through driving events, which in general are described by the actions of accelerating, braking and to turn. This work explores traditional machine learning and deep learning algorithms to perform the classification of aggressive and non-aggressive behaviors through driving events. For that, linear acceleration and angular velocity signals are used, processing only the signal components that present the greatest discriminatory potential to categorize the events. The classification models are based on Support Vector Machine (SVM), Multilayer Perceptron Neural Networks (MLP) and 1D and 2D Convolutional Neural Networks. Models based on SVM and MLP use data resulting from attribute engineering and implementation of the Dynamic Time Warping (DTW) algorithm as input. The models based on 1D and 2D Convolutional Neural Networks apply the AlexNet structure, with minor modifications, to the pre-processed data of the inertial sensors and the two-dimensional representations resulting from the conversion of temporal signals into recurrence plots. The results demonstrate the potential of the methods discussed, obtaining good performance in the classification of driving events with the SVM, MLP, 1D-AlexNet and 2D-AlexNet models. In addition, the results are consistent for the evaluation of data collected through inertial sensors and cover the most recurrent driving events during vehicular conduct, characterizing the aggressive behavior of drivers.

Keywords – Inertial Sensors. Digital signal processing. Embedded systems. Machine learning. Deep learning.

LISTA DE FIGURAS

1	Etapas de um sistema de inferência do comportamento de motoristas, baseada na abordagem com uso técnicas de engenharia de atributos.	27
2	Exemplo de solução linear com padrões de classes separáveis: (a) solução linear com padrões de classes não linearmente separáveis (b) e solução não linear para classes não linearmente separáveis (c).	31
3	Diagrama do modelo de um neurônio artificial com suas representações funcionais.	32
4	Gráfico das funções de ativação frequentemente utilizadas em RNAs.	33
5	Arquitetura de uma rede MLP de quatro camadas ocultas com 5 neurônios por camada.	34
6	Arquitetura típica de uma CNN para aplicações com dados bidimensionais (por exemplo, imagens).	38
7	Matriz de confusão de um classificador binário relacionado às classes A e B.	40
8	Diagrama em blocos das etapas e processos propostos para o reconhecimento de comportamento de motoristas utilizando sensores inerciais e aprendizagem de máquina.	43
9	Quantidade de amostras coletadas por eventos de direção pertinentes ao conjunto de dados do Júnior e Pessin (2016).	45
10	(a) Sistema de coordenadas do <i>smartphone</i> e (b) sistema de coordenadas fixas.	45
11	Exemplo do método de gráfico de recorrência de séries temporais: (a) representação da série temporal, (b) espaço de fase tridimensional reconstruído e (c) gráfico de recorrência correspondente à série temporal.	55
12	Arquitetura 1D-AlexNet. Cada bloco descreve os hiperparâmetros atribuídos às respectivas camadas, tal como para os blocos “Conv1D” e “MaxPooling1D”.	59

13	Arquitetura 2D-AlexNet. Cada bloco indica a descrição de hiperparâmetros atribuídos às respectivas camadas, tal como para os blocos “Conv2D” e “MaxPooling2D”.	60
14	Local de instalação do <i>smartphone</i> e sistema de referência de coordenadas x , y e z . O eixo x aponta para a esquerda e é perpendicular ao plano de simetria do veículo. O eixo y aponta para frente e é paralelo ao plano de simetria do veículo. O eixo z se estende para baixo. As rotações em torno do eixos x , y e z são sentido anti-horário (regra da mão direita).	61
15	Quantidade de amostras coletadas por eventos de direção pertinentes ao conjunto de dados obtido a partir do arranjo experimental proposto.	63
16	Dados brutos de aceleração linear na direção x (primeira coluna) e y (segunda coluna) e velocidade angular em z (terceira coluna) para eventos não agressivos (primeira linha), curva agressiva à direita e à esquerda (segunda e terceira linhas), troca de faixa agressiva à direita e à esquerda (quarta e quinta linhas), freada agressiva (sexta linha) e aceleração agressiva (sétima linha), respectivamente.	66
17	Dados filtrados de aceleração linear na direção x (primeira coluna) e y (segunda coluna) e velocidade angular em z (terceira coluna) para eventos não agressivos (primeira linha), curva agressiva à direita e à esquerda (segunda e terceira linhas), troca de faixa agressiva à direita e à esquerda (quarta e quinta linhas), freada agressiva (sexta linha) e aceleração agressiva (sétima linha), respectivamente.	69
18	Alinhamento DTW entre os sinais de aceleração na direção x do evento não agressivo de referência e uma amostra de cada classe de evento rotulado. O gráfico da última linha e segunda coluna, apresenta um condição de alinhamento DTW entre o evento não agressivo e aceleração agressiva indicando correspondência entre os sinais.	73
19	Conjunto de atributos em um espaço bidimensional obtido através do método t-SNE (perplexidade igual a 12,7).	74
20	Séries temporais e gráficos de recorrência (no lado direito de cada evento de condução) de sinais de aceleração linear nas direções x e y , primeira e segunda colunas e velocidade angular na direção z , terceira coluna.	75

21	Matriz de confusão do modelo SVM com <i>kernel</i> RBF e parâmetro C igual a 250 (modelo treinado e validado com validação cruzada 10- <i>fold</i>). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados. . . .	77
22	Matriz de confusão do modelo MLP de duas camadas ocultas com 32 neurônios por camada e função de ativação tangente hiperbólica (modelo treinado e validado com validação cruzada 10- <i>fold</i>). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados.	82
23	Matriz de confusão do modelo 1D-AlexNet (modelo treinado e validado com validação cruzada 10- <i>fold</i>). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados.	83
24	Matriz de confusão do modelo 2D-AlexNet (modelo treinado e validado com validação cruzada 10- <i>fold</i>). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados.	84
25	<i>Boxplot</i> das distribuições da métrica acurácia para os 10 particionamentos. <i>Boxplot</i> que se estendem do primeiro quartil, abaixo, ao terceiro quartil, acima. Linha mediana em vermelho e <i>whiskers</i> se estendendo até os valores mínimo e máximo.	84
26	Dados brutos coletados de aceleração linear na direção x (primeira coluna) e y (segunda coluna) e velocidade angular em z (terceira coluna) para eventos não agressivos (primeira linha), curva agressiva à direita e à esquerda (segunda e terceira linhas), troca de faixa agressiva à direita e à esquerda (quarta e quinta linhas), freada agressiva (sexta linha) e aceleração agressiva (sétima linha), respectivamente.	85
27	Dados filtrados coletados de aceleração linear na direção x (primeira coluna) e y (segunda coluna) e velocidade angular em z (terceira coluna) para eventos não agressivos (primeira linha), curva agressiva à direita e à esquerda (segunda e terceira linhas), troca de faixa agressiva à direita e à esquerda (quarta e quinta linhas), freada agressiva (sexta linha) e aceleração agressiva (sétima linha), respectivamente.	86
28	Matriz de confusão do modelo SVM com <i>kernel</i> RBF e parâmetro C igual a 1000 (modelo treinado e validado com validação cruzada 10- <i>fold</i>). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados unificados.	88

29	Matriz de confusão do modelo MLP de duas camadas ocultas com 512 neurônios na primeira camada, 32 neurônios na segunda camada e função de ativação tangente hiperbólica (modelo treinado e validado com validação cruzada 10- <i>fold</i>). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados unificados.	92
30	Matriz de confusão do modelo 1D-AlexNet (modelo treinado e validado com validação cruzada 10- <i>fold</i>). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados unificados.	93
31	Matriz de confusão do modelo 2D-AlexNet (modelo treinado e validado com validação cruzada 10- <i>fold</i>). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados unificados.	94
32	<i>Boxplot</i> das distribuições da métrica acurácia para os 10 particionamentos. <i>Boxplot</i> que se estendem do primeiro quartil, abaixo, ao terceiro quartil, acima. Linha mediana em vermelho e <i>whiskers</i> se estendendo até os valores mínimo e máximo.	94
33	Capturas de tela do <i>phyphox</i> no Android. Em (a) tem-se o menu principal com as opções de acesso aos dados brutos dos sensores e os experimentos com análise de dados, categorizados por tópicos. Em (b) tem-se um exemplo de dados coletados de aceleração linear nas direções x , y e z , a partir do acelerômetro.	113

LISTA DE TABELAS

1	Descrição das bibliotecas Python, com recursos para ciência de dados, utilizadas no desenvolvimento dos classificadores.	46
2	Atributos extraídos e selecionados através do <i>tsfresh</i> e os correspondentes valores-p. Seleção de atributos relevantes através do método de filtros. . . .	70
3	Valores de medida DTW entre os sinais de aceleração na direção <i>x</i> de um evento não agressivo e todos os demais eventos do conjunto de dados. . . .	72
4	Medida de desempenho através das métricas de acurácia e <i>F1-score</i> (médias macro e micro) dos diferentes modelos baseados em SVM. Valores médios obtidos através de <i>10-fold</i>	76
5	Medida de desempenho através das métricas de acurácia e <i>F1-score</i> (médias macro e micro) dos diferentes modelos baseados na rede MLP de uma única camada oculta. Valores médios obtidos através de <i>10-fold</i>	79
6	Medida de desempenho através das métricas de acurácia e <i>F1-score</i> (médias macro e micro) dos diferentes modelos baseados na rede MLP de duas camadas ocultas. Valores médios obtidos através de <i>10-fold</i>	80
7	Medida de desempenho através das métricas de acurácia e <i>F1-score</i> (médias macro e micro) dos diferentes modelos baseados na rede MLP de três camadas ocultas. Valores médios obtidos através de <i>10-fold</i>	81
8	Medida de desempenho através das métricas de acurácia e <i>F1-score</i> (médias macro e micro) do modelo 1D-AlexNet. Valores médios obtidos através de <i>10-fold</i>	81
9	Medida de desempenho através das métricas de acurácia e <i>F1-score</i> (médias macro e micro) do modelo 2D-AlexNet. Valores médios obtidos através de <i>10-fold</i>	81
10	Medida de desempenho através das métricas de acurácia e <i>F1-score</i> (médias macro e micro) dos diferentes modelos baseados em SVM. Avaliação realizada com o conjunto de dados unificados. Valores médios obtidos através de <i>10-fold</i>	87

11	Medida de desempenho através das métricas de acurácia e <i>F1-score</i> (médias macro e micro) dos diferentes modelos baseados na rede MLP de uma única camada oculta. Avaliação realizada com o conjunto de dados unificados. Valores médios obtidos através de <i>10-fold</i>	89
12	Medida de desempenho através das métricas de acurácia e <i>F1-score</i> (médias macro e micro) dos diferentes modelos baseados na rede MLP de duas camadas ocultas. Avaliação realizada com o conjunto de dados unificados. Valores médios obtidos através de <i>10-fold</i>	90
13	Medida de desempenho através das métricas de acurácia e <i>F1-score</i> (médias macro e micro) dos diferentes modelos baseados na rede MLP de três camadas ocultas. Avaliação realizada com o conjunto de dados unificados. Valores médios obtidos através de <i>10-fold</i>	91
14	Medida de desempenho através das métricas de acurácia e <i>F1-score</i> (médias macro e micro) do modelo 1D-AlexNet. Avaliação realizada com o conjunto de dados unificados. Valores médios obtidos através de <i>10-fold</i>	91
15	Medida de desempenho através das métricas de acurácia e <i>F1-score</i> (médias macro e micro) do modelo 2D-AlexNet. Avaliação realizada com o conjunto de dados unificados. Valores médios obtidos através de <i>10-fold</i>	91

LISTA DE SÍMBOLOS

w	Pesos sinápticos e <i>kernel</i>
b	Escalar viés (<i>bias</i>)
x	Valores de dados de entrada
ξ_i	Variáveis soltas não negativas
$J(\mathbf{w})$	Função custo para padrões linearmente separáveis do algoritmo SVM
$J(\mathbf{w}, \xi)$	Função custo para padrões não linearmente separáveis do algoritmo SVM
C	Constante de regularização da função custo $J(\mathbf{w}, \xi)$
$\phi(\mathbf{x})$	Mapeamento não linear das entradas no espaço de características
$f(\cdot)$	Função de ativação
η	Taxa de aprendizagem
$\nabla E(w)$	Gradiente da função custo para redes neurais MLP
$s[n]$	Resultado da operação de convolução
$S[i, j]$	Resultado da operação de convolução bidimensional
I	Entrada de dados bidimensional
K	<i>Kernel</i> bidimensional
VA	Instâncias previstas como Verdadeiro A
VB	Instâncias previstas como Verdadeiro B
FA	Instâncias previstas como Falso A
FB	Instâncias previstas como Falso B
x, y, z	Sistema de coordenadas dos dados de sensores inerciais
m	Número de coeficientes do filtro Savitzky-Golay
C_i	Coefficientes de convolução do filtro Savitzky-Golay
$C_{N \times M}$	Matriz de custos locais do algoritmo DTW
$C(i, j)$	Elementos da matriz de custos locais
$D_{N \times M}$	Matriz de custos acumulados do algoritmo DTW

$D(i, j)$	Elementos da matriz de custos acumulados
ε	Limiar de distância
τ	Atraso de tempo utilizado para reconstrução do espaço de fase
μ	Tamanho da incorporação utilizada para reconstrução do espaço de fase
$r(i, j)$	Estimativa de recorrências entre estados do espaço de fase
\mathbf{R}	Matriz de recorrências
$d(i, j)$	Estimativa de distância entre estados do espaço de fase
\mathbf{D}	Matriz de distâncias
\vec{x}_i	Estados de uma série temporal no espaço de fases correspondente ao instante i
\vec{x}_j	Estados de uma série temporal no espaço de fases correspondente ao instante j
C_{reg}	Conjunto de valores atribuídos ao parâmetro de regularização C
k	Número de divisões do conjunto de dados através do método k -fold
N_1, N_2, N_3	Número de neurônio dos modelos MLP na primeira, segunda e terceira camadas ocultas
ys_j	Saída suavizada pelo filtro Savitzky-Golay
\bar{x}	Valor de média de um conjunto de dados
\bar{x}_p	Valor do quantil de ordem p
α	Intercepto de um modelo de regressão linear
β	Coefficiente angular um modelo de regressão linear
ε_i	Termo de erro de um modelo de regressão linear

LISTA DE ABREVIATURAS

Adam	Estimativa de momento adaptativo (<i>Adaptive Moment Estimation</i>)
CAN	<i>Controller Area Network</i>
CNN	Redes Neurais Convolucionais (<i>Convolutional Neural Network</i>)
CPU	Unidade Central de Processamento (<i>Central Processing Unit</i>)
CSV	<i>Comma-separated values</i>
DTW	<i>Dynamic Time Warping</i>
FFT	Transformada Rápida de Fourier (<i>Fast Fourier Transform</i>)
FN	Falso Negativo (<i>False Negative</i>)
FP	Falso Positivo (<i>False Positive</i>)
GPS	Sistema de Posicionamento Global (<i>Global Positioning System</i>)
GPU	Unidade de Processamento Gráfico (<i>Graphics Processing Units</i>)
JSON	<i>Java Script Object Notation</i>
LIN	<i>Local Interconnect Network</i>
MEMS	Sistemas Microeletrônicos (<i>Micro Electro Mechanical Systems</i>)
MLP	<i>Multilayer Perceptron</i>
OBDII	<i>On-Board Diagnostic</i>
OMS	Organização Mundial de Saúde
OvR	<i>One-vs-Rest</i>
PHYD	<i>Pay-How-You-Drive</i>
PIB	Produto Interno Bruto
RAM	<i>Random Access Memory</i>
RBF	Função Base Radial (<i>Radial Basis Function</i>)
ReLU	<i>Rectified Linear Unit</i>
RFE	<i>Recursive Feature Elimination</i>
RFECV	<i>Recursive Feature Elimination and Cross-Validated</i>
RNA	Redes Neurais Artificiais

ROC	<i>Receiver Operating Characteristic</i>
ROC-AUC	<i>Area Under the ROC Curve</i>
SGD	Gradiente Descendente Estocástico (<i>Stochastic Gradient Descent</i>)
SVM	<i>Support Vector Machine</i>
Tanh	Tangente hiperbólica
TP	Verdadeiro Positivo (<i>True Positive</i>)
TPR	<i>True Positive Rate</i>
t-SNE	<i>t-Distributed Stochastic Neighbor Embedding</i>
UBI	<i>Usage-Based Insurance</i>

SUMÁRIO

1	INTRODUÇÃO	18
1.1	Motivação e Justificativas	18
1.2	Objetivos	20
1.3	Estrutura do Texto	21
2	RECONHECIMENTO DE PADRÕES UTILIZANDO SENSORES INERCIAIS: APLICAÇÕES E ALGORITMOS DE CLASSIFICAÇÃO	22
2.1	Caracterização do comportamento de motoristas em estilos de direção	22
2.2	Sensores inerciais para o reconhecimento de comportamento de motoristas e suas aplicações	23
2.3	Sistemas para monitoramento e reconhecimento de comportamento de motoristas baseados em aprendizagem de máquina	27
2.4	Aprendizagem de máquina e o monitoramento de motoristas	28
2.5	Métricas para avaliação de modelos de aprendizagem de máquina aplicados à classificação	38
3	Materiais e Métodos	42
3.1	Etapas propostas para o reconhecimento de eventos de direção	42
3.2	Conjunto de dados para reconhecimento de eventos de direção	43
3.3	Conjunto de dados utilizados para treino e validação dos modelos de classificação	44
3.4	Ferramentas e linguagens de programação utilizadas para o desenvolvimento e validação dos classificadores	46
3.5	Pré-processamento dos dados dos sensores inerciais	47
3.6	Extração de atributos dos sinais filtrados	48
3.7	Técnica de conversão de séries temporais em gráficos de recorrência	53

3.8	Classificação dos eventos de direção baseada no algoritmo SVM	56
3.9	Classificação dos eventos de direção baseada em uma rede MLP	57
3.10	Classificação dos eventos de direção baseada em rede neural profunda CNN 1D e CNN 2D	58
3.11	Conjunto de dados obtido experimentalmente sob condições reais	60
3.12	Classificação dos eventos de direção, a partir dos dados unificados, baseada nos modelos em SVM, MLP, CNN 1D e CNN 2D	63
4	Resultados	65
4.1	Conjunto de dados brutos dos sensores inerciais	65
4.2	Conjunto de dados dos sensores inerciais pré-processados	67
4.3	Vetor de características e dados de entrada dos algoritmos SVM e MLP	68
4.4	Análise dos sinais pré-processados através da técnica de gráficos de recorrência	73
4.5	Desempenho dos classificadores baseado no algoritmo SVM	75
4.6	Desempenho dos classificadores baseado na rede neural MLP	77
4.7	Desempenho do classificador baseado em rede neural profunda CNN 1D, estrutura 1D-AlexNet	78
4.8	Desempenho do classificador baseado em rede neural profunda CNN 2D, estrutura 2D-AlexNet	80
4.9	Dados brutos coletados através do <i>phyphox</i>	82
4.10	Desempenho dos classificadores baseado nos modelos em SVM, MLP, CNN 1D e CNN 2D, diante dos dados unificados	86
5	Conclusão	95
5.1	Trabalho publicado	97
	Referências	98
	Apêndice A – Dados suavizados pelo algoritmo de Savitzky-Golay	108
	Apêndice B – Atributos de maior relevância selecionados através do <i>tsfresh</i>	110

1 INTRODUÇÃO

1.1 Motivação e Justificativas

Acidentes de trânsito são cada vez mais letais em todo o mundo. Em 2018, a Organização Mundial da Saúde (OMS) publicou no relatório *Global status report on road safety* que o número de óbitos anuais relacionados à acidentes atingiu a marca de e 1,35 milhões de pessoas. Além disso, os acidentes de trânsito são a principal causa de mortes de crianças e jovens de 5 a 29 anos, principalmente, pedestres, ciclistas e motociclistas. Estima-se que os custos desses acidentes sejam equivalentes a 3% do Produto Interno Bruto (PIB) para a maioria dos países (ORGANIZATION et al., 2018). A OMS destaca algumas medidas que podem ser aplicadas para reduzir o número de acidentes. Dentre os fatores de riscos associados aos acidentes de trânsito, os principais tratam diretamente de questões relacionadas ao comportamento dos motoristas, afetados pela agressividade, distração, sonolência e embriaguez. O comportamento de motoristas potencialmente agressivos é uma das principais causas que levam às condições inseguras no trânsito e, por conseguinte, aumento do número de fatalidades (SHAHVERDY et al., 2020). Com isso, monitorar o comportamento de motoristas através do reconhecimento de padrões de eventos agressivos e não agressivos passou a ter uma significativa contribuição na tentativa de aumentar a segurança viária.

O monitoramento de eventos de direção é uma atividade que demanda um processo de coleta automática de dados de direção como, por exemplo, velocidade, aceleração, frenagem e localização, de modo que esses dados sejam processados e submetidos a algum modelo computacional para classificação do estilo de direção. A aquisição dos dados de direção pode ser realizada, basicamente, por sensores inerciais, acelerômetros e giroscópios, e por outros sensores, tais como magnetômetros. Para isto, podem ser utilizados dispositivos como *smartphones* ou eletrônica embarcada no próprio veículo.

Outros dispositivos podem associar dados coletados nos barramentos de comunicação dos automóveis, como é o caso do *Controller Area Network* (CAN) e *Local Interconnect*

Network (LIN) (CARVALHO et al., 2017). Nestes casos é possível explorar os dados, como a rotação do motor, consumo de combustível, nível de oxigênio medido na sonda lambda, que são disponibilizados em tempo real através do barramento de comunicação.

Muitos sistemas eletrônicos monitoram o comportamento dos motoristas de forma isolada (JOHNSON; TRIVEDI, 2011; DAI et al., 2010; CARVALHO et al., 2017) e alguns combinam a análise com fatores ambientais, estado do veículo (AL-SULTAN; AL-BAYATTI; ZEDAN, 2013) ou condições específicas das estradas (ALI; ATIA; MOSTAFA, 2017). Além disso, a maioria destes sistemas avaliam apenas um estado anormal de direção como: agressivo (JOHNSON; TRIVEDI, 2011), sob efeito de álcool (DAI et al., 2010), distraído (TRAN et al., 2018) e sonolento (SHAHVERDY et al., 2020), utilizando, majoritariamente, fusão de sensores.

Atualmente, sistemas com base no monitoramento de motoristas são objeto de interesse de operadoras de seguros de automóveis, locadoras de veículos e empresas de serviços compartilhados. Por sua vez, conceitos como *Usage-Based Insurance* (UBI) e *Pay-How-You-Drive* (PHYD) foram concebidos gerando uma cobrança ao serviço em conformidade com o perfil de direção dos motoristas. Dessa forma, é possível atuar com maior precisão na precificação de seguros, garantindo redução dos custos à condutores que obtém bons rótulos de direção e não mais aplicando uma avaliação generalizada baseada em fatores estatísticos como, por exemplo, idade, gênero e estado civil. Circunstâncias como roubo do veículo ou desonestidade em reivindicações do seguro, também podem ser identificadas por esses sistemas. Os gerenciamentos de fretes, automatizados e contínuos, também foram beneficiados por sistemas que monitoram o comportamento de motoristas, viabilizando a institucionalização de campanhas para melhorar a pontuação dos condutores e, conseqüentemente, reduzir a taxa de acidentes (CARVALHO et al., 2017). Há, ainda, um potencial de aplicações do monitoramento do comportamento de motoristas em sistemas de transportes inteligentes, detecção de sequestro, veículos autônomos, monitoramento de condições da estrada e veículos, ações para direção ecológica¹ e melhoria no conforto de direção.

A classificação dos estilos de direção está relacionada a fatores que envolvem às tomadas de decisões dos motoristas durante os eventos de direção. Grande parte dos problemas de classificação de comportamento de motoristas aplicam métodos tradicionais de aprendizagem de máquina (AL-SULTAN; AL-BAYATTI; ZEDAN, 2013; DAI et al., 2010). No entanto, muitos estudos têm explorado o uso de redes neurais profundas na classificação

¹*Ecodrive* ou direção ecológica é o movimento que estimula motoristas na melhoria das práticas de direção para garantir a redução do consumo de combustível, da emissão de poluentes e de acidentes.

de estilos de direção, uma vez que os resultados obtidos são promissores (SHAHVERDY et al., 2020; TRAN et al., 2018; ZHANG et al., 2019; JO et al., 2014).

Diante do contexto apresentado, este trabalho propõe duas abordagens para o reconhecimento do comportamento agressivo de motoristas através dos eventos de direção. Para tanto, são utilizados os sinais relevantes dos sensores inerciais (isto é, acelerômetro e giroscópio). A primeira abordagem utiliza os dados obtidos das estratégias de engenharia de atributos e do algoritmo DTW para avaliar modelos tradicionais de aprendizagem de máquina. Neste contexto, os modelos de classificação são baseados em SVM e MLP. A segunda abordagem propõe a classificação de eventos de direção utilizando aprendizagem profunda, em específico as CNNs. O modelo é baseado na arquitetura AlexNet com pequenas modificações como, por exemplo, o tamanho das entradas e saídas. A avaliação é realizada com dados unidimensionais e bidimensionais, sendo este, obtido da transformação de séries temporais em gráficos de recorrência.

1.2 Objetivos

O objetivo principal deste trabalho é avaliar modelos para o reconhecimento de comportamentos agressivos de motoristas, através de eventos de direção, utilizando dados de sensores inerciais e diferentes estratégias de aprendizagem de máquina.

Como objetivos específicos estão:

- Revisar através de um levantamento bibliográfico as diferentes técnicas e algoritmos de reconhecimento de padrões ao volante, utilizando sinais de sensores inerciais.
- Apresentar as estratégias de reconhecimento de padrões utilizando diferentes estratégias de aprendizagem de máquina.
- Desenvolver algoritmos para pré-processamento dos dados e posterior extração e seleção de atributos, avaliando a pertinência de cada atributo proposto em face de algoritmos tradicionais de aprendizagem de máquina.
- Avaliar as diferentes métricas de reconhecimento de padrões para a aplicação proposta, com foco na avaliação do desempenho dos modelos propostos.
- Propor um arranjo experimental para coleta de dados de eventos de direção com base em sensores inerciais de um *smartphone* utilizando um aplicativo Android.

- Realizar sob condições reais a aquisição de dados de eventos de direção, incluindo a construção de um conjunto de dados rotulados.
- Demonstrar a capacidade dos modelos de aprendizagem profunda para operar em estratégias fim-a-fim (*end-to-end*).

1.3 Estrutura do Texto

Este primeiro capítulo apresenta a motivação deste trabalho, bem como a justificativa e os objetivos principais e específicos da dissertação.

No segundo capítulo é feita uma revisão bibliográfica sobre a caracterização de estilos de direção, aspectos teóricos de sensores inerciais e diferentes técnicas de aprendizagem de máquina, baseados em SVM e Redes Neurais Artificiais (RNAs), aplicadas ao reconhecimento de padrões ao volante.

O terceiro capítulo descreve os materiais e métodos utilizados neste trabalho, apresentando o banco de dados, e as ferramentas e linguagens de programação utilizados, além das estratégias que envolvem a engenharia de atributos e as avaliações de desempenho dos modelos aplicados no reconhecimento do comportamento agressivo ao volante através dos eventos de direção. Ainda no capítulo é apresentada a metodologia utilizada para a coleta de outros dados de direção e posterior análise de desempenho dos modelos.

No quarto capítulo são apresentados os resultados de todo o desenvolvimento que envolve as avaliações de desempenho dos modelos propostos. São apresentados resultados referentes ao pré-processamento dos dados, às técnicas de extração e seleção de atributos e a transformação das séries temporais em gráficos de recorrência. Além disso, é apresentada a avaliação dos modelos diante do conjunto de dados inicial e, posteriormente, diante de um conjunto de dados resultante da unificação entre os dados iniciais e os dados obtidos experimentalmente.

O quinto capítulo apresenta as conclusões do trabalho, detalhando os benefícios, as limitações e a potencialidade da metodologia proposta. Por fim, é mencionado o trabalho publicado decorrente dos resultados obtidos durante o desenvolvimento da dissertação.

2 RECONHECIMENTO DE PADRÕES UTILIZANDO SENSORES INERCIAIS: APLICAÇÕES E ALGORITMOS DE CLASSIFICAÇÃO

Este capítulo apresenta uma revisão dos principais métodos de reconhecimento de eventos de direção e comportamento de motoristas utilizando sensores inerciais e técnicas de aprendizagem de máquina. Dentre as técnicas de aprendizagem de máquina, são destacados os algoritmos tradicionais tal como o SVM e as redes neurais MLP e algoritmos de aprendizagem profunda, com ênfase nas Redes Neurais Convolucionais.

2.1 Caracterização do comportamento de motoristas em estilos de direção

Os perfis de motoristas, sugeridos na literatura, podem ser caracterizados através dos comportamentos definidos pelos diferentes hábitos, ações e regras durante a condução veicular. Em geral, pode-se classificar os perfis de estilos em: condução normal ou segura, condução agressiva, condução distraída, condução sonolenta e condução no estado de embriaguez (SHAHVERDY et al., 2020).

As características de comportamentos de motoristas, dos diferentes perfis, são revisadas por Meiring e Myburgh (2015). Nele, identifica-se que uma condução agressiva está vinculada à fatores tais como impaciência ou aborrecimento do motorista e, ainda, a tentativa de minimização do tempo de viagem. Esses comportamentos estão associados às alterações de velocidades irregulares e instantâneas, aceleração e desaceleração excessiva, mudanças rápidas na posição lateral do veículo, e mudanças inadequadas de faixa (MEIRING; MYBURGH, 2015).

Uma condução distraída ocorre por um comportamento transitório do motorista que, normalmente, segue de uma reação rápida procurando corrigir o efeito resultante da dis-

tração. Fatores que geram um padrão distraído envolvem a interação com objetos no interior do veículo (por exemplo, manipulação de *smartphones*), a ocorrência de eventos dentro ou fora do veículo e ações simultâneas à condução como beber e comer. Ou seja, a condução distraída tem uma característica instantânea e esporádica, diferentemente da condução agressiva que apresenta um mau comportamento por um período de tempo mais longo (MEIRING; MYBURGH, 2015).

Um motorista cansado apresenta condições observáveis como bocejos, reações e respostas atrasadas, lapsos de concentração e respiração superficial. Estes fatos caracterizam a condução sonolenta que implicam no raro uso dos freios, invasão de faixas, mudança lenta na aceleração ou marcha e, normalmente, com uma velocidade abaixo do limite especificado (MEIRING; MYBURGH, 2015; SHAHVERDY et al., 2020).

Por outro lado, a condução no estado de embriaguez gera aspectos de uma condução agressiva. No entanto, devido à influência do álcool, há uma degradação de desempenho do motorista com relação a sua normalidade. Esses aspectos, podem classificar o condutor como embriagado (MEIRING; MYBURGH, 2015).

Outra área mensurável de comportamento de motorista é a condução segura, fazendo necessária sua definição para representação de um comportamento de referência, a fim de identificar comportamentos divergentes das condutas agressivas, distraídas, sonolentas e embriagadas (MEIRING; MYBURGH, 2015). Uma condução segura deve evitar condições relacionadas às mudanças rápidas de faixa, aumento rápido da velocidade e aceleração, desatenção ao dirigir, dirigir sobre influência de álcool e manutenção inadequada da posição lateral do veículo.

2.2 Sensores inerciais para o reconhecimento de comportamento de motoristas e suas aplicações

Os aspectos relacionados ao comportamento de motoristas, que envolvem fatores humanos e caracterizam os estilos de direção, são monitorados a partir de métodos categorizados em: recursos não visuais e recursos visuais. Os métodos baseados em recursos não visuais são aqueles que fazem uso de sinais de velocidade, aceleração, rotação, frequência de uso dos freios, entre outros (CARVALHO et al., 2017; DAI et al., 2010; JOHNSON; TRIVEDI, 2011; FERREIRA JÚNIOR et al., 2017; YU et al., 2016). Por outro lado, os métodos baseados em recursos visuais utilizam técnicas de visão computacional para identificar uma anormalidade presente em características físicas observáveis dos motoris-

tas como ações oculares, monitoração sob eventos de distração, entre outros (GALARZA et al., 2018; SUN et al., 2014; HARIRI et al., 2012; CYGANNEK; GRUSZCZYŃSKI, 2014).

Com relação aos métodos de recursos não visuais, os sensores inerciais são um dos meios de monitoramento mais atrativos e utilizados para obtenção de dados que caracterizam eventos de direção e o comportamento de motoristas. Dentro da categoria de sensores inerciais, os dispositivos mais aplicados são os acelerômetros e giroscópios. Uma vez que eventos de direção podem ser resumidos em acelerar, frear e virar, a composição destes eventos podem representar manobras complexas, onde cada motorista reage com diferentes estímulos quando submetidos a estes eventos, permitindo a identificação de estilos de direção.

Os sensores inerciais são amplamente utilizados no reconhecimento de movimentos humanos, na busca por informações relacionadas ao comportamento, saúde, estado de humor, desempenho de tarefas, entre outras. Os sistemas concebidos por sensores inerciais podem superar limitações presentes nos sistemas de reconhecimento que utilizam técnicas de visão computacional (XING et al., 2020). Tais limitações correspondem à violação da privacidade, possibilidade de *spoofing*¹ nos sistemas baseados em câmeras (SHAHVERDY et al., 2020), limitação de amplitude do movimento, complexidade do movimento no tempo e espaço e interferências do meio externo (XING et al., 2020). Grande parte dos sistemas que reconhecem perfis de motoristas utilizam a combinação de diferentes sensores, também conhecido por fusão de sensores.

Um dos primeiros sistemas com uso de sensores inerciais para o monitoramento de comportamento de motoristas, especificamente na detecção de embriaguez, foi proposto por Dai et al. (2010). O sistema utiliza um *smartphone* que coleta e analisa os dados a partir do seu acelerômetro detectando manobras anormais ou perigosas relacionadas à influência do álcool.

Uma classificação de estilos de direção agressivos e não agressivos, foi proposta por Johnson e Trivedi (2011) através da incorporação de um aplicativo, denominado por MIROAD, implementado na plataforma do *smartphone* iPhone. O sistema é baseado na fusão de sensores, sendo estes, acelerômetro, giroscópio e magnetômetro, além do uso de câmera traseira e *Global Positioning System* (GPS). O sistema opera em tempo real, no qual os dados dos sensores e as imagens são registrados assim que eventos potencialmente agressivos são detectados.

¹*Spoofing* refere-se a um tipo de crime virtual que acontece quando um cibercriminoso se passa por um contato ou fonte conhecida.

Eren et al. (2012) propõe um sistema para classificação do comportamento de motoristas baseados nos eventos de direção rotulados como seguros e não seguros. A detecção de eventos como trocas de faixa, curvas, freadas e acelerações agressivas são realizadas utilizando acelerômetro, giroscópio e o magnetômetro. Os eventos são comparados com modelos conhecidos através do algoritmo DTW e são classificados utilizando classificadores bayesianos.

Castignani et al. (2015) propõe uma plataforma de monitoramento de motoristas (*SenseFleet*) que visa detectar manobras arriscadas (por exemplo, aceleração, frenagem e curvas) e fornecer uma pontuação que reflete o desempenho do motorista. Os dados de eventos de direção são obtidos por sensores acelerômetro, giroscópio, magnetômetro e GPS. Um sistema de inferência *Fuzzy* é utilizado para detectar os eventos. Com base nos eventos identificados, uma função de pontuação atribui dinamicamente valor de 0 a 100, enquanto o motorista dirige.

Shahverdy et al. (2020) propõe um método de detecção e classificação dos comportamentos de direção segura, agressiva, sonolenta, distraída e embriagada, utilizando aprendizagem profunda. O sistema utiliza sinais de aceleração e gravidade que são obtidos a partir de um *smartphone* e sinais do acelerador, velocidade e rotação por minuto (rpm) obtidos do adaptador de diagnóstico integrado (OBDII). Esses sinais são convertidos em gráficos de recorrência e os dados gerados são direcionadas a uma Rede Neural Convolutiva para classificar o comportamento de motoristas.

2.2.1 Tecnologias de sensores inerciais

Como descrito anteriormente, a literatura indica diversos métodos de monitoramento do comportamento de motoristas através de sensores embarcados nos veículos, nas rodovias ou em *smartphones* pessoais. Grande parte dessas soluções combinam os sensores inerciais, tais como acelerômetros e giroscópios, com outros tipos de sensores, cujos mais utilizados são: magnetômetros (JOHNSON; TRIVEDI, 2011; EREN et al., 2012) e GPS para sistema de navegação inercial (DI LECCE; CALABRESE, 2009; CASTELLANOS; SUSIN; FRUETT, 2011); velocímetros (DAPZOL, 2005); câmeras (JAIN; BUSSO, 2011), microfones (LI; JAIN; BUSSO, 2013); e leitores de diagnóstico OBDII (IMKAMON et al., 2008; SHAHVERDY et al., 2020).

O desenvolvimento da tecnologia MEMS (*Micro Electro Mechanical Systems*) permitiu a aplicabilidade dos sensores inerciais no campo da microeletrônica, através de uma solução compacta e robusta. Por essas características, diversos sistemas eletrônicos são

concebidos com sensores inerciais MEMS. Em geral, os sensores inerciais são encontrados em automóveis (detecção de colisão e controle de *airbags*), eletrônicos de consumo como os *notebooks* (proteção do *hard disk*), *smartphones* (orientação do visor), câmeras fotográficas (estabilização de imagens), indústria (detecção de falhas e vibrações) e indústria aeroespacial (navegação e orientação de mísseis).

Os acelerômetros são dispositivos eletromecânicos que medem a aceleração de um corpo que está sujeito à uma força externa aplicada. Podem ser utilizados também para medição de inclinação e vibração. O princípio estrutural do acelerômetro envolve o conceito de um sistema massa-mola-amortecedor, onde a deflexão do sistema é relacionada à aceleração. Os acelerômetros MEMS, são construídos, usualmente, com base nos princípios físicos piezoresistivo ou capacitivo². Em termos práticos, os acelerômetros capacitivos apresentam uma faixa dinâmica de medição baixa e são mais utilizados em monitoramento estrutural e medições de aceleração. Os acelerômetros piezoresistivos apresentam uma faixa dinâmica maior e são mais utilizados em aplicações de impacto/choque.

Os giroscópios são utilizados para medição de velocidade angular. O princípio físico dos giroscópios MEMS é baseado no efeito de Coriolis. A força de Coriolis atua no sistema quando a massa vibrante do sensor é submetida a uma variação na taxa angular do movimento. Uma das formas de traduzir o efeito dessa força é através do princípio capacitivo.

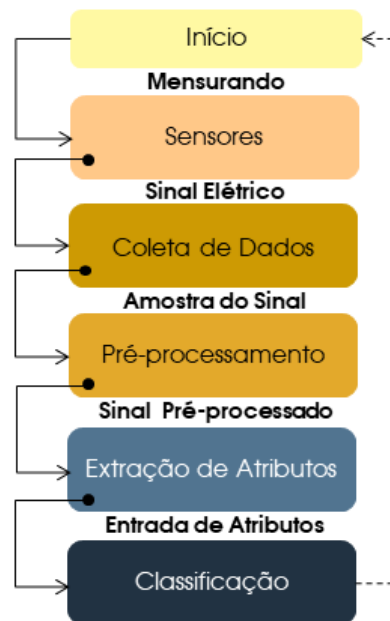
A aceleração e a velocidade angular são variáveis vetoriais, ou seja, possuem intensidade, direção e sentido. Se o sensor inercial for capaz de medir uma componente do mensurando, ele é caracterizado como uniaxial. Da mesma forma, para medições de duas e três componentes, os sensores caracterizam-se como biaxial e triaxial, respectivamente. A combinação entre acelerômetros e giroscópios resultam em dispositivos com até 6 graus de liberdade, conhecidos como unidades inerciais ou módulos inerciais. A partir dessas características, são amplamente utilizados na estabilização, posicionamento e controle de sistemas (MAENAKA, 2008).

²Os acelerômetros também podem ser obtidos utilizando outros efeitos, tais como: piezoelétrico, por efeito Hall, magnetoresistivo, ou por redes de Bragg em fibras ópticas. Para a implementação em tecnologia MEMS, são utilizados os efeitos apresentados anteriormente.

2.3 Sistemas para monitoramento e reconhecimento de comportamento de motoristas baseados em aprendizagem de máquina

O reconhecimento de comportamento de motoristas, trata-se de uma aplicação, cujo o principal objetivo é a classificação de motoristas dentro de categorias definidas pelos estilos de direção. Um sistema de classificação de comportamento de motoristas, baseado em aprendizagem de máquina, é caracterizado por uma sequência de etapas representadas na Figura 1. As etapas apresentadas são características de uma abordagem clássica que faz uso das estratégias de engenharia de atributos.

Figura 1: Etapas de um sistema de inferência do comportamento de motoristas, baseada na abordagem com uso técnicas de engenharia de atributos.



Fonte: Adaptado de (PÄRKKÄ, 2011).

A dinâmica entre as etapas inicia-se com a definição do mensurando de interesse que será detectado pelos sensores através de estímulos físicos. Os sensores são instalados no ambiente desejado estabelecendo uma relação de monitoração constante com os usuários. Em seguida, a coleta de dados é realizada através da amostragem do sinal elétrico produzido pelos sensores, resultando em uma sequência discreta no tempo. O objetivo da coleta de dados é gerar conjuntos de informações tal que apresentem variações representativas em diferentes circunstâncias (PÄRKKÄ, 2011).

Os dados brutos dos sensores podem conter informações indesejadas, como a presença de ruídos, necessitando a remoção na etapa de pré-processamento. Esta etapa pode con-

templar atividades como a filtragem do sinal e, outras menos comuns, como sincronização de dados de diferentes fontes e reamostragem. Com os dados pré-processados, é realizada a extração de atributos para obtenção de representações e informações discriminatórias dos dados e, posteriormente, utilizá-los como entrada nos classificadores.

A partir de algoritmos de aprendizagem de máquina, o classificador é responsável pela saída do sistema à qual contém a informação da classe que representa os dados fornecidos. Além disso, o desempenho do classificador é avaliado utilizando métricas que possibilitam observar o grau de desempenho do sistema ou, ainda, a necessidade de modificações em sua concepção para alcançar melhores resultados. A classificação é beneficiada com atributos que maximizam a distância/divergência entre as classes e minimizam a variabilidade intra-classe (PÄRKKÄ, 2011).

2.4 Aprendizagem de máquina e o monitoramento de motoristas

A detecção de comportamento de motoristas apresenta resultados significativos com uso de algoritmos de aprendizagem de máquina. Meiring e Myburgh (2015) apresentam uma revisão de trabalhos que utilizam algoritmos de aprendizagem de máquina em aplicações de reconhecimento de comportamento de motoristas.

Dentre os trabalhos apresentados na literatura, técnicas de clusterização (agrupamento), especificamente com uso do algoritmo *k-means*, são utilizadas para distinguir estilos de direção (CASTIGNANI et al., 2015; KALSOOM; HALIM, 2013). Nesta mesma linha de aplicação, modelos ocultos de Markov são utilizados, principalmente, na estimativa de comportamento de motoristas (GADEPALLY; KURT et al., 2011; GADEPALLY; KRISHNAMURTHY; OZGUNER, 2013), avaliação de desempenho (BOYRAZ; ACAR; KERR, 2007; MITROVIC, 2005; OLIVER; PENTLAND, 2000), reconhecimento de manobras de direção (BOYRAZ; ACAR; KERR, 2007; MITROVIC, 2005; OLIVER; PENTLAND, 2000) e detecção de perfis distraídos (TAKEDA et al., 2011). Classificadoras bayesianas também são aplicadas na detecção de comportamento de motoristas para resolver problemas de dados incompletos (AL-SULTAN; AL-BAYATTI; ZEDAN, 2013). Árvores de decisão são utilizadas na detecção de perfis sonolentos (WEI et al., 2013). Neste mesmo contexto, modelos SVM são amplamente utilizados na classificação de estilos de direção (WANG et al., 2017), detecção de eventos de condução (VAN LY; MARTIN;

TRIVEDI, 2013; SATHYANARAYANA; SADJADI; HANSEN, 2012), detecção de perfis sonolentos (LI; JIN et al., 2015) e avaliação de estado do veículo (BHORASKAR et al., 2012).

Dentre os principais trabalhos, as RNAs encontram-se aplicadas à detecção de perfis sonolentos, distraídos e previsão de comportamento de direção (DAZA et al., 2011; DEMCENKO et al., 2008; TANGO; BOTTA, 2013) . Algoritmos de aprendizagem profunda (*Deep Learning*) estão aplicados na classificação do nível de estresse de motoristas (RASTGOO et al., 2019) e na classificação de perfis de direção (GALARZA et al., 2018; JO et al., 2014; SABET et al., 2012; AGARWAL, 2019). Redes Neurais Recorrentes como subconjunto de aprendizagem profunda foram aplicadas na classificação de eventos de condução (CARVALHO et al., 2017) e predição de atividades e manobras realizadas pelos motoristas (OLABIYI et al., 2017).

Deste levantamento bibliográfico, observa-se que os métodos de aprendizagem de máquina, usualmente aplicados à classificação de eventos de direção e comportamento de motoristas, estão relacionados, em larga escala, aos classificadores baseados em SVM e RNAs. Além disso, com base no grande sucesso de RNAs, abordagens com arquiteturas utilizando os conceitos de aprendizagem profunda têm sido amplamente aplicadas para reconhecimento de eventos de direção e comportamento de motoristas.

2.4.1 Algoritmo *Support Vector Machine*

O SVM é um algoritmo de aprendizagem supervisionado que pode ser aplicado aos problemas de classificação (BOSER; GUYON; VAPNIK, 1992; VAPNIK, 1998). Neste contexto, o algoritmo SVM visa definir um hiperplano como superfície de decisão de modo que a margem de separação entre as classes distintas seja maximizada (HAYKIN, 2007). A margem de separação é definida pela distância entre uma classe e o hiperplano. O subconjunto dos pontos que geram o hiperplano são designados por vetores de suporte. Após o treinamento, obtém-se um hiperplano ótimo, representando a melhor generalização do classificador.

As classes apresentam padrões de separabilidade no espaço de atributos. Os casos mais simples são os padrões linearmente separáveis, onde o treinamento do SVM determina um hiperplano ótimo que separa os pontos de cada classe, em sua totalidade, com uma margem de separação máxima. Este hiperplano pode ser definido conforme equação 2.1:

$$\mathbf{w}^T \mathbf{x} + b = 0, \quad (2.1)$$

no qual, \mathbf{x} é o vetor de entradas, \mathbf{w} é o vetor de pesos e b é um escalar viés (*bias*).

Supondo um conjunto de treinamento $\{(x_1, y_1), \dots, (x_n, y_n)\}$, com $\mathbf{x}_i \in \mathbb{R}^d$ e $y_i \in \{-1, 1\}$, o algoritmo de treinamento do SVM para padrões linearmente separáveis determina valores ótimos para o vetor de pesos e para o viés a fim de que as restrições, apresentadas nas equações 2.2 e 2.3, sejam satisfeitas durante a minimização da função custo $J(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}$ (CHAVES, 2006).

$$\mathbf{w}^T \mathbf{x}_i + b \geq 1, \text{ para } y_i = +1 \quad (2.2)$$

$$\mathbf{w}^T \mathbf{x}_i + b \leq -1, \text{ para } y_i = -1 \quad (2.3)$$

Em um contexto de maior complexidade, a relação entre as classes conduzem a padrões com menor separabilidade. Assim, a construção de um hiperplano linear é limitada para classificar corretamente todos os pontos. No entanto, pode-se utilizar uma solução com uso de variáveis soltas ou variáveis de folga (*slack variables*) no processo de treinamento, minimizando os erros de classificação. Dessa forma, um vetor \mathbf{x}_i é classificado corretamente em uma das classes de y_i se as restrições apresentadas pelas equações 2.4 e 2.5 forem satisfeitas.

$$\mathbf{w}^T \mathbf{x}_i + b \geq 1 - \xi_i, \text{ para } y_i = +1 \quad (2.4)$$

$$\mathbf{w}^T \mathbf{x}_i + b \leq -1 - \xi_i, \text{ para } y_i = -1 \quad (2.5)$$

no qual, ξ_i são variáveis soltas não negativas.

Com base nos valores de ξ_i é possível descrever espacialmente os pontos de \mathbf{x}_i tomando como referência o hiperplano. Assim, se $0 \leq \xi_i \leq 1$, o ponto \mathbf{x}_i está no lado correto do hiperplano e, logo, a classificação é correta. Se $\xi_i > 1$, o ponto \mathbf{x}_i está do lado incorreto do hiperplano e, logo, a classificação é incorreta.

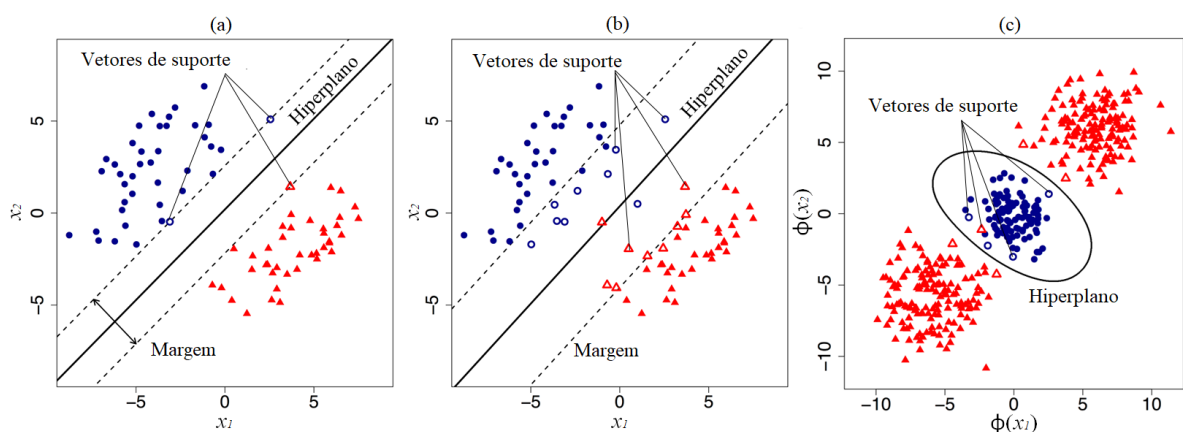
Da mesma forma que nos casos de padrões linearmente separáveis, a construção do SVM para padrões não linearmente separáveis visa a determinação de valores ótimos para o vetor de pesos e viés a fim de que as restrições, apresentada nas equações 2.4 e 2.5, sejam satisfeitas quando a função custo $J(\mathbf{w}, \xi)$ for minimizada (CHAVES, 2006). Neste caso, a função custo $J(\mathbf{w}, \xi) = \frac{1}{2}(\mathbf{w}^T \mathbf{w}) + C \sum_{i=1}^n \xi_i$, com $\mathbf{w} \in \mathbb{R}^d$ e $\xi_i \in \mathbb{R}^+$, apresenta um parâmetro adicional C , conhecido como constante de regularização, que pondera o valor

total do erro. Se C possuir valor elevado, poucos erros serão permitidos, gerando uma margem de separação pequena.

Os dois casos supracitados consideram hiperplanos de separação como combinação linear dos atributos dados. Em casos mais complexos, essa limitação não apresentará uma solução com boa capacidade de separação. Dessa forma, pode-se utilizar SVM não lineares que permitem soluções adequadas a essas condições.

A ideia dos SVM não lineares é adicionar ao produto escalar uma função *kernel* que realizará esta operação mapeando os dados de entrada em um espaço de maior dimensão. Esta modificação melhora a capacidade de distinção das classes (BECARI et al., 2016). As funções *kernels* mais utilizadas em SVMs são: lineares, gaussianas, polinomiais, hiperbólicas e base radial (RBF) (BECARI et al., 2016). Matematicamente, a implementação de um SVM não linear substitui o vetor de entrada \mathbf{x} , das equações apresentadas, por $\phi(\mathbf{x})$, onde $\phi(\cdot)$ é o mapeamento não linear do vetor \mathbf{x} no espaço de atributos. A Figura 2 apresenta exemplos de SVM para solução linear com padrões de classes separáveis (a), solução linear com padrões de classes não linearmente separáveis (b) e solução não linear para classes não linearmente separáveis (c). Nos exemplos, os vetores de suporte são representados por todos os círculos e triângulos não preenchidos e em (c), o hiperplano é definido empregando um *kernel* com função de base radial.

Figura 2: Exemplo de solução linear com padrões de classes separáveis: (a) solução linear com padrões de classes não linearmente separáveis (b) e solução não linear para classes não linearmente separáveis (c).



Fonte: Adaptado de (KIRCHNER; SIGNORINO, 2018).

2.4.2 Redes Neurais Artificiais

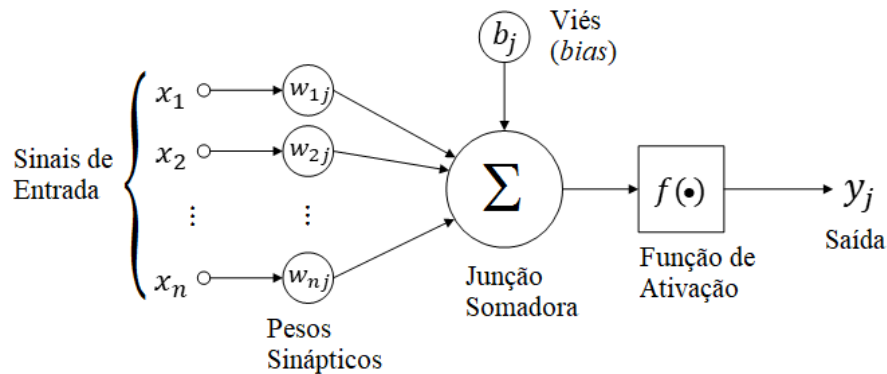
O desenvolvimento de RNAs é inspirado na capacidade do cérebro humano em processar informações complexas, não lineares e paralelas. A realização destes processamentos

é permitida pelo fato do cérebro humano se organizar em neurônios. O cérebro apresenta uma importante propriedade denominada por plasticidade neural, relacionada à capacidade que o sistema nervoso tem de desenvolver novas conexões sinápticas entre os neurônios a partir de experiências.

Segundo Haykin (2007), uma rede neural pode ser definida como um processador paralelo constituído de unidades de processamento simples com a capacidade de armazenar conhecimento experimental e torná-lo disponível para uso. Sua semelhança com o cérebro está atribuída aos aspectos de aquisição do conhecimento através do processo de aprendizagem e ao armazenamento do conhecimento a partir dos pesos sinápticos. Em síntese, o processo de aprendizagem é conseguido pela aplicação de algum algoritmo que visa manipular pesos sinápticos iterativamente. Alguns algoritmos de aprendizagem utilizados em RNAs são: Lei de Hebb, retropropagação (*backpropagation*), estratégia de competição e máquina de Boltzmann.

Diante da analogia neurobiológica, um modelo não linear de neurônio artificial pode ser definido como base para o projeto de redes neurais. A Figura 3 apresenta um diagrama com o modelo de um neurônio artificial. Nesta representação funcional, identifica-se aspectos básicos do modelo neural, sendo estes, a presença de um conjunto de sinapses caracterizadas pelos pesos, um somador de sinais de entradas ponderadas pelas respectivas sinapses e uma função de ativação que limita os sinais de saída dos neurônios ou introduz não linearidade no modelo (HAYKIN, 2007). Além disso, observa-se a presença de um viés (*bias*) que aumenta ou diminui a influência do valor de entrada líquida para ativação do neurônio.

Figura 3: Diagrama do modelo de um neurônio artificial com suas representações funcionais.



Fonte: Adaptado de (VAROL; CANAKCI; OZSAHIN, 2013).

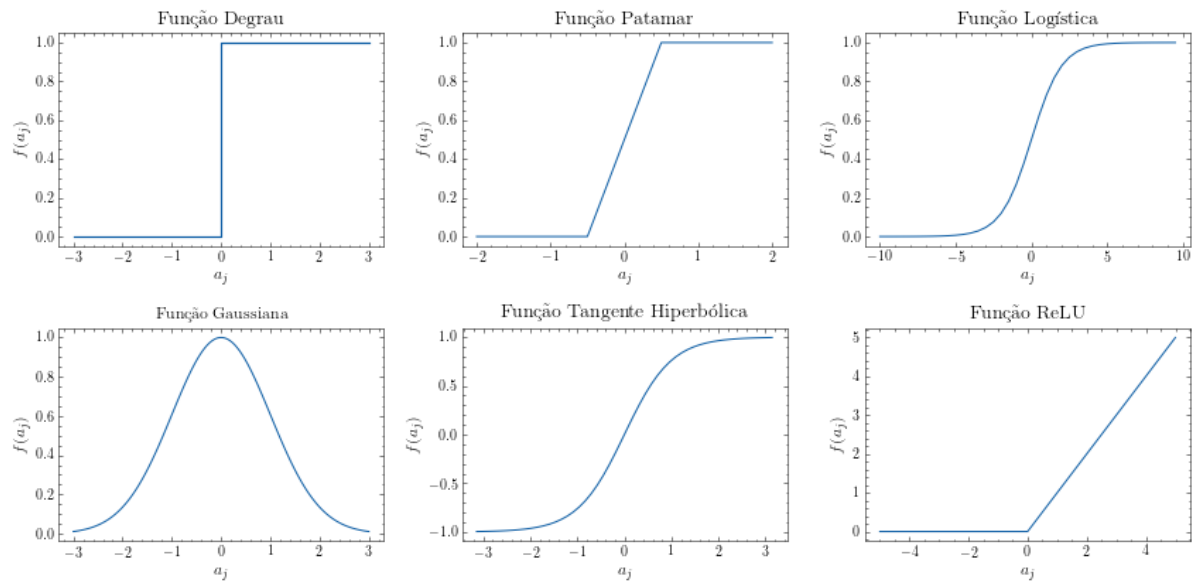
Em termos matemáticos, o neurônio apresentado na Figura 3 pode ser descrito conforme equações 2.6.

$$y_j = f\left(\sum_{i=1}^n (w_{ij}x_i) + b_j\right) \quad (2.6)$$

no qual, x_i indica a i -ésima entrada, w_{ij} é o peso sináptico da conexão entre os neurônios i e j , b_j é o viés (*bias*), $f(\cdot)$ é a função de ativação, y_j é a saída produzida e n indica a quantidade de entradas.

Conforme mencionado, a função de ativação representada por $f(\cdot)$ tem por objetivo modelar a não linearidade da transmissão sináptica entre neurônios. Além disso, algumas funções de ativação limitam a amplitude do sinal de saída dos neurônios. Esta amplitude normalizada, usualmente, encontra-se entre intervalos fechados de $[0,1]$ ou ainda, em alguns casos, entre $[-1,1]$. As funções de ativação mais utilizadas em RNAs são: degrau, patamar, logística, gaussiana, tangente hiperbólica (Tanh) e *Rectified Linear Unit* (ReLU) (PATAN; KORBICZ, 2007; SHARMA, 2017). A Figura 4 ilustra, graficamente, as funções de ativação mais utilizadas.

Figura 4: Gráfico das funções de ativação frequentemente utilizadas em RNAs.



Fonte: Autor, 2022.

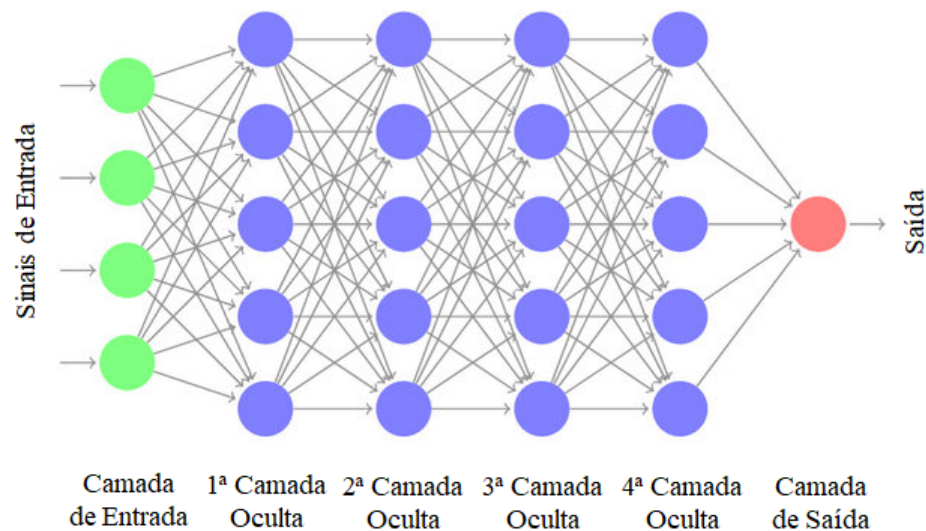
A forma com que os neurônios de uma rede neural estão estruturados define a arquitetura da rede. De modo geral, é possível caracterizar as arquiteturas de rede em três classes fundamentalmente diferentes, sendo estas, redes de camada única, redes de múltiplas camadas e redes recorrentes. A rede de camada única tem por composição: uma camada de entrada, uma camada oculta e a uma camada de saída. A rede com múltiplas camadas se diferem pelo fato de apresentar mais de uma camada oculta de neurônios entre a camada de entrada e a camada de saída.

2.4.3 Redes Neurais *MultiLayer Perceptron*

As redes neurais *MultiLayer Perceptron* apresentam uma arquitetura típica composta de uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída, onde o sinal de entrada propaga para frente, camada por camada. O MLP pode ser aplicado aos problemas de regressão e de classificação com padrões não lineares (CYBENKO, 1989). Dessa forma, são aplicados com satisfatoriedade na resolução de problemas complexos, através do seu treinamento supervisionado utilizando, por exemplo, o algoritmo de retropropagação (*backpropagation* ou *error backpropagation*) (HAYKIN, 2007).

O MLP apresenta características particulares. Como mencionado anteriormente, uma delas corresponde a existência de neurônios ocultos, que permite a rede ser um aproximador universal de funções (CYBENKO, 1989). Cada neurônio da rede tem uma função de ativação não linear, de modo que a relação de entrada/saída não seja reduzida a um perceptron de única camada. Além disso, a composição dos neurônios leva a uma rede totalmente conectada (*fully connected*). A combinação destas características juntamente com a capacidade de aprendizagem por treinamento determinam o poder computacional das MLPs. A Figura 5 apresenta a arquitetura de uma rede MLP de quatro camadas ocultas com 5 neurônios por camada. Observa-se que um neurônio de qualquer camada está conectado com todos os demais neurônios presentes nas camadas anterior e posterior.

Figura 5: Arquitetura de uma rede MLP de quatro camadas ocultas com 5 neurônios por camada.



Fonte: Adaptado de (PÉREZ-ENCISO; ZINGARETTI, 2019).

O treinamento do modelo é realizado em duas etapas, sendo estas, a propagação e a retropropagação. No estágio de propagação, um conjunto de entradas é aplicado à rede,

emergindo camada por camada, até que forme um conjunto de saídas. Nesta condições, os pesos sinápticos são fixos. No caso de um aprendizado supervisionado, esta saída é comparada com o valor alvo (*target*), obtendo-se desta maneira o valor de erro. Durante a retropropagação, esse sinal de erro é propagado para trás e os pesos sinápticos são ajustados. Essas etapas são realizadas até que um determinado conjunto de pesos obtidos forneça um erro mínimo global ou até que outra estratégia de parada do algoritmo seja adotada.

No contexto matemático, o algoritmo *error backpropagation* calcula o gradiente descendente da função custo em relação aos pesos da rede para uma amostra do padrão de treinamento. Por sua vez, a função custo compara através de diversas métricas (por exemplo, erro médio quadrático, erro médio absoluto, divergência de Kullback-Leibler, etc) a saída da rede e o valor alvo. Para tarefas de classificação, a função custo geralmente utilizada é a entropia cruzada (*cross-entropy*).

A atualização dos pesos sinápticos da rede é expressa pela equação

$$w^{\tau+1} = w^{\tau} - \eta \nabla E(w^{\tau}), \quad (2.7)$$

no qual τ é o índice das iterações, $w^{\tau+1}$ é peso sináptico atualizado, w^{τ} é o peso sináptico da iteração anterior, η é a taxa de aprendizagem e $\nabla E(w^{\tau})$ é o gradiente descendente da função custo, sendo $\nabla E(w) = \frac{\delta E}{\delta w_{ij}}$.

A taxa de aprendizagem η é um hiperparâmetro que influencia na convergência do processo de aprendizagem. Em geral, quanto menor for a taxa de aprendizagem, menor será a variação dos pesos sinápticos de uma iteração para outra, proporcionando uma aprendizagem lenta. Por outro lado, uma taxa de aprendizagem maior, apesar de acelerar a aprendizagem, pode gerar instabilidade oscilatória em torno do mínimo local. Uma maneira de obter um aumento da velocidade de aprendizagem sem provocar instabilidade é considerar a constante de momento (ou termo de *momentum*) na atualização dos pesos.

Em geral, não há um método específico que demonstre a convergência do algoritmo *error backpropagation* (HAYKIN, 2007). Com isso, utilizam-se critérios que definem a parada do processo iterativo de ajuste dos pesos. Um desses critérios é verificar se a norma euclidiana do vetor gradiente alcançou um limiar pequeno. Outra possibilidade é verificar se a variação absoluta do erro médio quadrático, entre épocas, é consideravelmente pequena (entre 0,1 a 1%, por exemplo). Por fim, outro critério de convergência é testar a capacidade de generalização da rede a cada iteração e definir a parada do algoritmo quando esta generalização for adequada ou máxima (HAYKIN, 2007).

2.4.4 Algoritmos de *Deep Learning*

Os tradicionais algoritmos de aprendizagem de máquina são insuficientes para aprender funções complexas em um espaço de características de alta dimensão. Sendo assim, o desenvolvimento de *Deep Learning* (aprendizagem profunda) é motivado pelo desafio de generalização desses dados. Tal potencialidade se dá através da concepção estrutural dos modelos de aprendizagem profunda (BENGIO; GOODFELLOW; COURVILLE, 2017), onde camadas e unidades dentro das camadas são adicionadas possibilitando observar padrões e correlações mais específicas em relação ao conjunto de dados.

As principais arquiteturas que apresentam bom desempenho na execução de tarefas de classificação são as Redes Neurais Convolucionais (*Convolutional Neural Network*, CNN), direcionadas, principalmente, ao processamento de imagens. Além disso, estruturas de CNNs unidimensionais tem sido amplamente investigadas para classificação de séries temporais (XIE; REN; ZHU, 2020; WU et al., 2018).

2.4.4.1 Redes Neurais Convolucionais

As redes neurais convolucionais, conhecidas como *ConvNets*, são projetadas, especificamente, para reconhecimento de dados com alto grau de invariância quanto as possíveis formas de distorções, assim como os escalamentos, translações, inclinações, entre outros (HAYKIN, 2007). Sua motivação neurobiológica provém da semelhança entre o padrão das conexões neuronais e a organização do córtex visual dos animais (HUBEL; WIESEL, 1962). As CNNs são redes neurais profundas com a capacidade de extração de características dos dados.

A arquitetura das CNNs é composta, essencialmente, de camadas convolucionais e de camadas *fully connected*. Além disso, outras camadas com funções específicas são utilizadas, tais como as funções de ativação e camadas de operação *pooling*.

As camadas convolucionais são compostas por filtros (*kernels*) que apresentam pesos e vieses (*bias*) ajustáveis. Os filtros realizam operação de convolução com os dados de entrada a fim de extrair atributos. A operação de convolução é representada por (GOODFELLOW; BENGIO; COURVILLE, 2016)

$$s[n] = (x * w)[n] = \sum_{k=-\infty}^{\infty} x[k]w[n - k]. \quad (2.8)$$

na qual, x refere-se ao dados de entrada e w ao *kernel*. Os resultados dão origem aos

mapas de atributos.

Em aplicações, cuja a entrada é representada por uma matriz multidimensional dos dados e o *kernel* é representado por uma matriz multidimensional dos parâmetros, a convolução é realizada em mais de um eixo por vez e pode ser representada por (GOOD-FELLOW; BENGIO; COURVILLE, 2016)

$$S[i, j] = (I * K)[i, j] = \sum_m \sum_n I[m, n]K[i - m, j - n]. \quad (2.9)$$

na qual, I é a entrada de dados bidimensional e K é o *kernel* bidimensional.

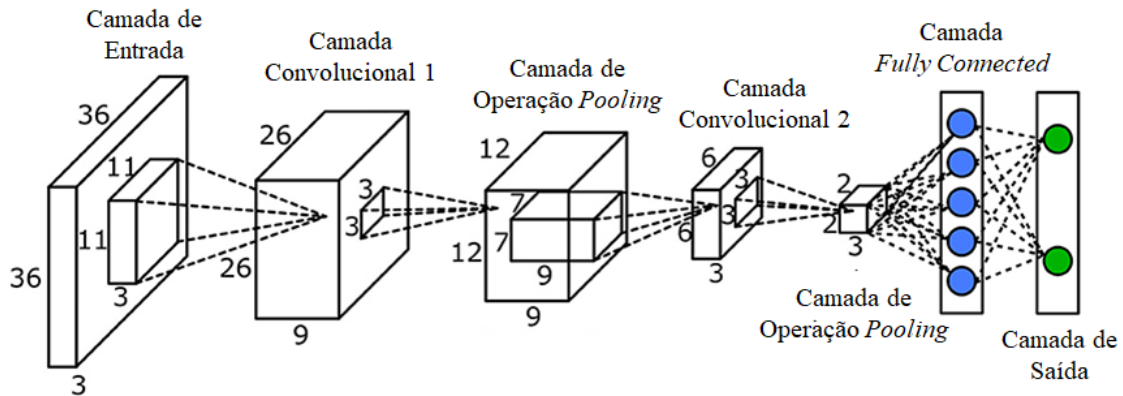
As principais bibliotecas que implementam as CNNs utilizam a operação de correlação cruzada ao invés da convolução. A operação de correlação cruzada é equivalente à convolução, no entanto, o *kernel* não é espelhado (*flipped*) em relação à entrada.

As camadas de operação *pooling* tem como finalidade a redução da dimensão dos atributos que resultam das camadas convolucionais. Dessa forma, é possível reduzir o custo computacional de processamento dos dados e realizar o controle de sobreajuste (*overfitting*). Tipicamente, essa camada funciona como um filtro, de tamanho e passo pré-determinado, que utiliza alguma função para redução de dimensionalidade. As funções mais utilizadas são: *max pooling*, *average pooling* e *sum pooling*. O *max pooling* retorna o valor máximo dentre os dados submetidos ao filtro, enquanto que o *average pooling* retorna a média de todos os dados submetidos ao filtro. Por fim, o *sum pooling* retorna a soma de todos os dados submetidos ao filtro.

As camadas *fully connected* são responsáveis por conceber um modelo capaz de realizar a classificação dos dados resultantes de toda a inferência entre as camadas convolucionais, ReLU e *pooling*. Esses dados são convertidos em um único vetor que alimenta uma RNA do tipo MLP. O processo iterativo de treinamento dessa rede é realizado pelo algoritmo *error backpropagation*. A saída da rede é normalizada por uma distribuição de probabilidades das classes previstas, através de uma função, cuja a mais utilizada é a *Softmax* (em especial a *Log Softmax*). A Figura 6 apresenta uma arquitetura típica de uma rede CNN com hiperparâmetros exemplificados para o mapeamento de imagens.

As CNNs, geralmente, são projetadas para processamento com dados bidimensionais e, portanto, podem ser denominadas por CNNs 2D. Uma versão simplificada das CNNs 2D, denominada por CNNs 1D, é aplicada aos dados unidimensionais, em geral, séries temporais. Neste caso, os filtros e os mapeamentos de atributos são representados por vetores. Segundo Kiranyaz et al. (2019), as CNNs 1D apresentam vantagens em aplicações

Figura 6: Arquitetura típica de uma CNN para aplicações com dados bidimensionais (por exemplo, imagens).



Fonte: Adaptado de (HIRSCHKIND; MOLLICK; PARI, 2020).

com dados de treinamento escassos e implementações em tempo real com a necessidade de baixo custo computacional. As operações de convolução minimizam a complexidade operacional e as etapas de propagação e retropropagação podem ser executadas em paralelo.

2.5 Métricas para avaliação de modelos de aprendizagem de máquina aplicados à classificação

O desenvolvimento de um modelo de aprendizagem de máquina para classificação requer a análise e a avaliação de seus resultados de modo a quantificar o seu desempenho. De modo geral, o desempenho é representado por valores escalares fornecidos por diferentes tipos de métricas ou, ainda, por métodos gráficos. Esta avaliação pode ser aplicada proporcionando diferentes observações (LAVESSON; DAVIDSSON, 2008), bem como a verificação da capacidade dos classificadores de generalizar dados invisíveis e a seleção do melhor modelo dentre os algoritmos treinados. As principais métricas de avaliação de desempenho relacionadas à classificação são: acurácia, precisão, *recall*, *F1-score*, curva ROC (*Receiver Operating Characteristic*) e área abaixo do curva ROC - AUC (*Area Under the ROC Curve*) (NOVAKOVIĆ et al., 2017; THARWAT, 2020; HOSSIN; SULAIMAN, 2015).

A seleção das métricas mais adequadas para realizar a avaliação do classificador depende das características do problema e sua forma de implementação (NOVAKOVIĆ et al., 2017). Além disso, a correta seleção das métricas de avaliação é fundamental para cada tarefa. Para os problemas em que há equilíbrio de dados entre as classes, a acurácia

é uma métrica de boa confiabilidade para mensurar o desempenho do classificador. Por outro lado, se os dados entre as classes apresentam desequilíbrio, a avaliação do classificador passa pela necessidade da utilização das métricas específicas das classes como a precisão, *recall* e o *F1-score*. Além disso, as métricas específicas devem ser aplicadas aos problemas onde há necessidade de minimização de falsos positivos (precisão) e falsos negativos (*recall*), como por exemplo, em análise da diminuição de limite de crédito (AGARWAL, 2019) e detecção da existência de uma nódulo maligno (NOVAKOVIĆ et al., 2017), respectivamente.

Um dos principais indicadores de desempenho para modelos de classificação é a matriz de confusão que possibilita a visualização dos valores previstos pelo modelo em relação aos valores reais das classes. Com isso, é possível verificar os principais erros obtidos pelo modelo e determinar a taxa de acertos ou taxa positiva verdadeira (*True Positive Rate - TPR*) individualmente para as classes. Além disso, com base na matriz de confusão muitas métricas de classificação podem ser obtidas (THARWAT, 2020) tanto para problemas de classificação binária quanto para multiclassés.

Supondo um problema de classificação binária, cujas as classes são A e B, a matriz de confusão é composta por rótulos de Verdadeiro ou Falso para descrever a relação entre as classes previstas e as classes reais. Neste contexto, existem quatro possíveis resultados, sendo estes, Verdadeiro A (VA), Falso A (FA), Verdadeiro B (VB) e Falso B (FB). Os Resultados de VA e VB indicam o número de amostras das classes A e B que estão classificadas corretamente. Por outro lado, o resultado de FA indica o número de amostras da classe B que foram mal classificadas como sendo da classe A. Da mesma forma, o resultado de FB indica o número de amostras da classe A que foram mal classificadas como sendo da classe B. A Figura 2 apresenta um exemplo de matriz de confusão de um classificador binário relacionado às classes A e B. Observa-se que os elementos da diagonal principal da matriz de confusão denotam as previsões corretas das classes, enquanto os demais elementos denotam previsões mal classificadas.

Conforme apresentado em estudos anteriores (CHAWLA; JAPKOWICZ; KOTCZ, 2004; GU; ZHU; CAI, 2009) a acurácia e o *F1-score* são as métricas mais utilizadas para avaliação dos classificadores binários e multiclassés (HOSSIN; SULAIMAN, 2015). A acurácia é a medida da razão entre as amostras classificadas corretamente e o número total de amostras. Com base na matriz de confusão da Figura 2, a acurácia é definida conforme a equação 2.10.

Figura 7: Matriz de confusão de um classificador binário relacionado às classes A e B.

Matriz de Confusão

Classe Real	Classe A	Verdadeiro A (VA)	Falso B (FB)
	Classe B	Falso A (FA)	Verdadeiro B (VB)
		Classe A	Classe B

Classe Prevista

Fonte: Autor, 2022.

$$\text{Acurácia} = \frac{VA + VB}{VA + FA + VB + FB} \quad (2.10)$$

O *F1-score* avalia o desempenho dos modelos pela média harmônica entre a precisão e o *recall*, que são apresentadas na equação 2.11 e 2.12, respectivamente.

$$\text{Precisão} = \frac{VA}{VA + FA} \quad (2.11)$$

$$\text{Recall} = \frac{VA}{VA + FB} \quad (2.12)$$

Dessa forma, o *F1-score* é calculado por

$$F1\text{-score} = 2 \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (2.13)$$

Em uma configuração de multiclases, pode-se calcular a pontuação *F1-score* em uma abordagem *One-vs-Rest (OvR)*, determinando as métricas para cada classe separadamente. Para calcular a média das várias métricas por classe para descrever um desempenho geral, pode-se aplicar média micro, macro e ponderada. No método de média micro, somam-se os Verdadeiro Positivo (*TP*), Falso Positivo (*FP*) e Falso Negativo (*FN*), individuais do modelo para diferentes conjuntos e aplica-se no cálculo para obter a pontuação. O método de média macro é considerado a média da precisão e do *recall* do modelo em

diferentes conjuntos. Finalmente, a média ponderada obtém a média da precisão e recall do modelo em diferentes conjuntos ponderados por suporte (ou seja, número de instâncias verdadeiras para cada rótulo). Em alguns casos, a média ponderada pode resultar em uma pontuação *F1-score* que não está entre precisão e *recall*.

A acurácia é sensível ao desequilíbrio e quantidade de dados, isto é, quando um conjunto de dados apresenta um número de amostras predominantemente maior de uma determinada classe com relação as outras. Além disso, dois classificadores diferentes podem produzir a mesma pontuação de acurácia, no entanto, atuam de forma diferentes com relação as decisões corretas e incorretas (THARWAT, 2020). Neste contexto, a acurácia indica uma métrica de desempenho mais global uma vez que não permite uma análise de distinção do tipos de erros, que em determinadas aplicações são úteis (NOVAKOVIĆ et al., 2017). Por conta disso, a matriz de confusão é um indicador que pode complementar a avaliação de desempenho dos classificadores a partir da acurácia, uma vez que a matriz de confusão permite visualizar os erros obtidos e as previsões realizadas de forma mais específica.

3 MATERIAIS E MÉTODOS

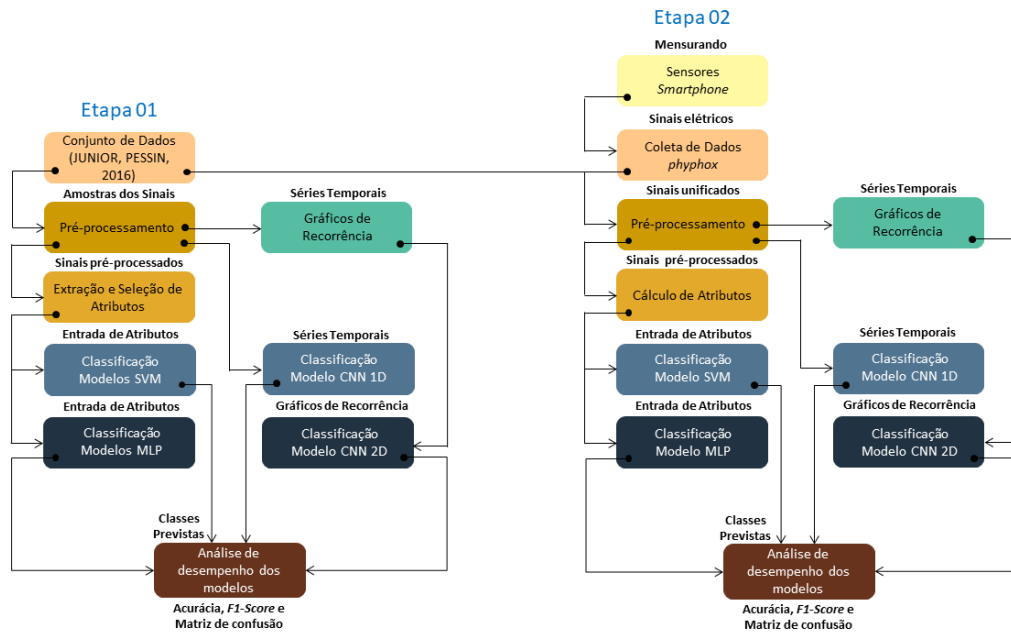
Neste capítulo são delimitados os materiais e métodos utilizados para o desenvolvimento dos modelos de classificação do comportamento de motoristas através dos eventos de direção. Os modelos incluem os algoritmos tradicionais de aprendizagem de máquina, tal como o SVM e MLP e algoritmos de aprendizagem profunda, tal como as CNNs unidimensional e bidimensional. A análise dos modelos é realizada considerando como alvo de classificação os eventos de direção não agressivos, curvas agressivas à direita e à esquerda, trocas de faixa agressivas à direita e à esquerda, freadas agressivas e acelerações agressivas. Além disso, são detalhados o método de pré-processamento dos sinais obtidos pelos sensores inerciais e o processo de extração e seleção dos atributos mais relevantes dos sinais. Por fim, são descritas as considerações utilizadas para concepção dos modelos de classificação, bem como sua análise de desempenho.

3.1 Etapas propostas para o reconhecimento de eventos de direção

A Figura 8 apresenta uma visão geral das etapas e processos propostos para o reconhecimento de eventos de direção e comportamento de motoristas utilizando sensores inerciais e aprendizado de máquina. A abordagem é realizada em duas etapas. Na Etapa 01, os modelos são testados e validados com base no conjunto de dados¹ coletados por Júnior e Pessin (2016). Além disso, a partir desse conjunto de dados, são extraídos e selecionados os atributos mais relevantes dos sinais utilizados. Posteriormente, na Etapa 02, um arranjo experimental é proposto para obter novos dados de eventos de direção complementares aos apresentados anteriormente. Com isso, o desempenho dos modelos é avaliado a partir da unificação entre o conjunto de dados de Júnior e Pessin (2016) e os dados coletados experimentalmente. Nesta condição, as estratégias aplicadas na Etapa 01 são reutilizadas.

¹Banco de dados disponível em: <https://github.com/jair-jr/driverBehaviorDataset>

Figura 8: Diagrama em blocos das etapas e processos propostos para o reconhecimento de comportamento de motoristas utilizando sensores inerciais e aprendizagem de máquina.



Fonte: Autor, 2022.

3.2 Conjunto de dados para reconhecimento de eventos de direção

A investigação sobre a direção agressiva concentra-se, principalmente, em dois aspectos. O primeiro é baseado em questionários sobre os motoristas, visando uma análise diante de fatores no campo da psicologia. O principal problema dessa análise é a dependência de aspectos subjetivos do motorista e não seu desempenho real.

O segundo tipo de investigação propõe o controle do ambiente de direção, impulsionando o motorista a ter um comportamento de direção mais agressivo. Neste contexto, muitos dados de direção são obtidos a partir de simuladores (CHAI; ZHAO, 2016; NING et al., 2009; ÖZEL, 2020). Comparativamente, existem menos bases de dados obtidas sob condições reais de veículos e estradas (MA et al., 2019).

As ações específicas que caracterizam a direção agressiva são visualizadas através dos eventos de direção mais recorrentes durante a condução veicular. As características dos eventos, baseadas nos dados de sensores inerciais, são mais propensas a serem reconhecidas. Existem, disponíveis, poucas bases de dados de eventos de direção agressivos sob condições reais e com dados amostrados de sensores inerciais (YUKSEL, 2020; JÚNIOR; PESSIN, 2016). Yuksel (2020) apresenta um conjunto de dados que emulam apenas

eventos de aceleração, frenagem e curvas repetidas (YUKSEL, 2020). Por outro lado, o conjunto de dados de Júnior e Pessin (2016) é composto por eventos agressivos como curvas (à direita e à esquerda), trocas de faixas (à direita e à esquerda), aceleração, frenagem e eventos não agressivos como classe de referência. Além disso, é visto que essa base de dados foi utilizada por outros trabalhos que propõem uma investigação sobre a condução agressiva (FERREIRA JÚNIOR et al., 2017; MA et al., 2019).

3.3 Conjunto de dados utilizados para treino e validação dos modelos de classificação

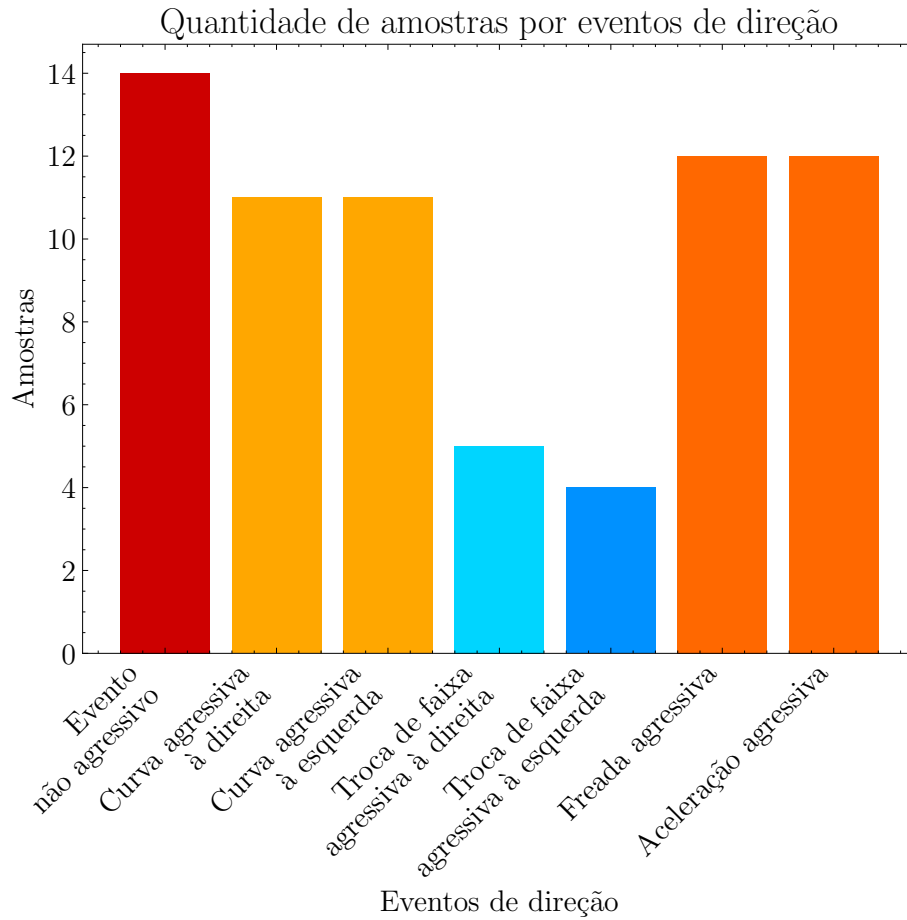
A primeira validação dos modelos de classificação é realizada através do conjunto de dados experimentalmente obtido por Júnior e Pessin (2016). Estes dados foram coletados a partir de um aplicativo Android que registrou amostras de sensores inerciais disponíveis em um *smartphone* Motorola XT1058 na versão Android 5.1. Durante o experimento, o *smartphone* foi fixado ao parabrisa do veículo e não houve qualquer manipulação ou operação ao dispositivo nesse período.

Os eventos de direção foram realizados por dois motoristas em quatro viagens, com duração média de 13 minutos, utilizando um veículo Honda Civic 2011 exposto às condições de pista seca. As manobras foram emuladas a fim de representar um comportamento agressivo nos eventos predefinidos. Dessa forma, o conjunto de dados apresenta 69 amostras subdivididas em 14 eventos não agressivos, 11 eventos de curva agressiva à direita, 11 eventos de curva agressiva à esquerda, 5 eventos de troca de faixa agressiva à direita, 4 eventos de troca de faixa agressiva à esquerda, 12 eventos de freada agressiva e 12 eventos de aceleração agressiva. A Figura 9 apresenta, graficamente, o número de amostras coletadas por classe de eventos de direção pertinentes ao conjunto de dados do Junior e Pessin (2016).

Os sinais extraídos dos sensores inerciais utilizados na coleta de dados foram: aceleração e aceleração linear (acelerômetro), campo magnético (magnetômetro) e aceleração angular (giroscópio). Estes sensores geram séries temporais nos eixos x , y e z , com uma marca de tempo (*timestamp*) de nanosegundos.

Neste conjunto de dados, o sistema de coordenadas x , y e z é fixado a fim de manter o mesmo referencial para todos os dados coletados. Com isso, mesmo que o *smartphone* seja fixado em posições diferentes, os dados são traduzidos para o sistema de coordenadas fixas, conforme a Figura 10, onde em (a) tem-se o sistema de coordenadas atribuídas ao

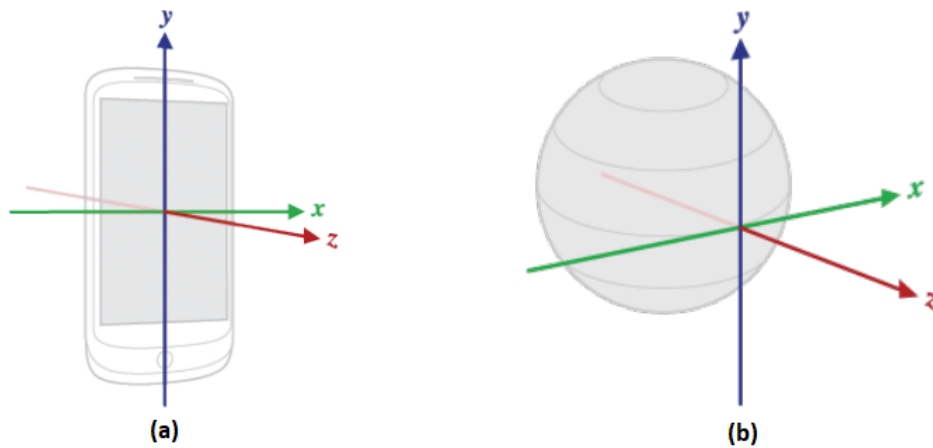
Figura 9: Quantidade de amostras coletadas por eventos de direção pertinentes ao conjunto de dados do Júnior e Pessin (2016).



Fonte: Autor, 2022.

smartphone e em (b) tem-se o sistema de coordenadas fixas.

Figura 10: (a) Sistema de coordenadas do *smartphone* e (b) sistema de coordenadas fixas.



Fonte: Adaptado de (JÚNIOR; PESSIN, 2016).

3.4 Ferramentas e linguagens de programação utilizadas para o desenvolvimento e validação dos classificadores

O desenvolvimento dos modelos de classificação foi construído utilizando-se a linguagem de programação Python². O Python é uma linguagem de alto nível que combina a estrutura simplificada e intuitiva da linguagem ABC, a liberdade da tipagem dinâmica, a orientação a objetos e o suporte à programação funcional. A linguagem Python carrega um conceito baseado em sintaxes concisas e recursos poderosos compostos pela sua biblioteca nativa e pelos módulos e *frameworks* desenvolvidos por terceiros. Esses potenciais tornam o Python muito difundido e útil em diversos setores, assim como no desenvolvimento web, análise de dados e aprendizado de máquina.

O desenvolvimento dos classificadores utiliza diversos recursos e ferramentas Python disponíveis para aplicações em ciência de dados. Dessa forma, as principais bibliotecas utilizadas neste contexto e no desenvolvimento proposto estão descritas na Tabela 1, que apresenta um descritivo funcional das bibliotecas supracitadas e as versões utilizadas no desenvolvimento.

Tabela 1: Descrição das bibliotecas Python, com recursos para ciência de dados, utilizadas no desenvolvimento dos classificadores.

Bibliotecas	Descrição
<i>NumPy</i> (v1.18.5)	Biblioteca científica e numérica que fornece métodos para manipulação de <i>arrays</i> e matrizes multidimensionais, além de uma variedade de operações e funções matemáticas para utilizar com essas estruturas.
<i>SciPy</i> (v1.5.0)	Biblioteca científica que apresenta uma coleção de algoritmos matemáticos e científicos com funções compatíveis com <i>NumPy</i> , possibilitando operações com <i>arrays</i> de forma eficiente.
<i>scikit-learn</i> (v0.23.1)	Biblioteca de aprendizado de máquina que inclui vários algoritmos de classificação, regressão e clusterização. Possui interoperabilidade com as bibliotecas <i>NumPy</i> e <i>SciPy</i> .
<i>pandas</i> (v1.0.5)	Biblioteca que oferece estruturas e operações para manipular tabelas numéricas e séries temporais.
<i>Matplotlib</i> (v3.2.2)	Biblioteca para criação de gráficos e visualizações de dados, desenvolvida para extensão <i>NumPy</i> .
<i>PyTorch</i> (v1.6.0)	Biblioteca de tensores otimizada para aprendizado profundo utilizando GPUs e CPUs.

Fonte: Autor, 2022.

²Esta linguagem foi concebida por Guido Van Rossum, no final de 1989, para suceder a linguagem ABC e dispor uma interface com o sistema operacional Amoeba. Documentação disponível em: docs.python.org

Outra ferramenta utilizada no desenvolvimento é o *Jupyter Notebook* (versão 2.1.5), um ambiente computacional web para execução do Python. Neste ambiente, é possível criar múltiplas linhas e blocos de códigos com diferentes células. A cada código executado é possível visualizar os resultados alcançados. Um documento Jupyter é estruturado no formato *Java Script Object Notation* (JSON) com uma lista de células de entradas e saídas que contém operações matemáticas, textos e gráficos, usualmente, na extensão “.ipynb”.

O processamento dos códigos foi realizado inicialmente utilizando notebook com processador Intel(R) Core(TM) i7-4510U CPU @ 2.00GHz com 8 GB de memória RAM e sistema operacional Windows 10. Para o treinamento dos modelos foi utilizada máquina virtual da ferramenta Google Colab Pro que fornece 27,3 GB de RAM, um processador Intel Xeon CPU 2,20 GHz com 4 núcleos e um acelerador de GPU Tesla P100-PCIE com 16 GB.

3.4.1 *PyTorch*

*PyTorch*³ é uma biblioteca de aprendizado de máquina que oferece recursos de alto nível como: processamento de tensores em unidades de processamento gráfico (GPUs) e construção de redes neurais profundas utilizando métodos de diferenciação automática (Autograd). Os principais módulos que contemplam o *PyTorch* são: *Tensor*, *Autograd*, *Optimizer* e *Neural Network*.

O módulo *Autograd* utiliza o método de diferenciação automática, permitindo o cálculo de gradientes. O módulo *Tensor* implementa a classe dos tensores e seus atributos. Os tensores do *PyTorch* são similares aos *arrays* do *NumPy* e são intercambiáveis entre si. O módulo *Optimizer* contém algoritmos de otimização para ajustes dos parâmetros da rede. Por fim, o módulo *Neural Network* apresenta a definição de classes que são usadas para conceber as redes neurais profundas. Além disso, contém a implementação das funções objetivo, utilizadas para cálculo da perda durante o treinamento.

3.5 Pré-processamento dos dados dos sensores inerciais

Os processos de medição e amostragem estão sujeitos a uma certa quantidade de variação aleatória no sinal. Em específico, sensores inerciais são afetados por ruídos de

³Biblioteca de código aberto desenvolvida principalmente pelo Facebook’s AI Research lab (FAIR). Documentação disponível em: pytorch.org/docs.

alta frequência (HAN et al., 2020). Essas condições são indesejáveis para o processamento do sinal, demandando a utilização de métodos que suavizam essas variações para tornar explícitos o padrão e a tendência dos sinais.

Existem diversos métodos para suavização dos dados relacionados às séries temporais, os mais simples são: filtragem por média móvel ou suavização exponencial (BILLAH et al., 2006; GUIÑÓN et al., 2007). Outro método de filtragem utilizado é o de Savitzky-Golay, que apresenta-se como opção vantajosa aos métodos baseados em média, uma vez que tende a preservar informações de pico e largura dos sinais (GUIÑÓN et al., 2007).

O algoritmo de média móvel retorna um vetor com a média das entradas de uma janela móvel com tamanho fixo. O procedimento é repetido para todo o conjunto de dados. A suavização exponencial é uma média móvel ponderada, onde pesos exponencialmente decrescentes são atribuídos às observações anteriores.

Nesse trabalho foi utilizado o filtro de suavização Savitzky-Golay. Este filtro realiza o ajuste de pontos, sucessivos e adjacentes, utilizando um polinômio de determinada ordem ajustado pelo método dos mínimos quadrados. Os parâmetros utilizados para realizar o ajuste são o tamanho da janela (*frame*) que define o número de pontos dos dados (um inteiro ímpar positivo) e a ordem do polinômio (menor que o tamanho da janela -1). Os valores de melhor ajuste para ordem do polinômio e tamanho da janela são estimados empiricamente, visando a menor distorção possível no sinal original e preservando as informações mais significativas, porém com menor ruído e/ou flutuações. Supondo um conjunto de pontos de um determinado dado, o sinal de saída suavizado pelo filtro Savitzky-Golay pode ser representado por um conjunto de m coeficientes de convolução C_i , conforme detalhado no Apêndice A.

3.6 Extração de atributos dos sinais filtrados

Em aprendizagem de máquina, a extração de atributos (*feature extraction*) é uma forma de redução de dimensionalidade. Essa técnica transforma os dados de entrada em um conjunto de dados reduzidos de características (vetor de características) que apresentam informações relevantes dos dados iniciais. Dessa forma, é possível que a dimensionalidade dos dados seja menor e as informações redundantes ou irrelevantes sejam removidas tornando-as mais adequadas ao processo subsequente de reconhecimento de padrões e posterior generalização.

A utilização de algoritmos de extração de atributos é necessária dependendo do al-

goritmo de aprendizagem de máquina aplicado. No caso de redes neurais profundas, os algoritmos apresentam arquiteturas capazes de extrair características dos dados primários, em uma abordagem conhecida como fim-a-fim (*end-to-end*), como é visto nas redes convolucionais. Em contrapartida, nas aplicações com algoritmos tradicionais de aprendizagem de máquina, a extração de atributos é necessária e os dados extraídos podem ser obtidos no domínio do tempo, da frequência e tempo-frequência.

No que se refere ao domínio do tempo, os algoritmos são aplicados sobre as variáveis discretas no tempo e, portanto, as características são extraídas diretamente da forma de onda do sinal. Alguns atributos estatísticos neste domínio são: média, variância, mediana, integral, derivada primeira, derivada segunda, área, valor máximo, valor mínimo, diferença, curtose, quantil, entropia amostral e coeficientes de ajustes polinomiais (YAN et al., 2015; PÄRKKÄ, 2011; PHINYOMARK et al., 2012).

Com relação ao domínio da frequência, as operações matemáticas ocorrem no espectro de frequência do sinal. A decomposição do sinal no tempo em frequência é, geralmente obtido pelo algoritmo da transformada rápida de Fourier (*Fast Fourier Transform*, FFT). Alguns algoritmos nesse domínio são: potência total, potência média, frequência média, frequência mediana, frequência de pico, razão de frequência e 1º, 2º e 3º momentos espectrais (ALTIN; ER, 2016; PÄRKKÄ, 2011; PHINYOMARK et al., 2012).

Por fim, as características no domínio tempo-frequencial estão relacionadas à representação do espectro de frequência ao longo do tempo (GODFREY et al., 2011).

A extração de atributos é um processo particular que, geralmente, exige uma ampla investigação para alcançar uma forma otimizada de construção do modelo. Neste trabalho, as características dos sinais são extraídas utilizando o pacote *tsfresh*⁴⁵ do Python (CHRIST et al., 2018). Esta ferramenta é dedicada a extração de atributos de séries temporais e possibilita a obtenção de inúmeros atributos baseados em uma combinação de, aproximadamente, 77 algoritmos disponíveis entre os domínios do tempo, frequência e tempo-frequência. Todos os atributos extraídos consideram uma janela com tamanho equivalente ao número total de amostras de cada evento, onde os dados utilizados são baseados nos sinais filtrados. A partir dos atributos calculados, aplica-se técnicas de seleção de atributos para obtenção dos mais relevantes.

⁴Documentação disponível em: tsfresh.readthedocs.io/en/latest/

⁵Outras bibliotecas tais como a *canonical time-series characteristics* (CATCH22), implementada em C com *wrapper* para Python, e a *highly comparative time-series analysis* (*hctsa*), implementada em MATLAB, também realizam a extração de atributos de séries temporais, sendo que ambas são utilizadas em aplicações de extração de informações de sinais biológicos.

3.6.1 Seleção dos atributos extraídos

A seleção de atributos é o processo que visa selecionar, dos atributos extraídos, os mais relevantes para o algoritmo de reconhecimento de padrões. Neste contexto, as principais contribuições da seleção de atributos são a redução do tempo de treinamento e do *overfitting*, e o aprimoramento do desempenho dos modelos. Algumas técnicas de seleção de atributos, para aplicações em aprendizagem supervisionada, são baseadas nos métodos de filtro, de eliminação recursiva de atributos e intrínseco (CHANDRASHEKAR; SAHIN, 2014).

Os métodos por seleção de filtro apresentam algoritmos que utilizam medidas estatísticas, a fim de ordenar os atributos por relevância com base em alguma métrica. Uma das formas de analisar a relevância é a partir da análise do valor-p, que é calculado a partir de um teste estatístico. Esses testes são fundamentados em técnicas estatísticas como: análise de variância (ANOVA), estatísticas qui-quadradas, correlação de Pearson, entre outros. Por outro lado, os métodos por eliminação recursiva de atributos apresentam algoritmos que treinam um determinado modelo com diferentes subconjuntos de atributos e verificam qual combinação gerou o melhor resultado. Neste conceito, alguns algoritmos utilizados são: *Recursive Feature Elimination* (RFE) e *Recursive Feature Elimination and Cross-Validated* (RFECV). Por fim, o método intrínseco compreende algoritmos que realizam a seleção automática de atributos durante a etapa de treinamento como, por exemplo, as Árvores de Decisão.

O método de filtro independe de algoritmos de aprendizagem para seleção de atributos. Em geral, métodos que demandam algum tipo de algoritmo de aprendizagem para seleção de atributos resultam em atributos enviesados para o modelo escolhido. Além disso, a literatura apresenta trabalhos que comprovam o bom funcionamento do método de filtro para diferentes tipos de dados incluindo as séries temporais (CRONE; KOURTIZES, 2010; LIU; SETIONO et al., 1996; LAZAR et al., 2012). Por conta disso, a seleção de atributos realizada neste trabalho utiliza o método de filtro e conta com sua implementação a partir do pacote *tsfresh*⁶.

Os testes estatísticos, utilizados no método de filtro, correspondem a uma regra decisória que permite rejeitar ou não uma hipótese estatística com base nos resultados da amostra. Nesta aplicação, as hipóteses estatísticas são definidas assumindo que os atributos são irrelevantes (hipótese nula) ou relevantes (hipótese alternativa). O valor-p é

⁶Outras bibliotecas tais como a Boruta, SHAP e BorutaSHAP, em Python, também têm sido utilizadas para a seleção de atributos. No contexto do trabalho, deu-se preferência às ferramentas apresentadas pela biblioteca *tsfresh*.

uma probabilidade que representa uma evidência contra a hipótese nula definida. Sendo assim, quanto menor o seu valor, maior será a evidência para rejeitar a hipótese nula e mais relevante será o atributo. Os atributos a serem mantidos são determinados a partir do procedimento de Benjamini Hochberg, visando a minimização das hipóteses nulas que foram rejeitadas incorretamente (BENJAMINI; YEKUTIELI, 2001).

3.6.2 Algoritmo *Dynamic Time Warping*

O algoritmo DTW permite realizar uma medida de similaridade, entre duas séries temporais, que minimizam os deslocamentos e as distorções no tempo, detectando formas semelhantes com fases diferentes (SENIN, 2008). Isso é obtido através do alinhamento entre as séries temporais quando a função de custo ou função de distância é minimizada. Em geral, o DTW pode ser utilizado em aplicações que apresentam sequências com tamanhos ou número de elementos diferentes, sequências em que os pontos de início e fim não estão alocados na mesma posição da janela e sequências de eventos de mesma características, porém com durações diferentes, como nas tarefas de reconhecimento de voz.

Supondo duas séries temporais $X = (x_1, x_2, \dots, x_N)$ e $Y = (y_1, y_2, \dots, y_M)$, o algoritmo DTW inicia a construção de uma matriz de custos $C \in \mathbb{R}^{N \times M}$, onde cada elemento da matriz representa o valor da distância entre os pares de X e Y . Esses valores de distâncias são designados como medida de custo local. Tipicamente, elementos equivalentes apresentam pequenos valores de distância. De outra forma, se os valores de distâncias forem elevados, os elementos apresentam maior diferença entre si.

O alinhamento entre as sequências X e Y é definido por um conjunto de pontos $p = (p_1, \dots, p_L)$ denominado como caminho de dobra ótimo (*optimal warping path*). Esses pontos são determinados a partir da matriz de custos e devem satisfazer algumas condições: condição de limite, condição de monotonicidade e condição de tamanho do passo. A condição de limite impõe que os pontos extremos do caminho de dobra sejam necessariamente os primeiros e os últimos elementos da sequência X e Y , respectivamente. A condição de monotonicidade preserva a ordenação temporal dos pontos. Por fim, a condição de tamanho do passo garante que o caminho de dobra seja composto por valores correspondentes às células adjacentes da matriz de custos, incluindo diagonais.

Há vários caminhos de dobra que satisfazem as condições descritas, no entanto, é de interesse determinar o caminho que apresenta o mínimo custo dentre todas as possibilidades, isto é, o caminho de dobra ótimo. Uma solução de baixa complexidade para determinar o caminho de dobra ótimo é proporcionada pelo algoritmo de programação

dinâmica. De forma primária, o algoritmo calcula uma matriz de custos acumulados $D_{N \times M}$, cujos elementos da primeira linha, primeira coluna e todos os outros elementos são definidos conforme as equações 3.1, 3.2 e 3.3, respectivamente, sendo que os elementos com índice zero são considerados nulos.

$$D(1, j) = C(1, j) + D(1, j - 1), \quad j \in [1, M] \quad (3.1)$$

$$D(i, 1) = C(i, 1) + D(i - 1, 1), \quad i \in [1, N] \quad (3.2)$$

$$D(i, j) = C(i, j) + \min \begin{cases} D(i - 1, j) \\ D(i, j - 1) \\ D(i - 1, j - 1), \end{cases} \quad i \in [1, N], \quad j \in [1, M] \quad (3.3)$$

no qual $C(i, j)$ são elementos da matriz de custo e $D(i - 1, j)$, $D(i, j - 1)$ e $D(i - 1, j - 1)$ são elementos da matriz de custos acumulados.

Uma vez que a matriz de custos acumulados é definida, o caminho de dobra ótimo pode ser encontrado pelo simples retrocesso (*backtracking*) do ponto $p_{final} = (N, M)$ para o $p_{inicial} = (1, 1)$, seguindo a estratégia de mínimo valor entre os elementos $D(i - 1, j)$, $D(i, j - 1)$ e $D(i - 1, j - 1)$, onde (i, j) são coordenadas referentes ao ponto anteriormente determinado. O algoritmo DTW retorna o valor de similaridade (distância DTW) entre as seqüências, definido pelo custo total do caminho de dobra ótimo.

A implementação do DTW é realizada utilizando a biblioteca *tslearn* implementada em Python. Por padrão da biblioteca, a função de custo para cálculo das medidas locais é a distância euclidiana quadrática, embora seja possível utilizar outros algoritmos para cálculo da distância, como é caso do Manhattan. O algoritmo DTW implementado retorna dois resultados: o conjunto de pontos do caminho de dobra ótimo e o valor da distância DTW. Esse último, é utilizado como base para formar os atributos aplicados aos algoritmos de aprendizagem, de modo que, os valores calculados são obtidos entre todas as amostras de eventos de condução disponíveis. Com os valores de distância DTW calculados entre as amostras, a média dos resultados de cada classe incide em um atributo.

De forma detalhada, supondo um evento não agressivo como referência e comparando-o às demais amostras de eventos, a média dos resultados referentes a todos os eventos não agressivos, curvas agressiva à direita e à esquerda, troca de faixa agressiva à direita e à esquerda, freada agressiva e aceleração agressiva, gera um atributo por classe ao vetor de

características referente a medida DTW.

Isto posto, o vetor de características, que implica nos dados de entrada dos modelos SVM e MLP, é composto por cinco atributos estatísticos (domínio do tempo) que são extraídos e selecionados das séries temporais utilizadas e sete atributos provenientes da medida de similaridade DTW para cada sinal utilizado.

3.7 Técnica de conversão de séries temporais em gráficos de recorrência

Uma vez que os dados dos sensores inerciais são unidimensionais, a proposta do modelo baseado em rede neural profunda CNN 2D, requer o uso de alguma estratégia para tornar as séries temporais em dados bidimensionais, possibilitando seu processamento. Para isso, é possível utilizar algumas técnicas de transformação de séries temporais em imagens, tais como: gráficos de recorrência (*Recurrence Plot*), campo angular de Gramian (*Gramian Angular Field*) e campo de transição de Markov (*Markov Transition Field*).

Neste trabalho, é utilizada a técnica de gráficos de recorrência para conversão de sinais temporais em imagens. Tecnicamente, o gráfico de recorrência é uma ferramenta que visualiza o comportamento recorrente da trajetória no espaço de fase⁷ em sistemas dinâmicos. Uma recorrência é definida quando a distância entre dois estados (pontos na trajetória do espaço de fase) é menor que um limiar preestabelecido. Matematicamente, o gráfico de recorrência, obtido pela matriz de recorrência \mathbf{R} , pode ser calculado conforme a equação 3.4.

$$r(i, j) = \begin{cases} 1, & \text{se } \|\vec{x}_i - \vec{x}_j\| \leq \varepsilon, \quad i, j = 1, \dots, N \\ 0, & \text{caso contrário} \end{cases} \quad (3.4)$$

no qual, \vec{x}_i e \vec{x}_j são os estados correspondentes aos instantes i e j ; N é o número de estados considerados; $\|\cdot\|$ é uma norma entre os estados (por exemplo, norma L_2 , L_1 ou L_∞), e ε é o limiar de distância.

O gráfico de recorrência é visualizado como uma imagem formada por uma matriz, cujos elementos da matriz correspondem às estimativas de recorrências $r(i, j)$ e as colunas

⁷O espaço de fase é composto por um sistema de coordenadas definido pela quantidade de variáveis dinâmicas que o sistema apresenta. As variáveis dinâmicas definem um vetor de estados. A dinâmica do sistema é descrita de modo que a cada instante de tempo, o vetor de estados está em uma posição diferente no espaço de fase. A evolução cronológica desses pontos definem uma trajetória.

e linhas correspondem aos pares de tempos i e j . Dessa forma, os elementos da matriz de valor 1 correspondem a um pixel preto na imagem e os elementos de valor 0 correspondem ao pixel branco.

Por outro lado, é usual a construção de uma matriz simétrica \mathbf{D} calculando as distâncias entre todos os pares de vetores como

$$d(i, j) = \|\vec{x}_i - \vec{x}_j\|, \quad i, j = 1, \dots, N. \quad (3.5)$$

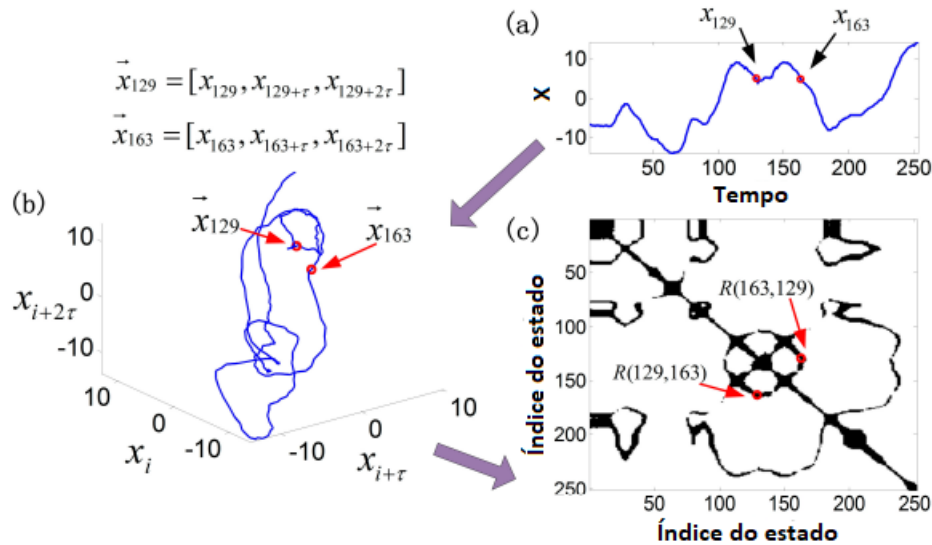
Neste caso, cada valor de distância é associado a uma cor da escala de cinza, por exemplo, cujo contraste corresponde a magnitude do valor em uma matriz bidimensional. Essa estratégia é também conhecida como gráficos de recorrência sem limiar (*unthresholded recurrence plots*) e foi utilizada neste trabalho para representar as séries temporais obtidas com os sensores inerciais.

As séries temporais podem ser representadas no espaço de fases a partir de suas observações. Isso é possível aplicando a incorporação de atraso de tempo (*delay embedding*) (TAKENS, 1981) que define a quantidade de observações e seus valores, capazes de descrever um estado da série temporal no espaço de fase. Sendo assim, dado uma série temporal (x_1, \dots, x_n) , seus estados podem ser representados no espaço de fase como sendo $\vec{x}_i = (x_i, x_{i+\tau}, \dots, x_{i+(\mu-1)\tau})$, onde μ é o tamanho da incorporação, τ é o tempo de atraso e $(x_i, x_{i+\tau}, \dots, x_{i+(\mu-1)\tau})$ são observações da série temporal. Além disso, essas observações definem os eixos do espaço de fase μ -dimensional, reconstruído.

A Figura 11 apresenta um exemplo que demonstra o método de gráfico de recorrência. Em (a), tem-se a representação da série temporal. Em (b), tem-se a representação do espaço de fase reconstruído pela incorporação de atraso de tempo, cujos parâmetros são $\mu = 3$ e $\tau = 2$. Destacam-se os estados \vec{x}_{129} e \vec{x}_{163} , descritos no espaço de fase a partir de três observações da série temporal definidas por um intervalo de tempo τ . A relação entre \vec{x}_{129} e \vec{x}_{163} , definida através da equação 3.4, considerando ε igual a 3, reflete nos pontos $R(129, 163)$ e $R(163, 129)$ apresentados no gráfico de recorrência em (c).

A implementação do gráfico de recorrência foi realizado através da biblioteca *pyts*, desenvolvida em Python, que contempla recursos para transformação de séries temporais em imagens. Essa biblioteca possibilita a construção do gráfico de recorrência de séries temporais aplicando a incorporação de atraso de tempo. Os parâmetros de entrada necessários para obtenção do gráfico de recorrência são: a dimensão da incorporação, o atraso de tempo e o limiar de distância ε . Além disso, a biblioteca faz uso da norma

Figura 11: Exemplo do método de gráfico de recorrência de séries temporais: (a) representação da série temporal, (b) espaço de fase tridimensional reconstruído e (c) gráfico de recorrência correspondente à série temporal.



Fonte: Adaptado de (ZHOU; ZHANG, 2015).

euclidiana (norma L_2). Nessa proposta, atribuiu-se um valor unitário para a dimensão de incorporação e para o atraso de tempo.

Estimar um valor adequado para o limiar de distância ε não é uma tarefa trivial (SHENDRYK et al., 2019). A literatura apresenta diversos estudos para definição dos valores de ε , alguns são baseados em regras práticas, como, por exemplo, definir um percentual sobre a distância máxima (MINDLIN; GILMORE, 1992). Outras propostas incluem a escolha de um valor baseado na densidade de pontos de recorrência (ZBILUT; ZALDIVAR-COMENGES; STROZZI, 2002) e, ainda, de um valor cinco vezes maior do que o desvio padrão do ruído observacional (THIEL et al., 2002).

Nessa proposta, devido a dificuldade de estimar um valor de limiar adequado, as informações contidas no gráfico de recorrência são valores de distâncias calculados entre os estados da série temporal, que resulta em um gráfico de recorrência não binarizado. Dessa forma, como descrito anteriormente, cada valor de distância é associada a uma cor na escala de cinza. Sendo assim, os sinais filtrados são representados por uma imagem na escala de cinza, apresentando um canal por sinal, de dimensão 252×252 pixels, para uma janela com 252 amostras.

Diferentes abordagens podem ser propostas para avaliar os padrões gerados pelos gráficos de recorrência. Eckmann, Kamphorst e Ruelle (1987) descreveu padrões de pequena e grande escala (ou seja, padrões de textura e tipologia, respectivamente) que

podem ser obtidos com os gráficos de recorrência (MARWAN et al., 2007). A textura representa estruturas de pequena escala, como linhas diagonais, pontos, linhas verticais e horizontais. Por outro lado, a tipologia caracterizada como homogênea, periódica, deriva ou interrompida representa a estrutura global.

3.8 Classificação dos eventos de direção baseada no algoritmo SVM

A partir dos dados obtidos da estratégia de engenharia de atributos, é proposto uma avaliação de desempenho de modelos baseados em SVM para o reconhecimento de comportamentos agressivos ao volante através dos eventos de direção. Para isso, diferentes combinações de hiperparâmetros do algoritmo são testados a fim de obter o melhor desempenho.

Para gerar essas combinações, os hiperparâmetros manipulados são o tipo de *kernel* e a intensidade de regularização C . Dessa forma, os *kernels* avaliados são o linear, sigmoidal, RBF e polinomial de ordem 2, 3, 4, 6 e 9. Além disso, os valores atribuídos ao parâmetro de regularização C são descritos através do conjunto $C_{reg} = \{1, 50, 100, 250, 500, 1000, 3000\}$.

Uma das adversidades encontradas no desenvolvimento é o baixo número de amostras do conjunto de dados. Isso implica que o particionamento dos dados em conjunto de treino, validação e teste reduz, consideravelmente, o número de amostras que são usadas para a aprendizagem dos modelos. Sendo assim, os resultados acabam dependendo de escolhas aleatórias dos dados e levam à modelos com capacidade de generalização inapropriada. Para não gerar esse problema, a avaliação de desempenho de cada configuração proposta é realizada através da estratégia de validação cruzada *k-fold*. Além disso, somente são realizados o particionamento dos dados em grupos de treino e validação. Os resultados finais são obtidos a partir das médias das métricas obtidas com os *k-folds* e a matriz de confusão é avaliada sobre todo o conjunto do banco de dados.

O método *k-fold* propõe a divisão do conjunto de dados em k subconjuntos mutuamente exclusivos, onde $k - 1$ subconjuntos são utilizados para treinamento e a parte restante dos dados é utilizada para validação, calculando-se uma pontuação. Esse processo é realizado em k iterações, alternando os subconjuntos utilizados como treino e validação. Ao final, a média das pontuações de cada iteração, representa a medida de desempenho relativa à validação cruzada. Em experiências anteriores, é visto que o valor de k igual a 10 é amplamente utilizado em tarefas de aprendizagem de máquina, por resultar

em uma estimativa de desempenho de baixo viés (KUHN; JOHNSON et al., 2013). Com isso, todos os modelos propostos nesse trabalho são treinados e validados com validação cruzada *10-fold*.

A implementação do SVM é realizada a partir do pacote *scikit-learn*. O conjunto de entradas é definido pelo vetor de características obtidas da engenharia de atributos e do algoritmo DTW. O treinamento é realizado com um critério de parada com tolerância de 10^{-5} e um número máximo de 1000 épocas. O desempenho dos modelos é baseado nas métricas de acurácia e *F1-score* (médias macro e micro). A partir do melhor resultado geral da validação cruzada *10-fold*, o modelo de melhor desempenho, dentre as iterações realizadas, tem sua matriz de confusão detalhada. Essa matriz de confusão é obtida inferindo todo o conjunto de dados.

3.9 Classificação dos eventos de direção baseada em uma rede MLP

Outros modelos avaliados para a tarefa de reconhecimento de comportamentos agressivos ao volante através dos eventos de direção, são baseados em rede neural MLP. Dessa forma, diferentes combinações de hiperparâmetros da rede e parâmetros de treinamento, são testados a fim de obter o melhor desempenho.

Os modelos avaliados são decorrentes da manipulação de hiperparâmetros tais como o número de camadas ocultas, o número de neurônios nas camadas ocultas e a função de ativação. Além disso, são propostos dois diferentes métodos de otimização do algoritmo de treinamento da rede. Para o número de camadas ocultas, são analisados modelos de 1 a 3 camadas com uma variação do número de neurônios, por camada, entre os valores 5, 32, 64, 128, 256, 512 e 1024 neurônios. As funções de ativação utilizadas são a ReLU e a tangente hiperbólica e os algoritmos de otimização são o gradiente descendente estocástico (*Stochastic Gradient Descent*, SGD) e a estimativa de momento adaptativo (Adam).

Os primeiros modelos a serem avaliados são os de uma única camada oculta. Nessas condições, a configuração que apresentar as melhores pontuações de desempenho, define o número de neurônios da primeira camada oculta dos modelos de duas camadas ocultas. Sendo assim, a análise do número de neurônios, para modelos com duas camadas ocultas, é realizada apenas na segunda camada oculta mantendo o número de neurônios da primeira camada oculta fixo. Da mesma forma, para os modelos com três camadas ocultas, a primeira e a segunda camadas ocultas são concebidas com o número de neurônios

decorrentes do modelo de duas camadas ocultas de melhor desempenho.

A implementação da rede MLP é realizada a partir do pacote *scikit-learn*. O conjunto de entradas, como já descrito, é definido pelo vetor de características obtidas da engenharia de atributos e do algoritmo DTW. O treinamento é realizado considerando como critério de parada uma tolerância de 10^{-5} , um número máximo de 1000 épocas e uma taxa de aprendizado de 0,005. O desempenho dos modelos é baseado nas métricas de acurácia e *F1-score* (médias macro e micro). A partir do melhor resultado geral da validação cruzada *10-fold*, o modelo de melhor desempenho, dentre as iterações realizadas, tem sua matriz de confusão detalhada. Essa matriz de confusão é obtida inferindo todo o conjunto de dados.

3.10 Classificação dos eventos de direção baseada em rede neural profunda CNN 1D e CNN 2D

O reconhecimento de comportamentos agressivos ao volante através de eventos de direção é, também, avaliado com base nas redes neurais convolucionais unidimensionais (CNNs 1D) e bidimensionais (CNNs 2D). Para isso, os modelos propostos são baseados na arquitetura AlexNet unidimensional (1D-AlexNet) e bidimensional (2D-AlexNet) (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). O modelo 1D-AlexNet tem demonstrado bom desempenho na classificação com dados de séries temporais (XIE; REN; ZHU, 2020; WU et al., 2018) e o modelo 2D-AlexNet é uma arquitetura renomada para processamento de imagens, principalmente devido à sua robustez.

A AlexNet foi o pioneiro em modelos profundos de CNN e reorientou a pesquisa em aprendizado profundo depois de vencer a competição ImageNet ILSVRC-2012. Embora tenha havido várias melhorias nos modelos de CNN profundas desde a introdução do AlexNet em 2012, a forma como a arquitetura foi cuidadosamente trabalhada, o uso de funções de ativação ReLU e a estratégia usada para evitar *overfitting* por regularização mostram que este modelo ainda tem pontencialidade para aplicações recentes.

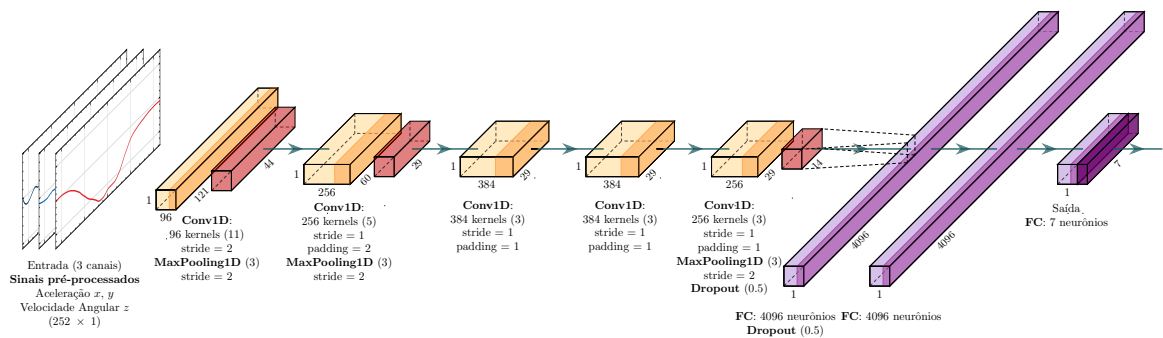
O modelo consiste em oito camadas com parâmetros ajustáveis. As cinco primeiras são as camadas convolucionais e as três últimas são as camadas *fully-connected*. Após a primeira, segunda e a quinta camadas convolucionais, tem-se as camadas de operação *Max-pooling*. Como mencionado acima, a arquitetura também é composta por três camadas FC e a saída da rede é normalizada pela função Softmax. A função de ativação utilizada nas camadas convolucionais e *fully-connected* é a ReLU. Além disso, as camadas *dropout*

são usadas antes e depois da primeira camada *fully-connected* para evitar *overfitting*.

Com base na arquitetura AlexNet, a primeira camada convolucional filtra os dados de entrada com 96 *kernels* de tamanho 11×11 e *stride* de 2×2 . A segunda camada convolucional recebe como entrada os dados reduzidos da primeira camada convolucional e aplica 256 *kernels* de tamanho 5×5 . A terceira, quarta e quinta camadas convolucionais são conectadas sem nenhuma camada de normalização ou operação de *pooling* (agrupamento). A terceira e quarta camadas convolucionais tem 384 *kernels* de tamanho 3×3 , e a quinta camada convolucional tem 256 *kernels* de tamanho 3×3 . O modelo também possui duas camadas de *dropout* com regularização de 0,5. As duas primeiras camadas *fully-connected* tem 4096 neurônios cada. Finalmente, a camada de saída tem dimensão 7, que é o número de eventos de direção rotulados. A função objetivo utilizada é a entropia cruzada.

A arquitetura 1D-AlexNet possui a mesma concepção sequencial e parâmetros descritos anteriormente. No entanto, as convoluções 2D são substituídas por operações de convolução 1D, bem como os demais parâmetros da rede. As Figuras 12 e 13 apresentam as arquiteturas de rede 1D-AlexNet e 2D-AlexNet, respectivamente, com a descrição dos parâmetros e dimensões das camadas.

Figura 12: Arquitetura 1D-AlexNet. Cada bloco descreve os hiperparâmetros atribuídos às respectivas camadas, tal como para os blocos “Conv1D” e “MaxPooling1D”.

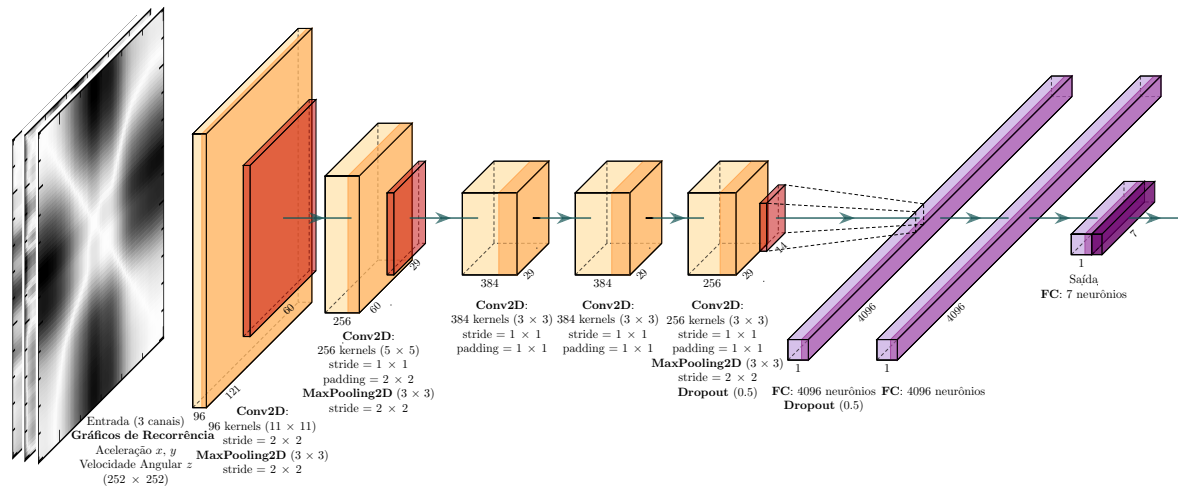


Fonte: Autor, 2022.

A implementação do modelo AlexNet é realizada a partir do biblioteca *PyTorch*. Nota-se que essa biblioteca implementa a operação de correlação cruzada nas camadas convolucionais. Na prática os resultados não se alteram, uma vez que os valores de pesos ajustáveis seriam os mesmos, porém de forma espelhada (*flipped*).

Os dados de entrada do modelo 1D-AlexNet são compostos por canais correspondentes aos sinais temporais pré-processados, cada um com 252 amostras. Da mesma forma,

Figura 13: Arquitetura 2D-AlexNet. Cada bloco indica a descrição de hiperparâmetros atribuídos às respectivas camadas, tal como para os blocos “Conv2D” e “MaxPooling2D”.



Fonte: Autor, 2022.

os dados de entrada do modelo 2D-AlexNet são compostos por canais relacionados aos gráficos de recorrência dos sinais propostos. Cada gráfico de recorrência corresponde a uma imagem de tamanho 252×252 pixels. Em ambos os casos, sinais menores do que a maior amostra de dados (ou seja, 252 amostras), foram preenchidos com a técnica de *zero-padding*.

O desempenho dos modelos é baseado nas métricas de acurácia e *F1-score* (médias macro e micro). O treinamento é realizado com 1000 épocas, algoritmo de otimização Adam e taxa de aprendizado de 0,001. A partir do melhor resultado geral da validação cruzada *10-fold*, os modelos de melhor desempenho, dentre as iterações realizadas, tem sua matriz de confusão detalhada. Essa matriz de confusão é obtida inferindo todo o conjunto de dados.

3.11 Conjunto de dados obtido experimentalmente sob condições reais

Para aumentar a quantidade de amostras para o treinamento e validação e ampliar o número de motoristas envolvidos na avaliação dos modelos, foi proposto, de forma experimental, a aquisição de novos dados de eventos de direção em condições reais. Neste experimento, os dados de cada evento são obtidos através dos sensores inerciais de um *smartphone* utilizando-se o aplicativo *phyphox*, desenvolvido pela *RWTH Aachen University* (STACKS et al., 2018), conforme apresentado no Apêndice C. Com isso, os eventos

de direção executados são rotulados conforme descrito no subitem 3.3, sendo estes, eventos não agressivos, curvas agressivas à direita e à esquerda, trocas de faixa agressivas à direita e à esquerda, freadas agressivas e acelerações agressivas.

Os experimentos foram realizados por um único motorista, a partir de três viagens com tempos aproximados de 25 minutos, 16 minutos e 27 minutos. Os eventos de direção foram executados utilizando-se um único veículo, modelo Volkswagen Fox 1.0 ano 2010, sob condições de tempo parcialmente nublado, pista seca e asfaltos regulares. O *smartphone* utilizado foi um modelo Xiaomi Redmi Note 8 Pro com sistema Android versão 10 (ESCOTTÁ; BECCARO, 2022). Durante as viagens, o *smartphone* foi fixado ao painel de instrumentos do veículo, por meio de um suporte, na orientação paisagem. A Figura 14 apresenta o sistema de coordenadas do dispositivo x , y e z , que define a direção dos dados coletados pelos sensores.

Figura 14: Local de instalação do *smartphone* e sistema de referência de coordenadas x , y e z . O eixo x aponta para a esquerda e é perpendicular ao plano de simetria do veículo. O eixo y aponta para frente e é paralelo ao plano de simetria do veículo. O eixo z se estende para baixo. As rotações em torno dos eixos x , y e z são sentido anti-horário (regra da mão direita).



Fonte: Autor, 2022.

Existem muitas definições de comportamento de agressivo ao volante. No entanto, é possível associar esse comportamento a algumas ações específicas, como excesso de velocidade, trocas de faixa repetidas e frequentes, aceleração inconsistente e excessiva e

frenagens bruscas. Assim, o experimento foi baseado em eventos que emulam os atributos definidos para direção agressiva e não agressiva, com base na experiência do motorista.

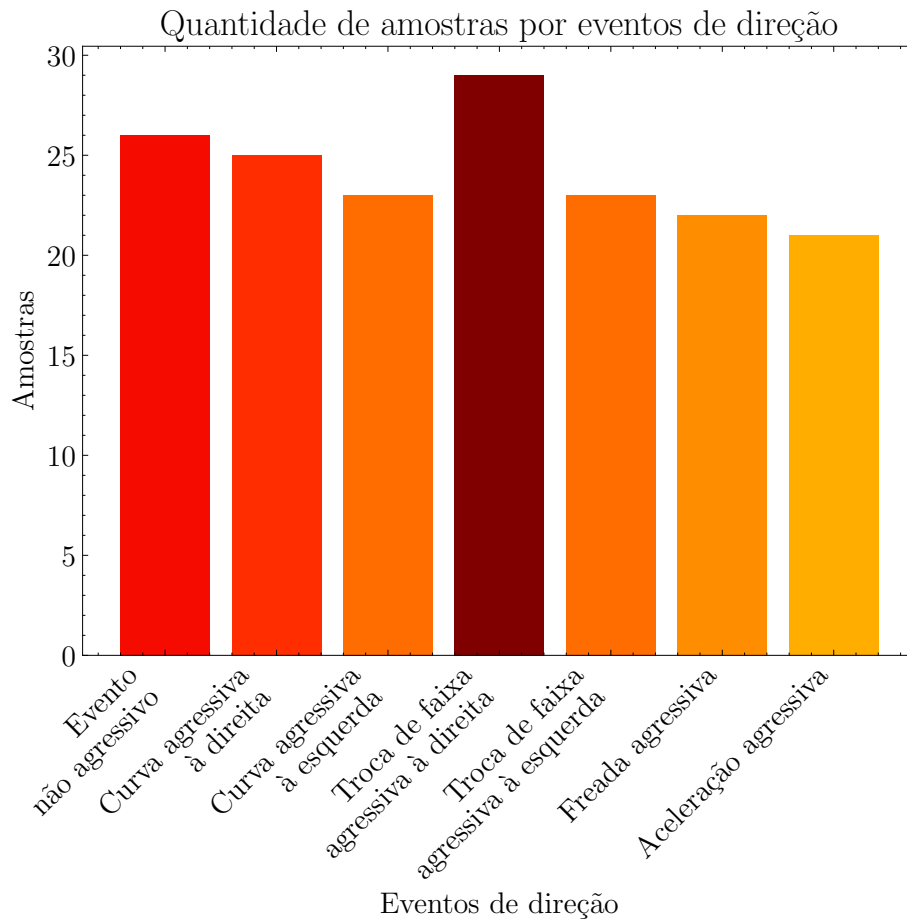
O comportamento agressivo foi emulado por eventos de direção com características de aceleração e desaceleração severas, além de esterçamentos abruptos nas mudanças de direção. Por outro lado, os eventos não agressivos replicam os eventos de aceleração, frenagem, trocas de faixa e curvas, porém, com menor intensidade e maior cautela durante os movimentos. Esses eventos foram rotulados pelo motorista, registrando o início e fim dos eventos, através do gravador de áudio do *smartphone* executado em segundo plano durante todo o período de viagem. Com esse arranjo, é possível ao ouvir o áudio gravado, marcar exatamente qual evento de direção foi realizado e quando o evento começou e terminou. Os dados de cada evento de direção apresentam informações triaxiais de aceleração linear (aceleração sem influência da gravidade) e velocidade angular.

Embora alguns eventos sejam uma combinação de duas ações de direção (por exemplo, desacelerar ao virar ou acelerar e trocar de faixa), esses eventos são classificados de acordo com a classe que mais se assemelham, no entanto, o mais importante é que os eventos sejam categorizados como agressivos e não agressivos.

Os dados dos sensores inerciais (ou seja, acelerômetro e giroscópio) foram coletados a uma taxa de 400 amostras/s, o que implica em um intervalo de tempo 2,5 ms por amostra de dados. Do experimento realizado, obteve-se um conjunto de dados com 169 eventos de direção, subdivididas em 26 eventos não agressivos, 25 eventos de curva agressiva à direita, 23 eventos de curva agressiva à esquerda, 29 eventos de troca de faixa agressiva à direita, 23 eventos de troca de faixa agressiva à esquerda, 22 eventos de freada agressiva e 21 eventos de aceleração agressiva. A Figura 15 apresenta, graficamente, o número de amostras coletadas por classe de eventos de direção pertinentes ao conjunto de dados obtido a partir do arranjo experimental proposto.

Durante o experimento, buscou-se coletar um número equilibrado de eventos para cada classe e procurou-se garantir que os eventos fossem repetíveis. Foram descartados alguns eventos devido a erros durante o experimento, como eventos que foram iniciados, mas não foram concluídos por motivos de segurança.

Figura 15: Quantidade de amostras coletadas por eventos de direção pertinentes ao conjunto de dados obtido a partir do arranjo experimental proposto.



Fonte: Autor, 2022.

3.12 Classificação dos eventos de direção, a partir dos dados unificados, baseada nos modelos em SVM, MLP, CNN 1D e CNN 2D

Nesta etapa, o desempenho dos modelos é avaliado a partir da unificação dos dados do conjunto de eventos de direção do subitem 3.3 e dos novos eventos de direção obtidos experimentalmente, subitem 3.11. Com isso, o banco de dados resultante é formado por 238 eventos de direção, permitindo uma análise em vista de diferentes motoristas, veículos e sensores.

Para realizar a unificação dos dados, é necessário compatibilizar a taxa de amostragem entre os bancos de dados. Dessa forma, foi proposto aos dois conjuntos de dados a redução da taxa de amostragem para 25 amostras/s, através da técnica de reamostragem (*downsampling*). Como estamos lidando com eventos de baixa frequência, após a aplicação de um filtro de 8ª ordem passa-baixa Chebyshev tipo I, o número de amostras por evento

de direção foram reduzidos, possibilitando a redução do tempo de treinamento da rede sem a perda de informações significativas. Além disso, os sistemas de coordenadas são diferentes entre os bancos de dados. Para manter o mesmo referencial, os dados obtidos através do experimento realizado, foram manipulados de modo a atender o sistema de coordenadas do banco de dados do Júnior e Pessin (2016).

Os parâmetros do filtro de suavização do filtro Savitzky-Golay obtidos empiricamente, são utilizados no pré-processamento dados unificados. Além disso, os atributos calculados são os mesmos atributos aplicados na avaliação dos modelos SVM e MLP, a partir dos dados do subitem 3.3. Com a redução da taxa de amostragem, os dados de entrada do modelo 1D-AlexNet são compostos por canais que apresentam sinais com 252 amostras aproximadamente 10,08 segundos o que corresponde a duração do maior evento de direção rotulado. Da mesma forma, os dados de entrada do modelo 2D-AlexNet são compostos por canais que apresentam os gráficos de recorrência correspondentes a imagens de tamanho 252×252 pixels. Em ambos os casos, sinais menores do que a maior amostra de dados (isto é, 252 amostras), foram preenchidos com a técnica de *zero-padding*.

O desempenho dos modelos é baseado nas métricas de acurácia e *F1-score* (médias macro e micro). Os treinamentos são realizados utilizando os mesmos parâmetros apresentados anteriormente para cada modelo. Os modelos são treinados e validados a partir da validação cruzada *10-fold*. Com base no melhor resultado geral, os modelos de melhor desempenho, dentre as iterações realizadas na validação cruzada, tem sua matriz de confusão detalhada. As matrizes de confusão são obtidas inferindo todo o conjunto de dados unificados.

4 RESULTADOS

Neste capítulo são apresentados os resultados de todo o desenvolvimento proposto que envolve a avaliação de desempenho de modelos de aprendizagem de máquina para o reconhecimento de comportamentos agressivos ao volante, através dos eventos de direção.

Os resultados apresentados nas seções 4.1 a 4.8 são baseados no conjunto de dados dos trabalhos de Júnior e Pessin (2016). Posteriormente, com a adição do conjunto de dados experimentalmente coletado, são avaliados, utilizando os mesmos modelos e metodologias, os dados unificados nas seções 4.9 a 4.10.

4.1 Conjunto de dados brutos dos sensores inerciais

O conjunto de dados coletados por Júnior e Pessin (2016) apresentam marca de tempo de nanosegundos e os eventos são definidos e rotulados por intervalos de tempo de início e fim. Dessa forma, os dados de tempo são convertidos em segundos, permitindo a separação e caracterização das sequências de dados de cada evento de condução.

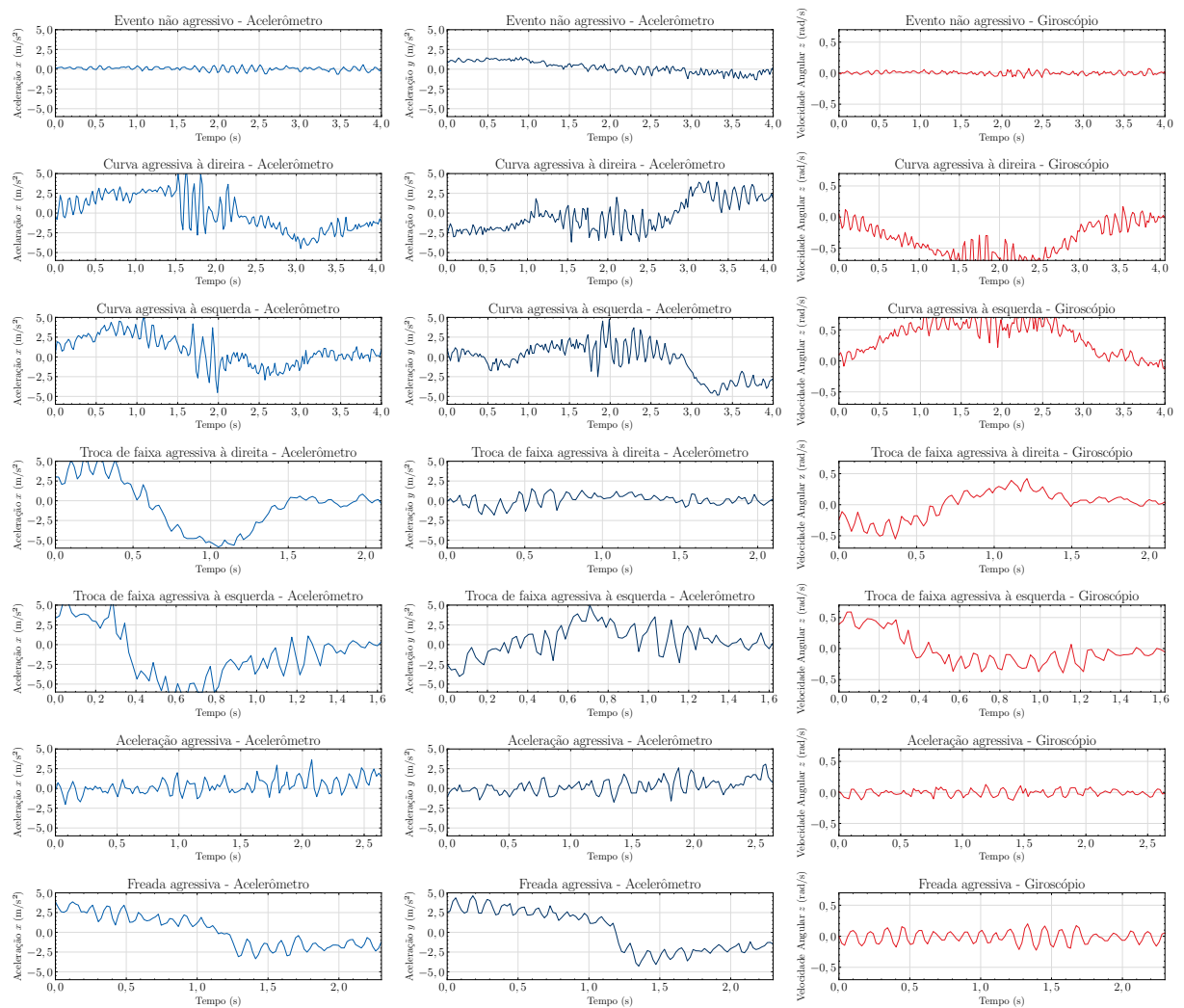
Verificou-se que os dados de aceleração linear nas direções x e y apresentam maior potencial discriminatório para a categorização dos eventos de direção, uma vez que os eventos provocam mudanças significativas ao longo desses eixos. Além disso, para inferir a dinâmica lateral do veículo, bem como diferenciar a direção de ocorrência dos eventos, podem ser utilizados os dados de velocidade angular na direção z .

Embora exista um determinado comportamento de variação dos sinais de aceleração e velocidade angular em todas as outras direções, esses sinais possuem menor capacidade de descrever o comportamento dos eventos ocorridos e são redundantes aos eixos selecionados. Isto posto, ao invés de utilizar os dados triaxiais dos sensores, foram considerados apenas os sinais de aceleração na direção x e y (acelerômetro) e velocidade angular na direção z (giroscópio). A Figura 16 apresenta um conjunto de dados brutos dos sinais de aceleração nas direções x e y e velocidade angular na direção z para cada classe de evento de direção

rotulado.

Eventos que ocorram durante a trajetória, tais como lombadas, buracos ou perturbações na pista podem ser avaliados com o sinal de aceleração na direção z . Os bancos de dados em questão não rotularam este tipo de evento que é usualmente utilizado para análise do estado das vias.

Figura 16: Dados brutos de aceleração linear na direção x (primeira coluna) e y (segunda coluna) e velocidade angular em z (terceira coluna) para eventos não agressivos (primeira linha), curva agressiva à direita e à esquerda (segunda e terceira linhas), troca de faixa agressiva à direita e à esquerda (quarta e quinta linhas), freada agressiva (sexta linha) e aceleração agressiva (sétima linha), respectivamente.



Fonte: Autor, 2022.

4.2 Conjunto de dados dos sensores inerciais pré-processados

O pré-processamento dos dados passa pela suavização dos sinais utilizando o filtro Savitzky-Golay. A implementação é realizada através da biblioteca *SciPy*. Os parâmetros da função foram determinados empiricamente, visando a combinação que proporciona melhor visualização dos padrões e tendências dos dados. Nessas condições, para todos os dados de aceleração nas direções x e y e velocidade angular na direção z , foi aplicado um polinômio de ordem 3 e um tamanho de janela de 21 pontos. Para tratar os dados nas fronteiras, o polinômio é ajustado aos próprios valores. Na implementação, a estratégia utilizada é definida pelo modo “interp”, onde nenhuma extensão é aplicada. A Figura 17 mostra os mesmos eventos apresentados na Figura 16, no entanto, suavizados pelo filtro Savitzky-Golay.

É possível identificar diferentes características entre os sinais dos eventos apresentados. Nota-se que após o pré-processamento do dados, embora haja um decréscimo em termos de amplitude dos sinais, verifica-se que o comportamento dos eventos não se perde, pelo contrário, fica ressaltado. Dessa forma, é possível identificar diferentes características entre os sinais dos eventos apresentados.

Com base nos sinais de aceleração nas direções x e y , observa-se que o evento não agressivo é caracterizado por oscilações de baixa amplitude ao longo da janela do evento. Isso ocorre porque eventos não agressivos apresentam acelerações, desacelerações e mudanças de direção suaves.

Por outro lado, eventos de freada agressiva e aceleração agressiva apresentam variações mais significativas na direção y do sinal de aceleração. Quando os freios são pressionados, qualquer coisa que não esteja rigidamente fixada (ou dentro) do veículo tende a continuar se movendo para frente. Da mesma forma, quando o carro começa a se mover para frente, qualquer coisa que não esteja rigidamente fixa sofre uma aceleração para trás.

Observa-se que os eventos de curvas agressivas e trocas de faixa agressivas apresentam variações mais significativas no eixo x do sinal de aceleração, pois é a direção substancialmente afetada por esses movimentos. O aspecto agressivo é observado pelo aumento da amplitude do sinal, uma vez que movimentos não agressivos tem quase a mesma forma com menor variação de amplitude.

Com base nos sinais de velocidade angular na direção z , observa-se que os eventos não agressivos, frenagem agressiva e aceleração agressiva não apresentam variação acentuada,

pois esses eventos não mudam de direção. Nesses eventos, os sinais são caracterizados por pequenas oscilações. Por outro lado, eventos como curvas e trocas de faixa, apresentam comportamentos distintos, uma vez que os dados de velocidade angular absorvem as mudanças na orientação do movimento durante a execução do evento.

Os eventos de troca de faixa e curva apresentam padrões específicos no eixo z do giroscópio. Assumindo que o veículo está fazendo uma troca de faixa à esquerda, o sinal no eixo z do giroscópio aumenta e depois diminui, ou seja, o volante gira duas vezes, sendo uma para a esquerda e uma para a direita para correção de direção do veículo. O padrão é o inverso para a trocas de faixa à direita. Nos eventos de curva, o volante é girado apenas para um lado, portanto, para o evento de curva à esquerda, há um aumento no valor do giroscópio seguido de um retorno a zero. O padrão é o inverso para curvas à direita.

4.3 Vetor de características e dados de entrada dos algoritmos SVM e MLP

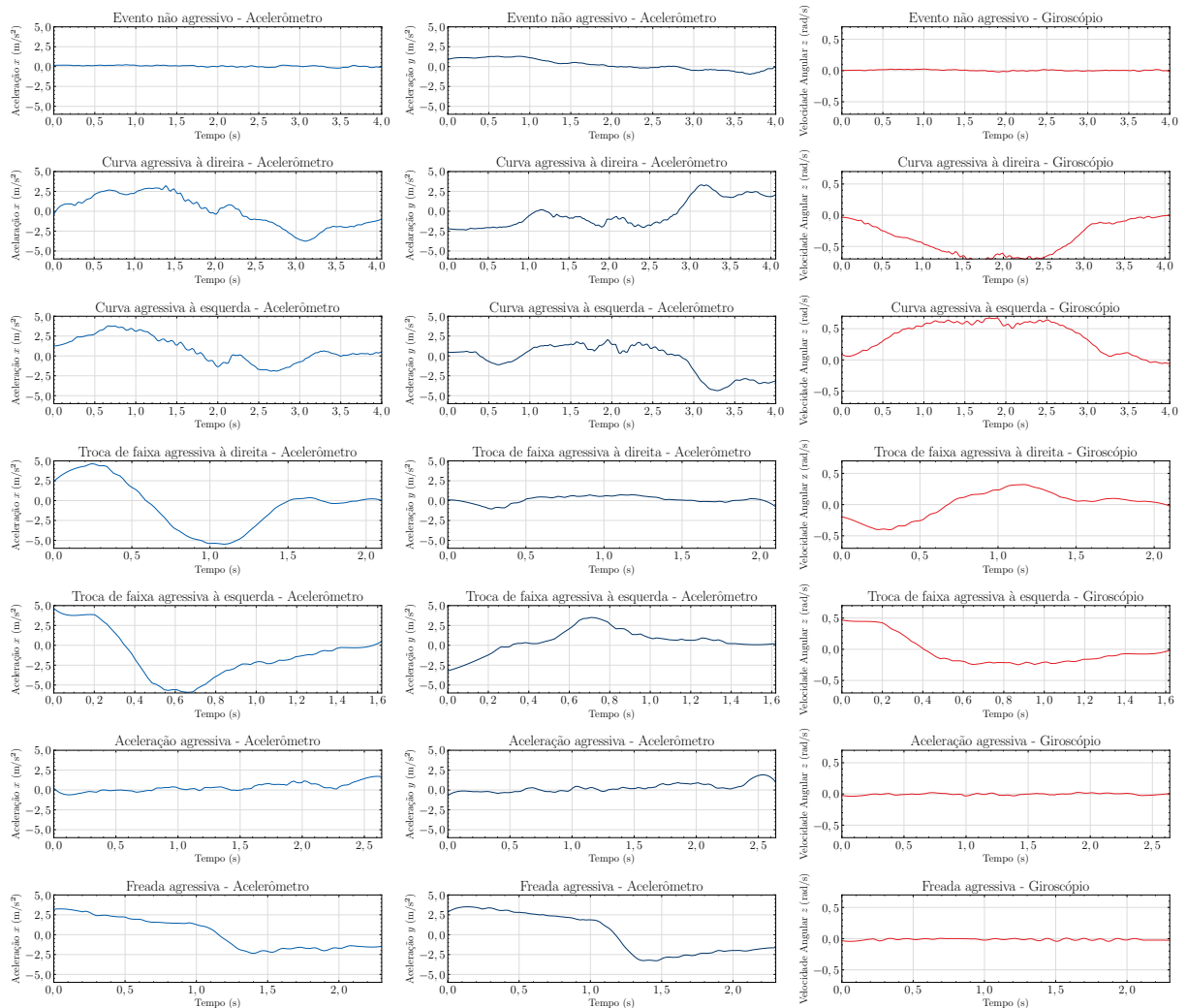
O vetor de características utilizado no desenvolvimento dos modelos em SVM e MLP apresentam atributos provenientes dos sinais pré-processados de aceleração nas direções x e y e velocidade angular na direção z .

Com relação a extração de características, a implementação utilizando o pacote *tsfresh* requer informações como os dados dos sinais filtrados (séries temporais), os rótulos dos tempos correspondentes aos dados e o índice identificador atribuído a cada sinal. Inferindo os sinais filtrados dos 69 eventos de direção (do conjunto de Júnior e Pessin (2016)), foi obtido uma matriz com 1558 atributos calculados, combinando diferentes parâmetros entre os algoritmos disponíveis¹.

Após a extração das características, o processo de seleção dos atributos mais relevantes foi realizado. A aplicação do método de filtro, através do pacote *tsfresh*, implementa a seleção de atributos que requer informações tal como a matriz dos atributos extraídos, os valores dos alvos relacionados aos sinais filtrados, o tipo de tarefa de aprendizagem de máquina, o número de testes a serem realizados e a taxa de descoberta falsa. Com isso, os alvos utilizados nesse processo são os que categorizam cada evento de direção, individualmente, sendo estes, evento não agressivo, curva agressiva à direita e à esquerda, troca de faixa agressiva à direita e à esquerda, freada agressiva e aceleração agressiva.

¹Alguns atributos retonaram valores NaN (*Not a Number*) e, portanto, foram desconsiderados, resultando em um conjunto 496 atributos que são, posteriormente, selecionados. Ao todo, a operação de extração de características demandou, aproximadamente, 1 minuto e 21 segundos para ser realizada.

Figura 17: Dados filtrados de aceleração linear na direção x (primeira coluna) e y (segunda coluna) e velocidade angular em z (terceira coluna) para eventos não agressivos (primeira linha), curva agressiva à direita e à esquerda (segunda e terceira linhas), troca de faixa agressiva à direita e à esquerda (quarta e quinta linhas), freada agressiva (sexta linha) e aceleração agressiva (sétima linha), respectivamente.



Fonte: Autor, 2022.

A tarefa de aprendizagem de máquina é definida como classificação e são considerados valores padrões do *tsfresh* para os parâmetros de número de testes e taxa de descoberta falsa, sendo estes, 1 e 5%, respectivamente. Além disso, os valores-p obtidos possibilitam a ordenação dos atributos calculados em critério de relevância.

Avaliando os valores-p calculados, os atributos que apresentaram maior relevância para os sinais de aceleração nas direções x e y e velocidade angular z foram a média, o valor máximo, o valor mínimo, o quantil e o coeficiente da regressão linear. Desta forma, justifica-se o uso destes atributos para a aplicação na entrada dos modelos de reconhecimento de padrões baseados em SVM e MLP. O detalhamento desses atributos é

apresentado no Apêndice B.

A Tabela 2 apresenta os valores-p dos atributos mais relevantes para os sinais de aceleração nas direções x e y e de velocidade angular z , bem como a descrição das características extraídas de cada sinal. Para o sinal de aceleração, os resultados mais relevantes foram obtidos com um p-quantil de ordem $p = 0,8$ e para velocidade angular com um p-quantil de ordem $p = 0,3$. A operação de regressão linear, retorna o valor médio dos interceptos obtidos pelo subconjunto de regressões realizadas a cada 10 amostras de dados dos sinais.

Tabela 2: Atributos extraídos e selecionados através do *tsfresh* e os correspondentes valores-p. Seleção de atributos relevantes através do método de filtros.

Atributos	Descrição dos Atributos	Sinais Temporais	valor-p
Valor máximo	Retorna o valor mais alto da série temporal	Aceleração em x	$3,353746 \times 10^{-6}$
		Aceleração em y	$3,478190 \times 10^{-7}$
		Velocidade angular em z	$1,781097 \times 10^{-7}$
Valor mínimo	Retorna o valor mais baixo da série temporal	Aceleração em x	$1,219650 \times 10^{-4}$
		Aceleração em y	$4,471064 \times 10^{-6}$
		Velocidade angular em z	$1,781097 \times 10^{-7}$
Média	Retorna o valor da média dos dados da série temporal	Aceleração em x	$4,207156 \times 10^{-5}$
		Aceleração em y	$1,295978 \times 10^{-4}$
Quantil	Retorna o quantil de ordem p da série temporal	Velocidade angular em z	$1,781097 \times 10^{-7}$
		Aceleração em x	$5,155762 \times 10^{-6}$
		Aceleração em y	$1,037827 \times 10^{-5}$
Coefficiente da regressão linear	Retorna o valor médio dos interceptos dos modelos de regressão linear obtidos a cada 10 amostras da série temporal	Velocidade angular em z	$1,781097 \times 10^{-7}$
		Aceleração em x	$2,43298 \times 10^{-5}$
		Aceleração em y	$2,336009 \times 10^{-5}$
		Velocidade angular em z	$1,781097 \times 10^{-7}$

Fonte: Autor, 2022.

4.3.1 Medida de similaridade entre os sinais pré-processados com base no algoritmo DTW

Para finalizar a composição do vetor de características, são obtidos os atributos relacionados ao algoritmo DTW que retorna o valor de similaridade entre os sinais pré-processados. A estratégia utilizada para obtenção dos valores propõe a análise um contra todos, onde cada amostra de evento é comparada com todas as outras amostras e, inclusive, com ela mesma. Como descrito anteriormente, estas amostras estão rotuladas. A média correspondente aos resultados de cada conjunto de evento dão origem aos atributos DTW aplicados. O objetivo desta média é diminuir o efeito de possíveis *outliers* ou de amostras que estejam mal rotuladas, sem que haja descarte de dados. Além disso, a média permite uma agregação de resultados.

A implementação do algoritmo é realizada através do pacote *tslearn* que necessita, essencialmente, das séries temporais a serem analisadas. Para este propósito, o pacote utiliza a distância euclidiana como função custo e determina o caminho de dobra ótimo através do algoritmo de programação dinâmica. A aplicação do algoritmo DTW é realizada para sinais pré-processados de aceleração nas direções x e y e velocidade angular na direção z .

Para exemplificar os resultados da implementação, a Tabela 3 apresenta os valores de medida DTW entre os sinais de aceleração na direção x de um evento não agressivo em comparação com todos os demais eventos do conjunto de dados, incluindo os não agressivos. Dessa forma, é possível evidenciar a similaridade entre os sinais de forma quantitativa, observando que para o conjunto de eventos não agressivos, os valores DTW são pequenos, indicando maior correspondência entre os sinais. Ao contrário, para os conjuntos de eventos que apresentam ações agressivas, os valores de DTW são maiores, indicando maior diferença entre os sinais.

A Figura 18 apresenta os alinhamentos DTW, ponto a ponto, entre o sinal de referência do evento não agressivo e uma amostra de cada evento rotulado. Ressalta-se que eventos de aceleração agressiva apresentam, com maior frequência, algumas amostras, cujos resultados indicam uma condição de similaridade com o sinal comparado. Para visualizar essa condição, as representações localizadas na última linha da Figura 18, apresentam condições de similaridade (medida DTW igual a 1,460) e não similaridade (medida DTW igual a 10,115) entre os eventos.

É possível observar que sinais semelhantes, apresentam um padrão de alinhamento onde os dados do sinal comparado correspondem a um ou poucos dados do sinal de

referência. Por outro lado, sinais diferentes, apresentam um padrão de alinhamento onde os dados do sinal comparado correspondem a vários dados do sinal de referência ou vice-versa.

Tabela 3: Valores de medida DTW entre os sinais de aceleração na direção x de um evento não agressivo e todos os demais eventos do conjunto de dados.

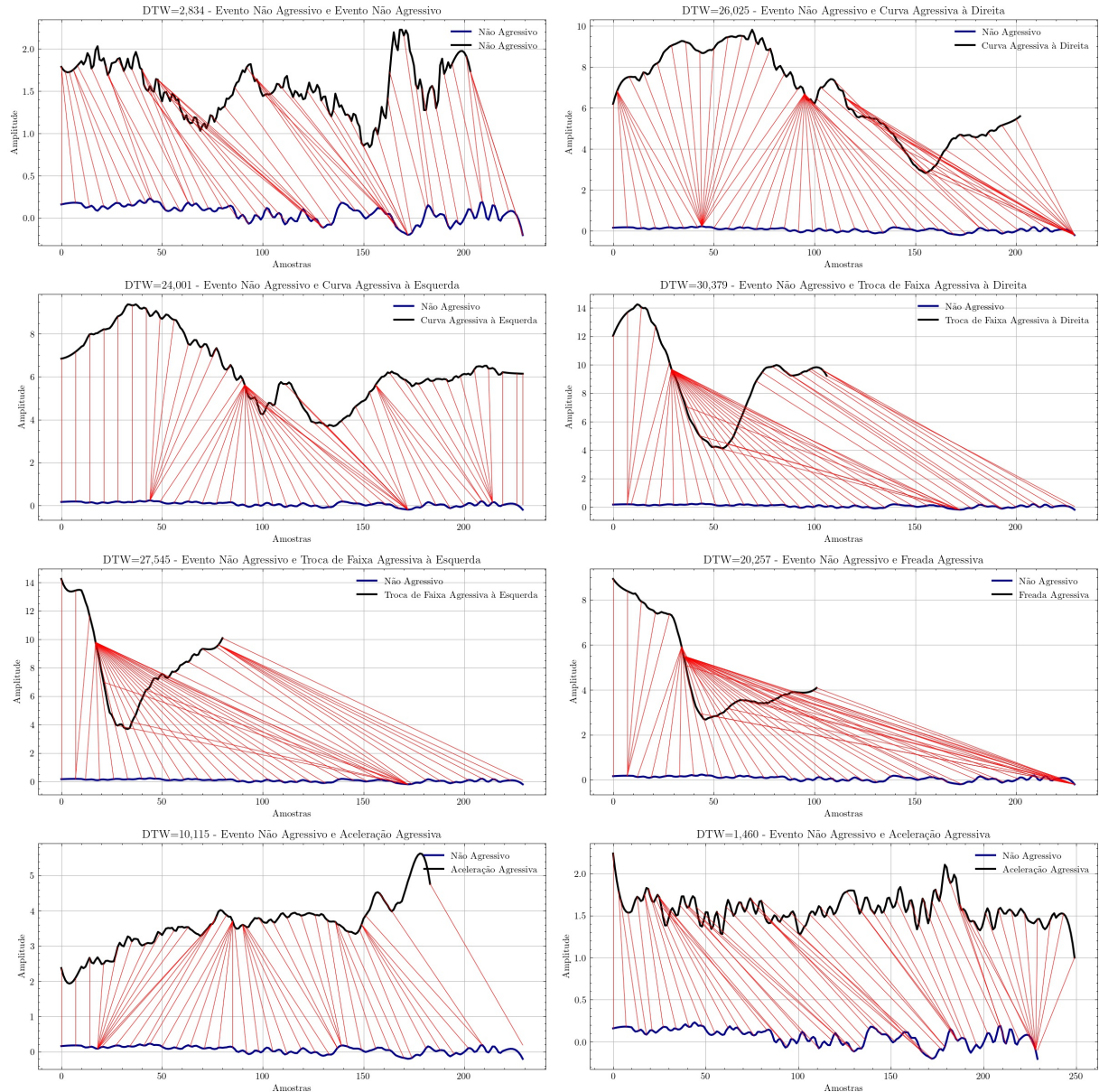
Eventos não agressivos	Curva agressiva à direita	Freada agressiva	Aceleração agressiva	Curva agressiva à esquerda	Troca de faixa agressiva à direita	Troca de faixa agressiva à esquerda
5,374	29,814	31,567	13,516	30,425	18,212	27,546
10,655	17,960	30,884	5,113	12,964	23,613	27,973
2,806	30,012	25,027	16,217	23,761	20,458	36,290
2,178	30,360	26,310	2,287	28,158	35,807	33,547
8,076	25,330	15,788	15,871	32,776	30,379	
13,583	32,181	20,257	13,347	24,001		
11,756	26,025	19,140	6,476	25,480		
2,834	25,698	17,217	10,115	26,116		
7,963	26,527	5,315	1,767	22,077		
6,823	29,299	4,750	1,460	26,272		
3,845	27,237	2,776	1,621	13,501		
0,0		18,787	8,025			
3,854						
6,840						
Média dos conjuntos						
6,141	27,312	18,151	7,992	24,139	25,693	31,338

Fonte: Autor, 2022.

Com os valores DTW calculados e, posteriormente, a média DTW de cada conjunto, o vetor de características é definido, para os sinais de aceleração nas direções x e y e velocidade angular na direção z , com os seguintes atributos: valor máximo, valor mínimo, média, p-quantil, intercepto da regressão linear e a média dos valores DTW relacionados aos eventos não agressivos, curva agressiva à direita e à esquerda, troca de faixa agressiva à direita e à esquerda, freada agressiva e aceleração agressiva. Dessa forma, o vetor de características é composto de 36 atributos que estão subdivididos entre 12 atributos de cada sinal utilizado.

A Figura 19 apresenta um gráfico bidimensional que visualiza, a partir da redução de dimensionalidade utilizando o algoritmo t-SNE (*t-distributed stochastic neighbor embedding*), o agrupamento dos atributos extraídos com relação às classes de eventos. Nota-se visível agrupamento de cada uma das classes agressivas e distribuição dos eventos não agressivos entre as diferentes classes/eventos.

Figura 18: Alinhamento DTW entre os sinais de aceleração na direção x do evento não agressivo de referência e uma amostra de cada classe de evento rotulado. O gráfico da última linha e segunda coluna, apresenta um condição de alinhamento DTW entre o evento não agressivo e aceleração agressiva indicando correspondência entre os sinais.

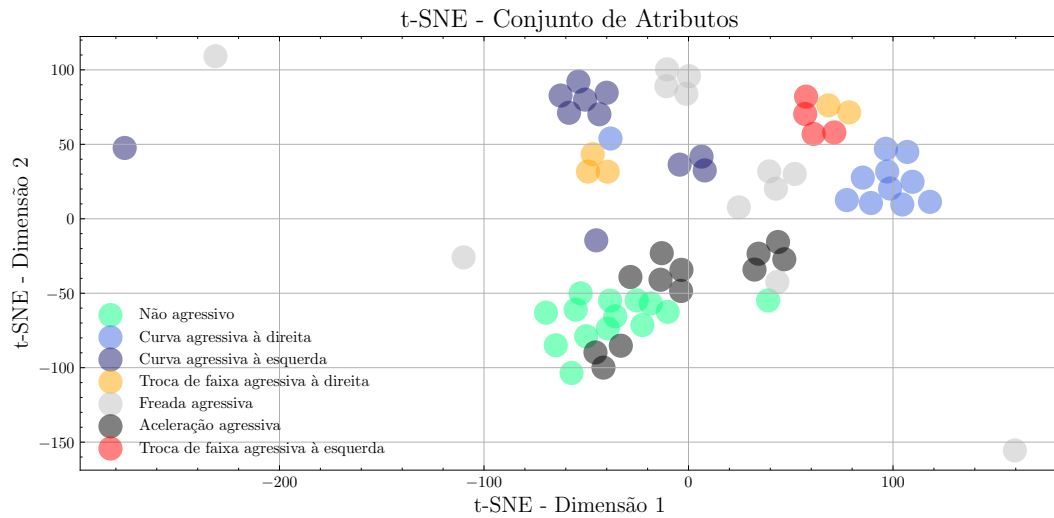


Fonte: Autor, 2022.

4.4 Análise dos sinais pré-processados através da técnica de gráficos de recorrência

A conversão dos sinais temporais em gráfico de recorrência é realizada utilizando os recursos da biblioteca *pyts* (Python). Sendo assim, os sinais de aceleração nas direções x e y e velocidade angular na direção z , de cada evento de direção, são transformados em imagens em tons de cinza, apresentando um canal e dimensão 252×252 pixels.

Figura 19: Conjunto de atributos em um espaço bidimensional obtido através do método t-SNE (perplexidade igual a 12,7).



Fonte: Autor, 2022.

Neste caso, não há necessidade de estimar um valor de limiar ε , conforme detalhado no subitem 3.7. A representação do gráfico de recorrência contém valores de distância euclidiana (norma L_2) calculados entre cada estado da série temporal, sendo que cada valor é associado a uma tonalidade na escala de cinza. As imagens resultantes são utilizadas como dados de entrada para o modelo 2D-AlexNet.

A Figura 20 apresenta os gráficos de recorrência correspondentes aos sinais pré-processados de aceleração linear nas direções x e y , e velocidade angular no eixo z . Em geral, verifica-se padrões repetitivos das recorrências para todas as classes equivalentes, demonstrando que os gráficos de recorrência são boas representações dos sinais de aceleração e velocidade angular.

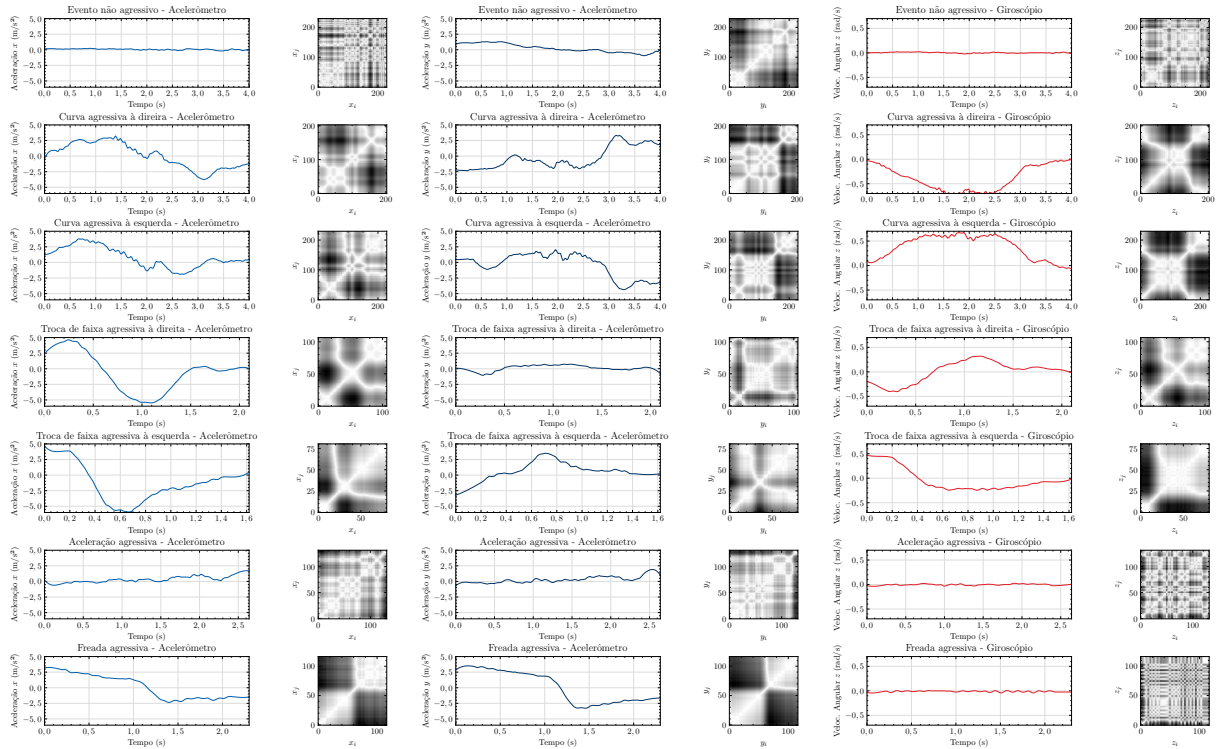
A partir dos resultados exibidos através dos gráficos de recorrência, alguns padrões característicos de sua estrutura podem ser observados. Um dos aspectos típicos está relacionado às mudanças abruptas na dinâmica do sistema, que produzem áreas ou faixas brancas nos gráficos de recorrência e são observadas nas imagens de eventos agressivos. Além disso, os aspectos visuais dos gráficos de recorrência podem ser descritos a partir de estruturas típicas, tais como, padrões homogêneos, periódicos, derivas ou interrompidos.

Para eventos agressivos que apresentam maior variação na amplitude da aceleração e velocidade angular, a estrutura do gráfico de recorrência apresenta aspectos visuais com estrutura interrompida. Essas características são mais evidentes para os sinais de aceleração em x e y e velocidade angular em z para eventos agressivos de curva e troca de faixa, além do sinal de aceleração em y para eventos de freada agressiva e aceleração

agressiva.

Por outro lado, o aspecto observado nos sinais de eventos não agressivos e eventos que apresentam baixa dinâmica apresentam características das estruturas homogênea e deriva. Essas características são observadas, com maior ênfase, nos sinais de aceleração linear em x e velocidade angular em z dos eventos de freada agressiva e aceleração agressiva.

Figura 20: Séries temporais e gráficos de recorrência (no lado direito de cada evento de condução) de sinais de aceleração linear nas direções x e y , primeira e segunda colunas e velocidade angular na direção z , terceira coluna.



Fonte: Autor, 2022.

4.5 Desempenho dos classificadores baseado no algoritmo SVM

A avaliação de desempenho dos classificadores baseados em SVM, foi realizada propondo diferentes combinações de hiperparâmetros, tal como o tipo de *kernel* e a intensidade de regularização C . Os modelos propostos são treinados e validados a partir da estratégia de validação cruzada k -fold com parâmetro de k igual a 10.

O desempenho individual para cada modelo é realizado através do cálculo de acurácia e $F1$ -score (médias macro e micro), cujos valores podem ser visualizados na Tabela 4. O melhor desempenho geral é obtido pelo classificador SVM com *kernel* RBF e parâmetro

de regularização $C = 250$, cujos os valores de acurácia e $F1$ -score macro e micro são de 87,14%, 82,29% e 87,14%, respectivamente.

Tabela 4: Medida de desempenho através das métricas de acurácia e $F1$ -score (médias macro e micro) dos diferentes modelos baseados em SVM. Valores médios obtidos através de 10-*fold*.

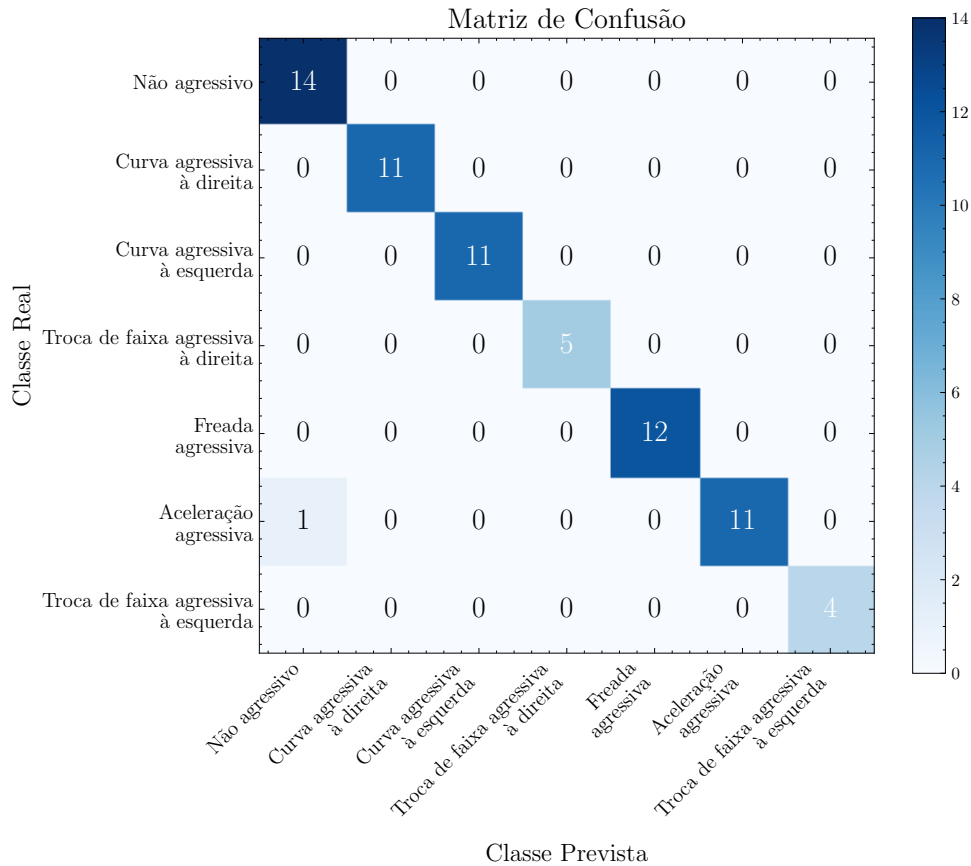
Métricas	Parâmetro C	Linear	Polinômio 2ª ordem	Polinômio 3ª ordem	Polinômio 4ª ordem	Polinômio 6ª ordem	Polinômio 9ª ordem	Sigmoidal	RBF
Acurácia	1	85,71%	66,90%	72,62%	81,19%	79,76%	72,62%	14,52%	53,81%
$F1$ -score macro	1	81,05%	58,44%	60,04%	68,89%	68,36%	63,10%	6,37%	40,64%
$F1$ -score micro	1	85,71%	66,90%	72,62%	81,19%	79,76%	72,62%	14,52%	53,81%
Acurácia	50	85,71%	85,71%	87,14%	84,29%	77,14%	71,19%	44,76%	87,14%
$F1$ -score macro	50	81,05%	77,22%	79,39%	76,79%	69,34%	57,71%	40,23%	80,70%
$F1$ -score micro	50	85,71%	85,71%	87,14%	84,29%	77,14%	71,19%	44,76%	87,14%
Acurácia	100	85,71%	85,71%	85,71%	84,29%	75,71%	71,19%	40,24%	82,86%
$F1$ -score macro	100	81,05%	77,16%	78,10%	76,79%	68,97%	57,71%	35,81%	76,83%
$F1$ -score micro	100	85,71%	85,71%	85,71%	84,29%	75,71%	71,19%	40,24%	82,86%
Acurácia	250	85,71%	87,14%	84,29%	82,86%	75,71%	71,19%	37,62%	87,14%
$F1$ -score macro	250	81,05%	79,39%	77,72%	76,41%	68,97%	57,71%	33,31%	82,29%
$F1$ -score micro	250	85,71%	87,14%	84,29%	82,86%	75,71%	71,19%	37,62%	87,14%
Acurácia	500	85,71%	84,29%	84,29%	82,86%	75,71%	71,19%	37,62%	84,29%
$F1$ -score macro	500	81,05%	77,72%	77,72%	76,41%	68,97%	57,71%	33,31%	78,96%
$F1$ -score micro	500	85,71%	84,29%	84,29%	82,86%	75,71%	71,19%	37,62%	84,29%
Acurácia	1000	85,71%	84,29%	84,29%	82,86%	75,71%	71,19%	34,76%	84,29%
$F1$ -score macro	1000	81,05%	77,72%	77,72%	76,41%	68,97%	57,71%	30,73%	78,96%
$F1$ -score micro	1000	85,71%	84,29%	84,29%	82,86%	75,71%	71,19%	34,76%	84,29%
Acurácia	3000	85,71%	84,29%	84,29%	82,86%	75,71%	71,19%	31,67%	84,29%
$F1$ -score macro	3000	81,05%	77,72%	77,72%	76,41%	68,97%	57,71%	27,49%	78,96%
$F1$ -score micro	3000	85,71%	84,29%	84,29%	82,86%	75,71%	71,19%	31,67%	84,29%

Fonte: Autor, 2022.

Com base no modelo de melhor desempenho durante a validação cruzada, a Figura 21 apresenta sua matriz de confusão obtida com a avaliação de todo o conjunto de dados, visualizando a relação entre os valores previstos e os valores reais de cada classe de eventos. Além disso, é possível observar os principais erros obtidos pelo classificador.

A partir da Figura 21, observa-se que eventos não agressivos, curva agressiva à direita e à esquerda, troca de faixa agressiva à direita e à esquerda e freada agressiva apresentam uma taxa de verdadeiro positivo (*True Positive Rate*, TPR) de 100%. O modelo obteve uma previsão incorreta referente ao reconhecimento de aceleração agressiva como evento não agressivo. Um dos possíveis motivos para o erro é devido ao mal rótulo dos eventos ou pela proximidade entre os critérios de agressividade e não agressividade reconhecidos pelo modelo.

Figura 21: Matriz de confusão do modelo SVM com *kernel* RBF e parâmetro C igual a 250 (modelo treinado e validado com validação cruzada 10-*fold*). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados.



Fonte: Autor, 2022.

4.6 Desempenho dos classificadores baseado na rede neural MLP

Diferentes combinações de hiperparâmetros da rede MLP, tal como o número de camadas ocultas, o número de neurônios nas camadas ocultas e a função de ativação, compõe diferentes modelos que são avaliados no reconhecimento dos eventos de direção. Os modelos propostos são treinados e validados a partir da estratégia de validação cruzada k -*fold* com parâmetro de k igual a 10. Além disso, é proposto o uso dos otimizadores Adam e SGD para o treinamento dos modelos.

A avaliação de desempenho dos modelos é realizada através do cálculo de acurácia e $F1$ -*score* (médias macro e micro). Os primeiros modelos a serem treinados e validados são compostos de uma única camada oculta. As medidas de desempenho obtidas para esses modelos são apresentadas na Tabela 5.

A partir dessas medidas, observa-se que o melhor desempenho para cada configuração

proposta, é alcançado por modelos com número de neurônios N_1 correspondentes a 64, 32, 1024 e 1024. Esses valores de neurônios são utilizados como valores fixos na primeira camada oculta dos modelos de duas camadas ocultas. Com isso, os modelos de duas camadas ocultas são identificados pela notação $(64, N_2)$, por exemplo, indicando que a primeira camada oculta da rede tem 64 neurônios e a segunda camada tem um número de neurônios N_2 avaliável. As medidas de desempenho obtidas para os modelos de duas camadas ocultas são apresentadas na Tabela 6.

Da mesma maneira, os modelos de três camadas ocultas contém, nas duas primeiras camadas, um número fixo de neurônios que são provenientes dos modelos com o melhor desempenho na avaliação apresentada na Tabela 6. Dessa forma, os modelos com três camadas ocultas são identificados pela notação $(64, 64, N_3)$, por exemplo, indicando que a primeira camada oculta da rede tem 64 neurônios, a segunda camada oculta tem 64 neurônios e a terceira camada oculta tem um número de neurônios N_3 avaliável. Os resultados de desempenho dos modelos com três camadas ocultas podem ser visualizados na Tabela 7.

A partir dos resultados apresentados nas Tabelas 5, 6 e 7, observa-se que o melhor desempenho geral é obtido pelo modelo MLP de duas camadas ocultas com 32 neurônios por camada e função de ativação tangente hiperbólica. Além disso, o desempenho é alcançado utilizando o otimizador Adam. Os valores obtidos de acurácia e *F1-score* macro e micro são 94,29%, 88,00% e 94,29%, respectivamente. Com base no modelo de melhor desempenho durante a validação cruzada, sua matriz de confusão é detalhada na Figura 22.

A partir da matriz de confusão, observa-se que eventos de curva agressiva à direita e à esquerda, troca de faixa agressiva à direita e à esquerda, freada agressiva e aceleração agressiva apresentam uma TPR de 100%. O modelo obteve uma previsão incorreta referente ao reconhecimento de evento não agressivo como evento aceleração agressiva. Neste sentido, os mesmos comentários feitos anteriormente valem em relação à incapacidade do modelo em discriminar corretamente eventos não agressivos em relação a agressivos.

4.7 Desempenho do classificador baseado em rede neural profunda CNN 1D, estrutura 1D-AlexNet

O modelo de classificação baseado em rede neural profunda CNN 1D, utiliza a estrutura 1D-AlexNet para o reconhecimento dos eventos de direção. O modelo é treinado e

Tabela 5: Medida de desempenho através das métricas de acurácia e *F1-score* (médias macro e micro) dos diferentes modelos baseados na rede MLP de uma única camada oculta. Valores médios obtidos através de *10-fold*.

Métricas	Neurônios (N_1)	(N_1) (Adam, ReLU)	(N_1) (Adam, Tanh)	(N_1) (SGD, ReLU)	(N_1) (SGD, Tanh)
Acurácia	5	48,33%	75,48%	18,81%	15,95%
<i>F1-score</i> macro	5	38,62%	68,06%	6,94%	9,10%
<i>F1-score</i> micro	5	48,33%	75,48%	18,81%	15,95%
Acurácia	32	87,14%	91,42%	85,71%	84,05%
<i>F1-score</i> macro	32	79,97%	85,79%	78,03%	73,49%
<i>F1-score</i> micro	32	87,14%	91,42%	85,71%	84,05%
Acurácia	64	88,57%	87,14%	82,85%	81,19%
<i>F1-score</i> macro	64	82,70%	80,90%	76,53%	68,47%
<i>F1-score</i> micro	64	88,57%	87,14%	82,85%	81,19%
Acurácia	128	87,14%	90,00%	85,71%	82,62%
<i>F1-score</i> macro	128	80,61%	84,60%	81,17%	69,31%
<i>F1-score</i> micro	128	87,14%	90,00%	85,71%	82,62%
Acurácia	256	88,57%	88,57%	84,29%	85,71%
<i>F1-score</i> macro	256	82,51%	81,35%	79,42%	78,08%
<i>F1-score</i> micro	256	88,57%	88,57%	84,29%	85,71%
Acurácia	512	85,71%	90,00%	90,00%	90,00%
<i>F1-score</i> macro	512	80,03%	82,57%	85,12%	83,30%
<i>F1-score</i> micro	512	85,71%	90,00%	90,00%	90,00%
Acurácia	1024	82,86%	91,43%	88,57%	90,00%
<i>F1-score</i> macro	1024	78,23%	85,47%	83,92%	83,30%
<i>F1-score</i> micro	1024	82,86%	91,43%	88,57%	90,00%

Fonte: Autor, 2022.

validado a partir da estratégia de validação cruzada *k-fold* com parâmetro *k* igual a 10. Além disso, é proposto o uso do otimizador Adam para o treinamento do modelo.

O desempenho do modelo é realizado através do cálculo de acurácia e *F1-score* macro e micro, cujos resultados alcançados são de 81,43%, 76,49% e 81,43%, respectivamente. A Tabela 8 apresenta as medidas de desempenho do modelo 1D-AlexNet. Com base no modelo de melhor desempenho durante a validação cruzada, sua matriz de confusão é detalhada na Figura 23.

A partir da matriz de confusão, observa-se que eventos não agressivos, curva agressiva à direita e à esquerda, troca de faixa agressiva à direita, freada agressiva e aceleração agressiva apresentam uma TPR de 100%. O modelo obteve uma previsão incorreta referente ao reconhecimento de troca de faixa agressiva à esquerda como troca de faixa agressiva à direita.

Tabela 6: Medida de desempenho através das métricas de acurácia e $F1$ -score (médias macro e micro) dos diferentes modelos baseados na rede MLP de duas camadas ocultas. Valores médios obtidos através de 10-*fold*.

Métricas	Neurônios (N_2)	(64, N_2) (Adam, ReLU)	(32, N_2) (Adam, Tanh)	(1024, N_2) (SGD, ReLU)	(1024, N_2) (SGD, Tanh)
Acurácia	5	17,14%	88,57%	13,10%	88,57%
$F1$ -score macro	5	5,90%	84,03%	6,03%	84,25%
$F1$ -score micro	5	17,14%	88,57%	13,10%	88,57%
Acurácia	32	85,48%	94,29%	87,14%	87,14%
$F1$ -score macro	32	76,58%	88,00%	87,14%	87,14%
$F1$ -score micro	32	85,48%	94,29%	87,14%	87,14%
Acurácia	64	85,71%	87,14%	87,14%	87,14%
$F1$ -score macro	64	81,99%	80,87%	82,59%	79,97%
$F1$ -score micro	64	85,71%	87,14%	87,14%	87,14%
Acurácia	128	85,71%	91,43%	87,14%	87,14%
$F1$ -score macro	128	81,09%	84,63%	81,57%	81,92%
$F1$ -score micro	128	85,71%	91,43%	87,14%	87,14%
Acurácia	256	84,29%	92,86%	87,14%	88,57%
$F1$ -score macro	256	76,85%	85,78%	82,59%	83,98%
$F1$ -score micro	256	84,29%	92,86%	87,14%	88,57%
Acurácia	512	85,71%	90,00%	90,00%	90,00%
$F1$ -score macro	512	80,03%	82,57%	85,12%	83,30%
$F1$ -score micro	512	85,71%	90,00%	90,00%	90,00%
Acurácia	1024	82,85%	91,43%	88,57%	90,00%
$F1$ -score macro	1024	78,22%	85,47%	83,92%	83,30%
$F1$ -score micro	1024	82,85%	91,43%	88,57%	90,00%

Fonte: Autor, 2022.

4.8 Desempenho do classificador baseado em rede neural profunda CNN 2D, estrutura 2D-AlexNet

O modelo baseado em rede neural profunda CNN 2D, utilizando da arquitetura 2D-AlexNet, é treinado e validado a partir da estratégia de validação cruzada k -*fold* com parâmetro k igual a 10. O treinamento é realizado utilizando-se o otimizador Adam.

O desempenho do modelo 2D-AlexNet também é avaliado pelo cálculo de acurácia, $F1$ -score macro e micro, cujos resultados alcançados são de 75,24%, 64,06% e 75,24%, respectivamente. A Tabela 9 apresenta um resumo dos resultados de desempenho do modelo 2D-AlexNet. Com base no modelo de melhor desempenho durante a validação cruzada, sua matriz de confusão é detalhada na Figura 24.

A partir da matriz de confusão, observa-se que eventos de curva agressiva à direita e à esquerda, troca de faixa agressiva à direita e à esquerda, freada agressiva e aceleração agressiva apresentam uma TPR de 100%. O modelo obteve uma previsão incorreta refe-

Tabela 7: Medida de desempenho através das métricas de acurácia e $F1$ -score (médias macro e micro) dos diferentes modelos baseados na rede MLP de três camadas ocultas. Valores médios obtidos através de 10-fold.

Métricas	Neurônios (N_3)	(64, 64, N_3) (Adam, ReLU)	(32, 32, N_3) (Adam, Tanh)	(1024, 512, N_3) (SGD, ReLU)	(1024, 512, N_3) (SGD, Tanh)
Acurácia	5	78,33%	88,57%	24,54%	85,71%
$F1$ -score macro	5	71,62%	83,59%	21,21%	80,83%
$F1$ -score micro	5	78,33%	88,57%	24,54%	85,71%
Acurácia	32	84,29%	90,00%	88,57%	87,14%
$F1$ -score macro	32	78,10%	81,46%	82,40%	83,54%
$F1$ -score micro	32	84,29%	90,00%	88,57%	87,14%
Acurácia	64	87,14%	88,57%	87,14%	87,14%
$F1$ -score macro	64	82,46%	82,01%	81,92%	80,77%
$F1$ -score micro	64	87,14%	88,57%	87,14%	87,14%
Acurácia	128	85,71%	85,71%	88,57%	87,14%
$F1$ -score macro	128	80,37%	83,21%	84,69%	81,43%
$F1$ -score micro	128	85,71%	85,71%	88,57%	87,14%
Acurácia	256	87,14%	88,57%	90,00%	88,57%
$F1$ -score macro	256	80,50%	79,19%	85,12%	81,97%
$F1$ -score micro	256	87,14%	88,57%	90,00%	88,57%
Acurácia	512	85,71%	87,14%	87,14%	85,71%
$F1$ -score macro	512	80,46%	81,76%	81,79%	80,59%
$F1$ -score micro	512	85,71%	87,14%	87,14%	85,71%
Acurácia	1024	88,57%	91,43%	87,14%	85,71%
$F1$ -score macro	1024	83,79%	84,32%	81,79%	80,59%
$F1$ -score micro	1024	88,57%	91,43%	87,14%	85,71%

Fonte: Autor, 2022.

Tabela 8: Medida de desempenho através das métricas de acurácia e $F1$ -score (médias macro e micro) do modelo 1D-AlexNet. Valores médios obtidos através de 10-fold.

Modelo	Otimizador	Acurácia	$F1$ -score macro	$F1$ -score micro
1D-AlexNet	Adam	81,43%	76,49%	81,43%

Fonte: Autor, 2022.

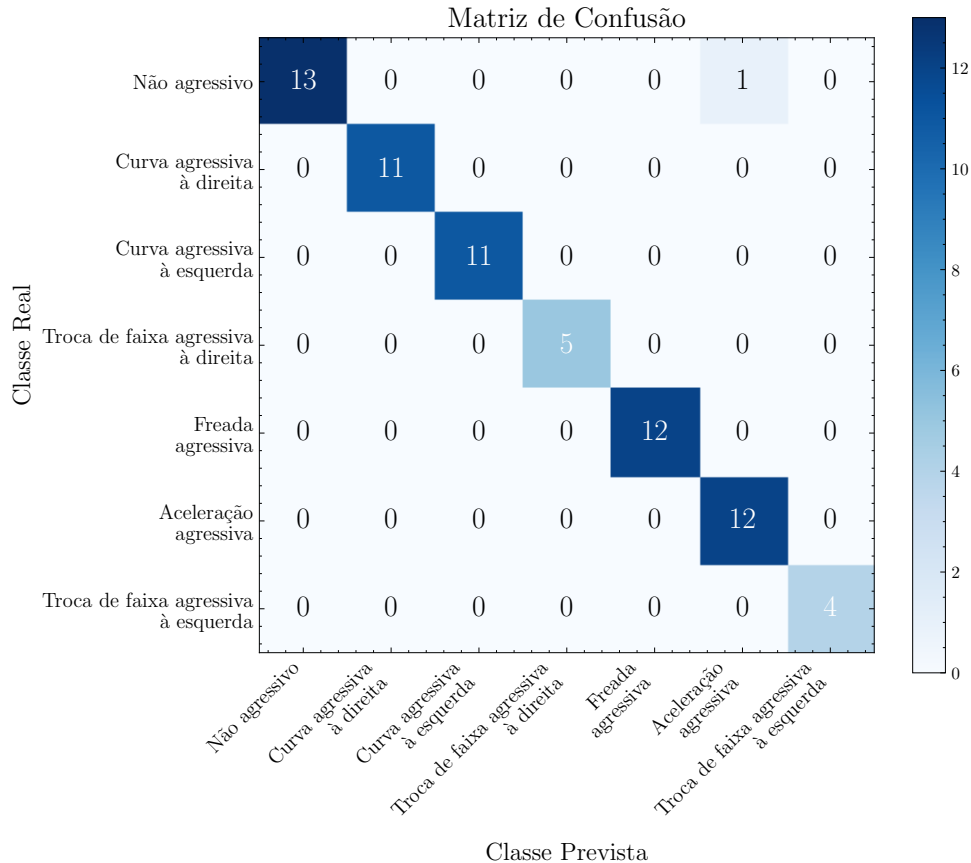
Tabela 9: Medida de desempenho através das métricas de acurácia e $F1$ -score (médias macro e micro) do modelo 2D-AlexNet. Valores médios obtidos através de 10-fold.

Modelo	Otimizador	Acurácia	$F1$ -score macro	$F1$ -score micro
2D-AlexNet	Adam	75,24%	64,06%	75,24%

Fonte: Autor, 2022.

rente ao reconhecimento de eventos não agressivo como troca de faixa agressiva à esquerda. O erro obtido pelo modelo 2D-AlexNet pode estar associado a um evento de mesma natureza executado sob condições de agressividade e não agressividade, neste último caso, categorizado como não agressivo.

Figura 22: Matriz de confusão do modelo MLP de duas camadas ocultas com 32 neurônios por camada e função de ativação tangente hiperbólica (modelo treinado e validado com validação cruzada 10-*fold*). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados.



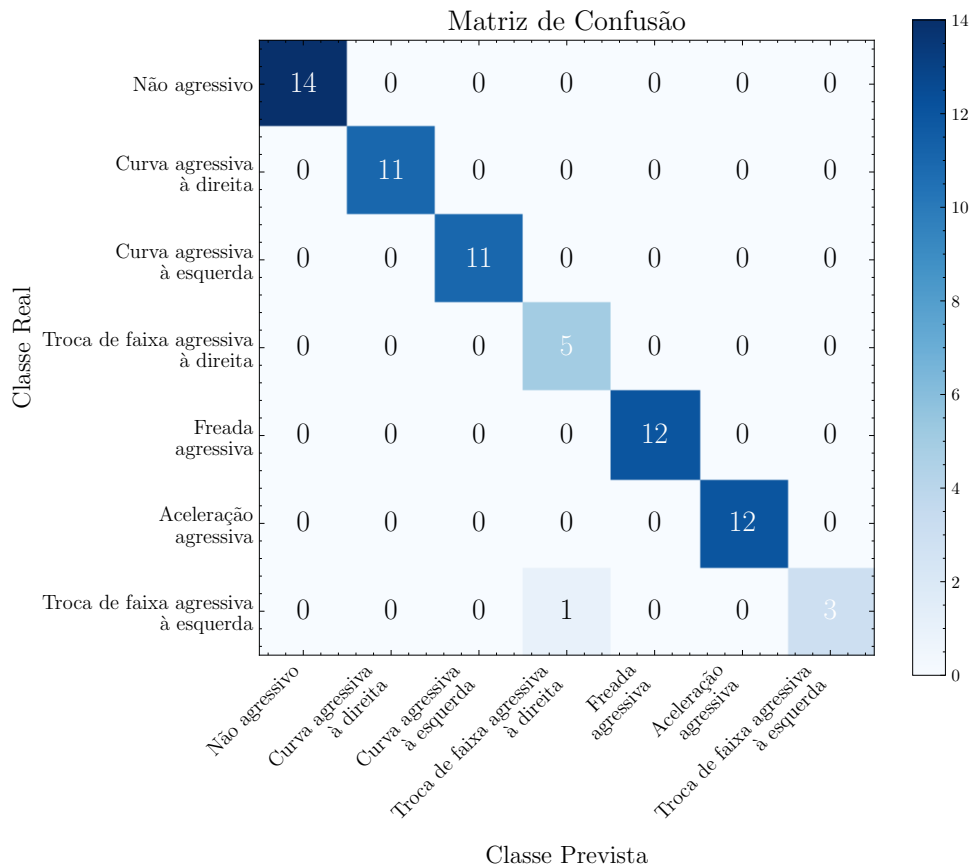
Fonte: Autor, 2022.

A Figura 25 apresenta o *boxplot* com os valores de acurácia obtidos para os melhores modelos considerando os 10 particionamentos realizados durante o treino e validação. Embora os modelos baseados em engenharia de atributos tenham valores de médias e medianas maiores, alguns resultados obtidos para o modelo 1D-Alexnet alcançaram acurácia elevada, compatível com os resultados obtidos com os modelos SVM e MLP.

4.9 Dados brutos coletados através do *phyphox*

O conjunto de dados coletados apresenta diferentes eventos que são definidos e rotulados por intervalos de tempo de início e fim, possibilitando a caracterização das sequências do sinais. A Figura 26 mostra um conjunto de dados brutos coletados de aceleração linear nas direções x e y e velocidade angular na direção z . A Figura 27 mostra os mesmos eventos, no entanto, suavizados pelo filtro Savitzky-Golay anteriormente proposto.

Figura 23: Matriz de confusão do modelo 1D-AlexNet (modelo treinado e validado com validação cruzada 10-*fold*). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados.



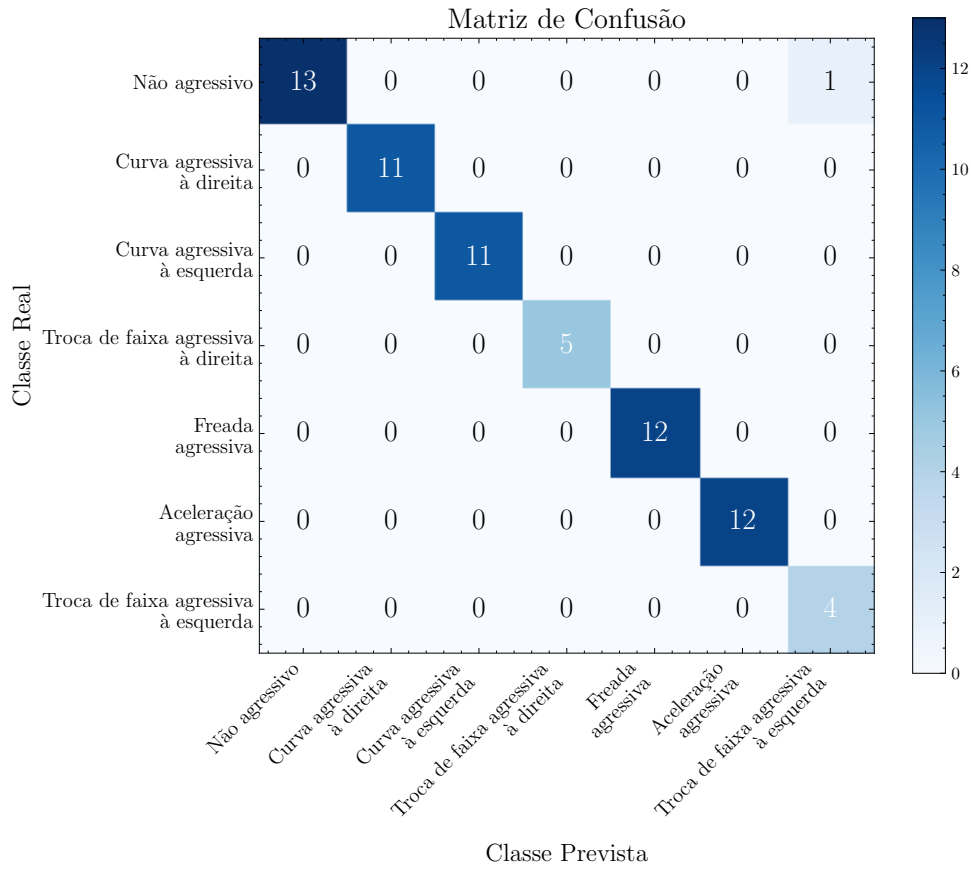
Fonte: Autor, 2022.

É possível identificar que as características dos sinais apresentados para cada evento são correspondentes àquelas descritas no subitem 4.1. Em síntese, sinais de aceleração nas direções x e y de eventos não agressivos são caracterizados por oscilações de baixa amplitude ao longo da janela do evento. Os eventos de curvas e trocas de faixa agressivas apresentam variações mais significativas no eixo x do sinal de aceleração. Por outro lado, sinais de freada e aceleração apresentam variações mais significativas no eixo y do sinal de aceleração.

Sinais de velocidade angular na direção z de eventos não agressivos, freada e aceleração não apresentam uma variação acentuada. Por outro lado, eventos de curvas e trocas de faixa agressivas, apresentam comportamentos distintos e característicos com base na mudança de direção dos movimentos.

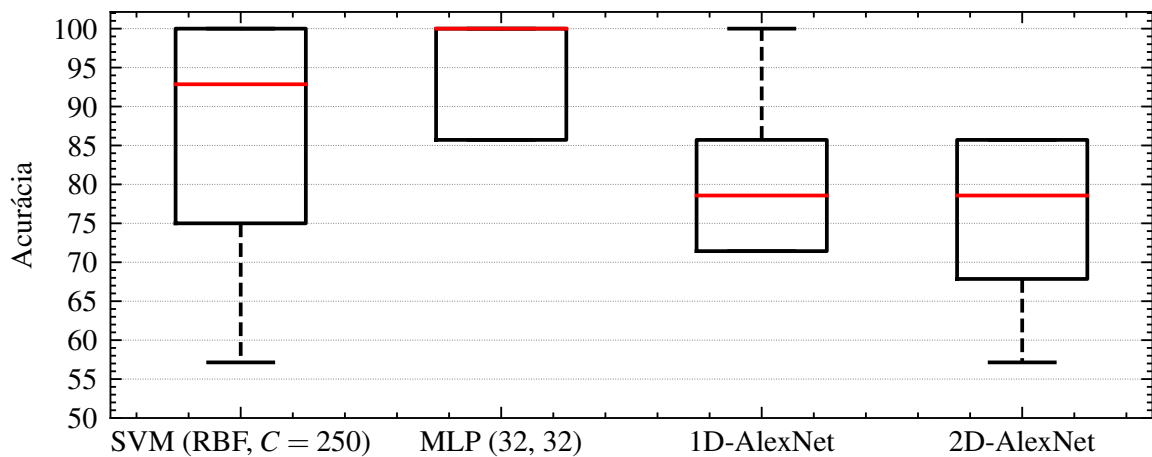
Nota-se que há pequenas diferenças entre os sinais apresentados pelo banco de dados do subitem 4.1 e os dados deste experimento. Essa diferença pode estar relacionada a um possível vazamento de dados entre eixos. O vazamento de dados é influenciado pelo

Figura 24: Matriz de confusão do modelo 2D-AlexNet (modelo treinado e validado com validação cruzada 10-*fold*). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados.



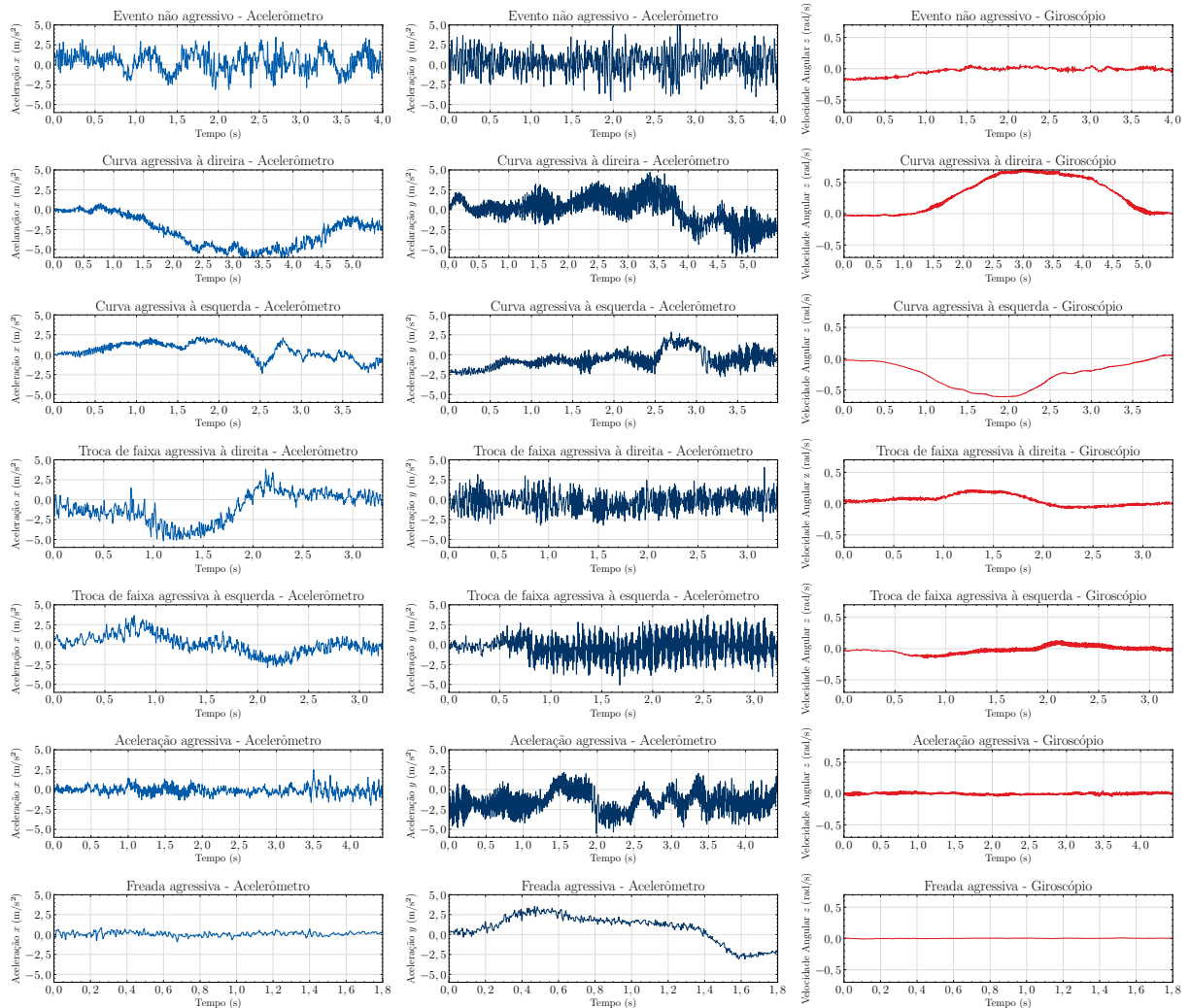
Fonte: Autor, 2022.

Figura 25: *Boxplot* das distribuições da métrica acurácia para os 10 particionamentos. *Boxplot* que se estendem do primeiro quartil, abaixo, ao terceiro quartil, acima. Linha mediana em vermelho e *whiskers* se estendendo até os valores mínimo e máximo.



Fonte: Autor, 2022.

Figura 26: Dados brutos coletados de aceleração linear na direção x (primeira coluna) e y (segunda coluna) e velocidade angular em z (terceira coluna) para eventos não agressivos (primeira linha), curva agressiva à direita e à esquerda (segunda e terceira linhas), troca de faixa agressiva à direita e à esquerda (quarta e quinta linhas), freada agressiva (sexta linha) e aceleração agressiva (sétima linha), respectivamente.

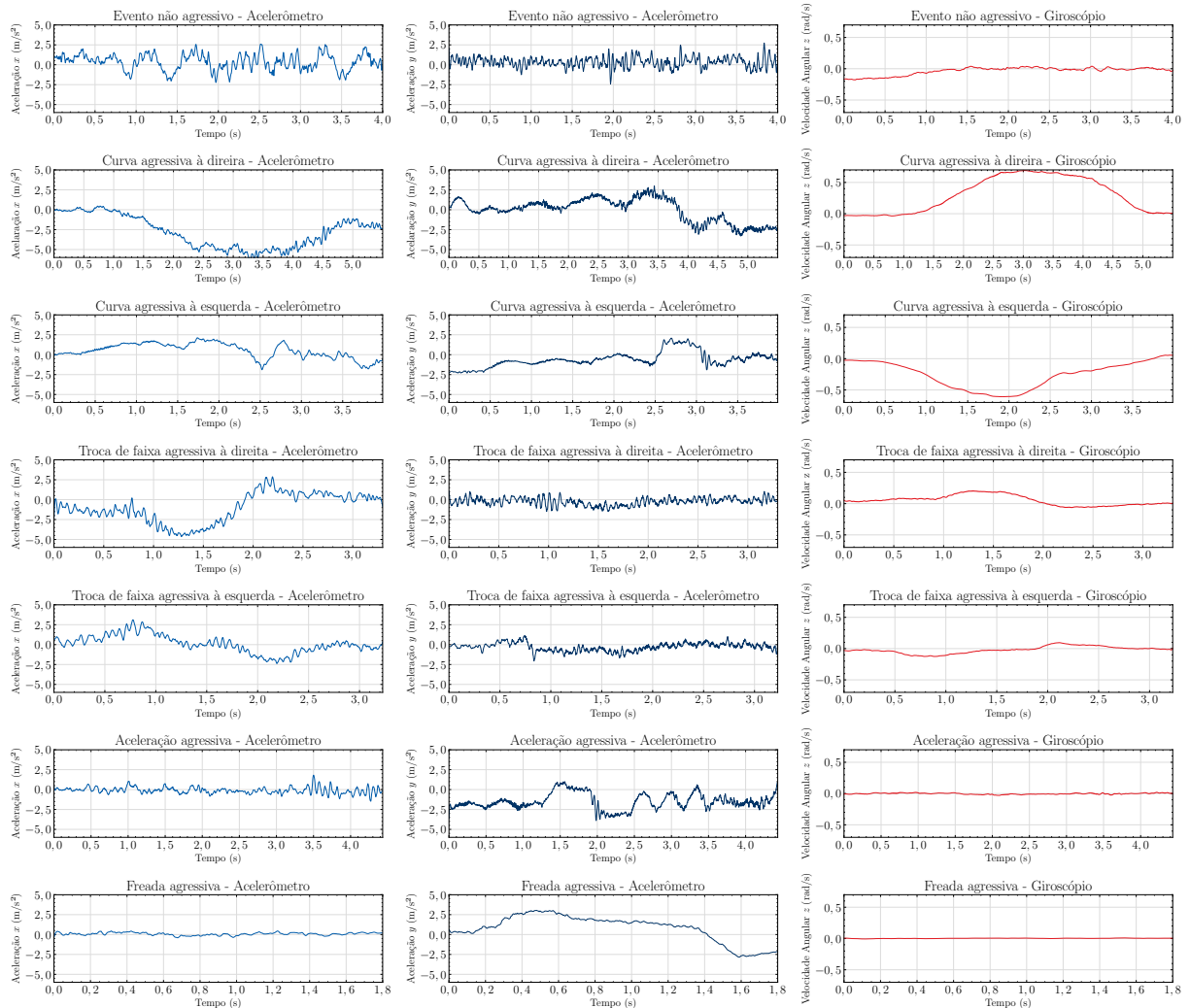


Fonte: Autor, 2022.

nível de inclinação em que o *smartphone* é fixado para a coleta de dados. Com isso, outras componentes tornam-se substancialmente sensíveis ao movimento, resultando e uma variação do sinal em direções distintas. É importante ressaltar que para a coleta de dados foi garantida a alocação adequada do *smartphone* no painel do veículo de forma a mitigar possíveis vazamentos entre eixos.

Verifica-se que os eventos seguem um padrão de série temporal, que pode variar, em geral, em termos de amplitude (dependendo da maior ou menor agressividade do motorista) ou da duração do evento. No entanto, as formas de onda apresentadas, tanto no subitem 4.1 quando neste experimento, sugerem que o comportamento dos eventos

Figura 27: Dados filtrados coletados de aceleração linear na direção x (primeira coluna) e y (segunda coluna) e velocidade angular em z (terceira coluna) para eventos não agressivos (primeira linha), curva agressiva à direita e à esquerda (segunda e terceira linhas), troca de faixa agressiva à direita e à esquerda (quarta e quinta linhas), freada agressiva (sexta linha) e aceleração agressiva (sétima linha), respectivamente.



Fonte: Autor, 2022.

segue um padrão definido para diferentes motoristas, veículos e sensores.

4.10 Desempenho dos classificadores baseado nos modelos em SVM, MLP, CNN 1D e CNN 2D, diante dos dados unificados

Os modelos em SVM, MLP, 1D-AlexNet e 2D-AlexNet foram avaliados utilizando o conjunto de dados proveniente da unificação entre os subconjuntos apresentados nos subitens 4.1 e 4.9. Vale ressaltar que manteve-se os mesmos atributos anteriormente

utilizados para os dados unificados. Os modelos são treinados e validados a partir da estratégia de validação cruzada *k-fold* com parâmetro *k* igual a 10.

Nessas condições, o desempenho dos modelos baseados em SVM são apresentados na Tabela 10. O melhor desempenho geral é obtido pelo modelo SVM com *kernel* RBF e parâmetro de regularização $C = 1000$, cujos os valores de acurácia e *F1-score* macro e micro são de 91,18%, 90,04% e 91,18%, respectivamente. Com base no modelo de melhor desempenho durante a validação cruzada, a Figura 28 apresenta sua matriz de confusão obtida a partir de todo o conjunto de dados unificados.

Tabela 10: Medida de desempenho através das métricas de acurácia e *F1-score* (médias macro e micro) dos diferentes modelos baseados em SVM. Avaliação realizada com o conjunto de dados unificados. Valores médios obtidos através de 10-*fold*.

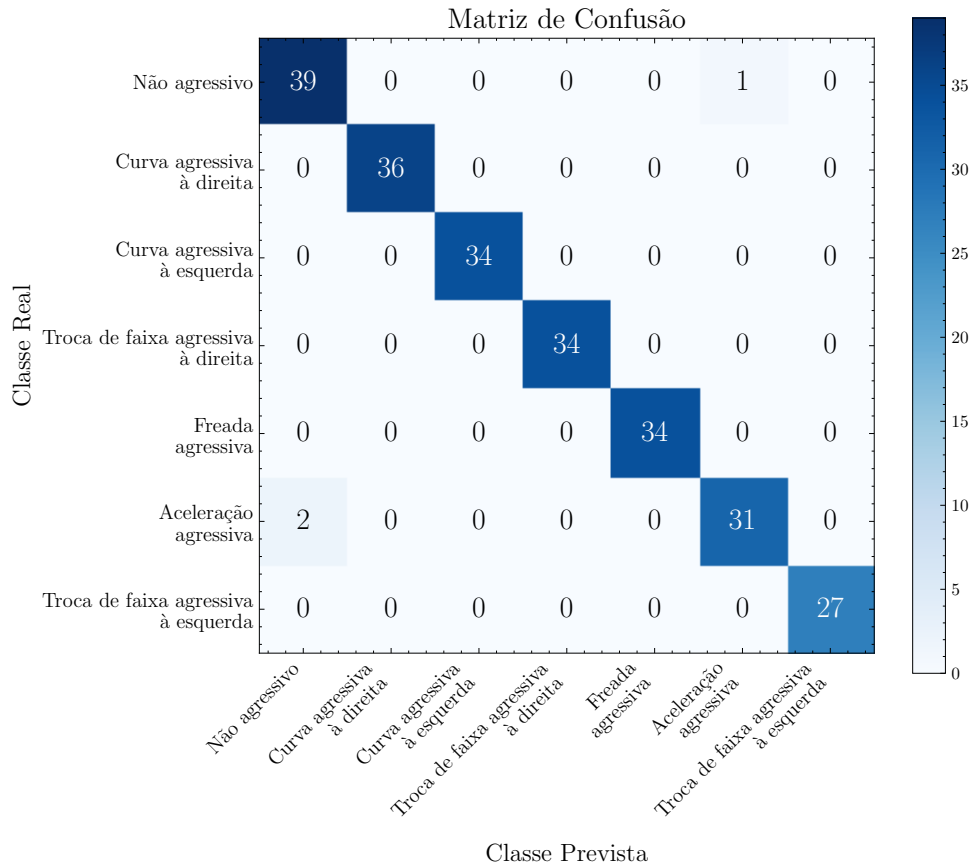
Métricas	Parâmetro C	Linear	Polinômio 2ª ordem	Polinômio 3ª ordem	Polinômio 4ª ordem	Polinômio 6ª ordem	Polinômio 9ª ordem	Sigmoidal	RBF
Acurácia	1	89,47%	67,53%	65,02%	63,35%	64,18%	63,35%	33,22%	72,63%
<i>F1-score</i> macro	1	89,77%	68,28%	65,92%	64,43%	66,67%	65,25%	23,98%	72,44%
<i>F1-score</i> micro	1	89,47%	67,53%	65,02%	63,35%	64,18%	63,35%	33,22%	72,63%
Acurácia	50	89,05%	88,62%	85,25%	82,34%	73,08%	65,92%	32,77%	89,93%
<i>F1-score</i> macro	50	89,14%	85,97%	81,88%	79,00%	69,94%	66,65%	30,00%	90,30%
<i>F1-score</i> micro	50	89,05%	88,62%	85,25%	82,34%	73,08%	65,92%	32,77%	89,93%
Acurácia	100	89,05%	89,46%	87,77%	83,99%	76,08%	64,66%	33,18%	90,78%
<i>F1-score</i> macro	100	89,14%	87,87%	83,97%	80,56%	72,77%	63,72%	28,73%	91,02%
<i>F1-score</i> micro	100	89,05%	89,46%	87,77%	83,99%	76,08%	64,66%	33,18%	90,78%
Acurácia	250	89,05%	89,86%	89,06%	86,07%	76,90%	67,59%	30,06%	90,34%
<i>F1-score</i> macro	250	89,14%	86,98%	85,67%	82,16%	73,46%	65,57%	33,31%	90,38%
<i>F1-score</i> micro	250	89,05%	89,86%	89,06%	86,07%	76,90%	67,59%	33,19%	90,34%
Acurácia	500	89,05%	89,86%	90,31%	85,71%	77,68%	70,56%	32,36%	90,34%
<i>F1-score</i> macro	500	89,14%	86,86%	86,75%	81,68%	73,26%	67,73%	28,91%	90,04%
<i>F1-score</i> micro	500	89,05%	89,86%	90,31%	85,71%	77,68%	70,56%	32,36%	90,34%
Acurácia	1000	89,05%	89,44%	89,89%	85,27%	79,35%	71,01%	31,94%	91,18%
<i>F1-score</i> macro	1000	89,14%	86,98%	86,19%	81,61%	74,12%	67,21%	28,61%	90,04%
<i>F1-score</i> micro	1000	89,05%	89,44%	89,89%	85,27%	79,35%	71,01%	31,94%	91,18%
Acurácia	3000	89,05%	89,44%	89,49%	84,86%	80,63%	74,80%	31,50%	88,64%
<i>F1-score</i> macro	3000	89,14%	86,62%	85,89%	81,08%	76,41%	71,44%	28,01%	88,56%
<i>F1-score</i> micro	3000	89,05%	89,44%	89,49%	84,86%	80,63%	74,80%	31,50%	88,64%

Fonte: Autor, 2022.

A partir da Figura 28, observa-se que eventos de curva agressiva à direita e à esquerda, troca de faixa agressiva à direita e à esquerda e freada agressiva apresentam uma TPR de 100%. O modelo obteve três previsões incorretas referentes ao reconhecimento de eventos de aceleração agressiva como evento não agressivo e um evento agressivo como aceleração agressiva.

O desempenho dos modelos baseados em redes neurais MLP são avaliados propondo diferentes combinações de hiperparâmetros da rede. Os resultados de desempenho individuais de cada modelo podem ser visualizados nas Tabelas 11, 12 e 13. O melhor desempenho geral é obtido pelo modelo MLP de duas camadas ocultas com 512 neurônios na primeira camada, 32 neurônios na segunda camada e função de ativação tangente hi-

Figura 28: Matriz de confusão do modelo SVM com *kernel* RBF e parâmetro C igual a 1000 (modelo treinado e validado com validação cruzada 10-*fold*). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados unificados.



Fonte: Autor, 2022.

perbólica. Tal desempenho foi alcançado utilizando otimizador Adam. Os valores obtidos de acurácia, $F1$ -score macro e micro são de 92,45%, 91,84% e 92,45%, respectivamente. Com base no modelo de melhor desempenho durante a validação cruzada, a Figura 29 apresenta sua matriz de confusão obtida a partir de todo o conjunto de dados unificados.

A partir da matriz de confusão do modelo MLP, observa-se que eventos de curva agressiva à direita e à esquerda, troca de faixa agressiva à direita e à esquerda e freada agressiva apresentam uma TPR de 100%. O modelo obteve cinco previsões incorretas referente ao reconhecimento de eventos não agressivo como aceleração agressiva e vice-versa.

O desempenho do modelo 1D-AlexNet é apresentado na Tabela 14. Os valores obtidos de acurácia e $F1$ -score macro e micro são de 78,55%, 72,69% e 78,55%, respectivamente. Com base no modelo de melhor desempenho durante a validação cruzada, a Figura 30 apresenta sua matriz de confusão obtida a partir de todo o conjunto de dados unificados.

Tabela 11: Medida de desempenho através das métricas de acurácia e *F1-score* (médias macro e micro) dos diferentes modelos baseados na rede MLP de uma única camada oculta. Avaliação realizada com o conjunto de dados unificados. Valores médios obtidos através de 10-*fold*.

Métricas	Neurônios (N_1)	(N_1) (Adam, ReLU)	(N_1) (Adam, Tanh)	(N_1) (SGD, ReLU)	(N_1) (SGD, Tanh)
Acurácia	5	29,07%	55,85%	18,34%	39,98%
<i>F1-score</i> macro	5	24,43%	44,18%	7,36%	28,08%
<i>F1-score</i> micro	5	29,08%	55,85%	18,44%	39,98%
Acurácia	32	89,11%	88,17%	84,07%	89,09%
<i>F1-score</i> macro	32	87,66%	87,23%	81,47%	89,11%
<i>F1-score</i> micro	32	89,11%	88,17%	84,07%	89,09%
Acurácia	64	86,56%	90,29%	87,39%	91,86%
<i>F1-score</i> macro	64	86,58%	90,22%	85,12%	91,98%
<i>F1-score</i> micro	64	86,56%	90,29%	87,39%	91,86%
Acurácia	128	90,36%	92,05%	85,72%	91,18%
<i>F1-score</i> macro	128	89,63%	91,51%	84,64%	90,99%
<i>F1-score</i> micro	128	90,36%	92,05%	85,72%	91,18%
Acurácia	256	85,74%	91,21%	89,09%	91,20%
<i>F1-score</i> macro	256	86,40%	91,41%	88,48%	91,29%
<i>F1-score</i> micro	256	85,74%	91,21%	89,09%	91,20%
Acurácia	512	88,70%	92,07%	89,06%	90,34%
<i>F1-score</i> macro	512	88,27%	92,05%	87,59%	89,01%
<i>F1-score</i> micro	512	88,70%	92,07%	89,06%	90,34%
Acurácia	1024	87,84%	92,07%	89,06%	90,34%
<i>F1-score</i> macro	1024	88,27%	92,05%	87,59%	89,01%
<i>F1-score</i> micro	1024	87,84%	92,07%	89,06%	90,34%

Fonte: Autor, 2022.

A partir da matriz de confusão da Figura 30, observa-se que eventos não agressivos, curva agressiva à direita e à esquerda, freada agressiva e troca de faixa agressiva à esquerda apresentam uma TPR de 100%. O modelo obteve duas previsões incorretas referentes ao reconhecimento de troca de faixa agressiva à direita e aceleração agressiva como eventos não agressivo.

Os resultados de desempenho do modelo 2D-AlexNet podem ser visualizados na Tabela 15. Os valores obtidos de acurácia e *F1-score* macro e micro são de 64,35%, 57,27% e 64,35%, respectivamente. Com base no modelo de melhor desempenho na validação cruzada, a Figura 31 apresenta sua matriz de confusão obtida a partir de todo o conjunto de dados unificados.

Com base na matriz de confusão da Figura 31, verifica-se que eventos de troca de faixa agressiva à direita, freada agressiva e aceleração agressiva apresentam uma TPR de 100%. Por outro lado, o modelo apresenta cinco previsões incorretas, sendo estas, no

Tabela 12: Medida de desempenho através das métricas de acurácia e $F1$ -score (médias macro e micro) dos diferentes modelos baseados na rede MLP de duas camadas ocultas. Avaliação realizada com o conjunto de dados unificados. Valores médios obtidos através de 10-fold.

Métricas	Neurônios (N_2)	(128, N_2) (Adam, ReLU)	(512, N_2) (Adam, Tanh)	(1024, N_2) (SGD, ReLU)	(64, N_2) (SGD, Tanh)
Acurácia	5	16,38%	90,36%	11,90%	89,49%
$F1$ -score macro	5	6,70%	87,75%	4,85%	88,68%
$F1$ -score micro	5	16,38%	90,36%	11,90%	89,49%
Acurácia	32	92,45%	92,45%	59,24%	91,20%
$F1$ -score macro	32	91,67%	91,84%	51,65%	90,74%
$F1$ -score micro	32	92,45%	92,45%	59,24%	91,20%
Acurácia	64	87,80%	89,09%	89,49%	91,21%
$F1$ -score macro	64	87,61%	87,97%	89,48%	91,46%
$F1$ -score micro	64	87,80%	89,09%	89,49%	91,21%
Acurácia	128	91,61%	89,47%	90,36%	90,38%
$F1$ -score macro	128	81,09%	84,63%	81,57%	81,92%
$F1$ -score micro	128	91,61%	89,47%	90,36%	90,38%
Acurácia	256	87,81%	89,93%	89,96%	90,74%
$F1$ -score macro	256	87,00%	90,04%	89,01%	90,43%
$F1$ -score micro	256	87,81%	89,93%	89,96%	90,74%
Acurácia	512	91,14%	90,34%	90,34%	92,45%
$F1$ -score macro	512	91,69%	89,99%	90,79%	92,33%
$F1$ -score micro	512	91,14%	90,34%	90,34%	92,45%
Acurácia	1024	86,54%	89,91%	90,34%	89,89%
$F1$ -score macro	1024	85,48%	89,62%	90,05%	89,70%
$F1$ -score micro	1024	86,54%	89,91%	90,34%	89,89%

Fonte: Autor, 2022.

reconhecimento de eventos não agressivos como freada agressiva, curva agressiva à direita como curva agressiva à esquerda, curva agressiva à esquerda como curva agressiva à direita e troca de faixa agressiva à esquerda como troca de faixa agressiva à direita.

A Figura 32 apresenta o *boxplot* com os valores de acurácia obtidos para os melhores modelos considerando os 10 particionamentos realizados durante o treino e validação. Da mesma forma como apresentado na seção 4.8, embora os modelos baseados em engenharia de atributos tenham valores de médias e medianas maiores, alguns resultados obtidos para o modelo 1D-Alexnet alcançaram acurácia elevada, compatível com os resultados obtidos com os modelos SVM e MLP.

Tabela 13: Medida de desempenho através das métricas de acurácia e $F1$ -score (médias macro e micro) dos diferentes modelos baseados na rede MLP de três camadas ocultas. Avaliação realizada com o conjunto de dados unificados. Valores médios obtidos através de 10-fold.

Métricas	Neurônios (N_3)	(128, 32, N_3) (Adam, ReLU)	(512, 32, N_3) (Adam, Tanh)	(1024, 128, N_3) (SGD, ReLU)	(64, 512, N_3) (SGD, Tanh)
Acurácia	5	63,04%	89,87%	40,31%	89,53%
$F1$ -score macro	5	54,90%	88,13%	32,93%	88,94%
$F1$ -score micro	5	63,04%	89,87%	40,31%	89,53%
Acurácia	32	90,72%	92,45%	89,49%	89,51%
$F1$ -score macro	32	78,10%	81,46%	82,40%	83,54%
$F1$ -score micro	32	90,72%	92,45%	89,49%	89,51%
Acurácia	64	91,56%	91,61%	87,84%	88,53%
$F1$ -score macro	64	90,10%	91,30%	87,34%	88,94%
$F1$ -score micro	64	91,56%	91,61%	87,84%	88,53%
Acurácia	128	90,34%	90,79%	87,37%	90,36%
$F1$ -score macro	128	89,72%	89,91%	87,62%	89,79%
$F1$ -score micro	128	90,34%	90,79%	87,37%	90,36%
Acurácia	256	90,34%	89,94%	87,86%	89,90%
$F1$ -score macro	256	88,95%	89,68%	88,17%	89,90%
$F1$ -score micro	256	90,34%	89,94%	87,86%	89,90%
Acurácia	512	88,68%	87,36%	87,86%	89,91%
$F1$ -score macro	512	87,35%	86,41%	90,27%	88,55%
$F1$ -score micro	512	88,68%	87,36%	87,86%	89,91%
Acurácia	1024	88,62%	87,41%	91,63%	89,09%
$F1$ -score macro	1024	88,50%	87,74%	91,91%	88,18%
$F1$ -score micro	1024	88,62%	87,41%	91,63%	89,09%

Fonte: Autor, 2022.

Tabela 14: Medida de desempenho através das métricas de acurácia e $F1$ -score (médias macro e micro) do modelo 1D-AlexNet. Avaliação realizada com o conjunto de dados unificados. Valores médios obtidos através de 10-fold.

Modelo	Otimizador	Acurácia	$F1$ -score macro	$F1$ -score micro
1D-AlexNet	Adam	78,55%	72,69%	78,55%

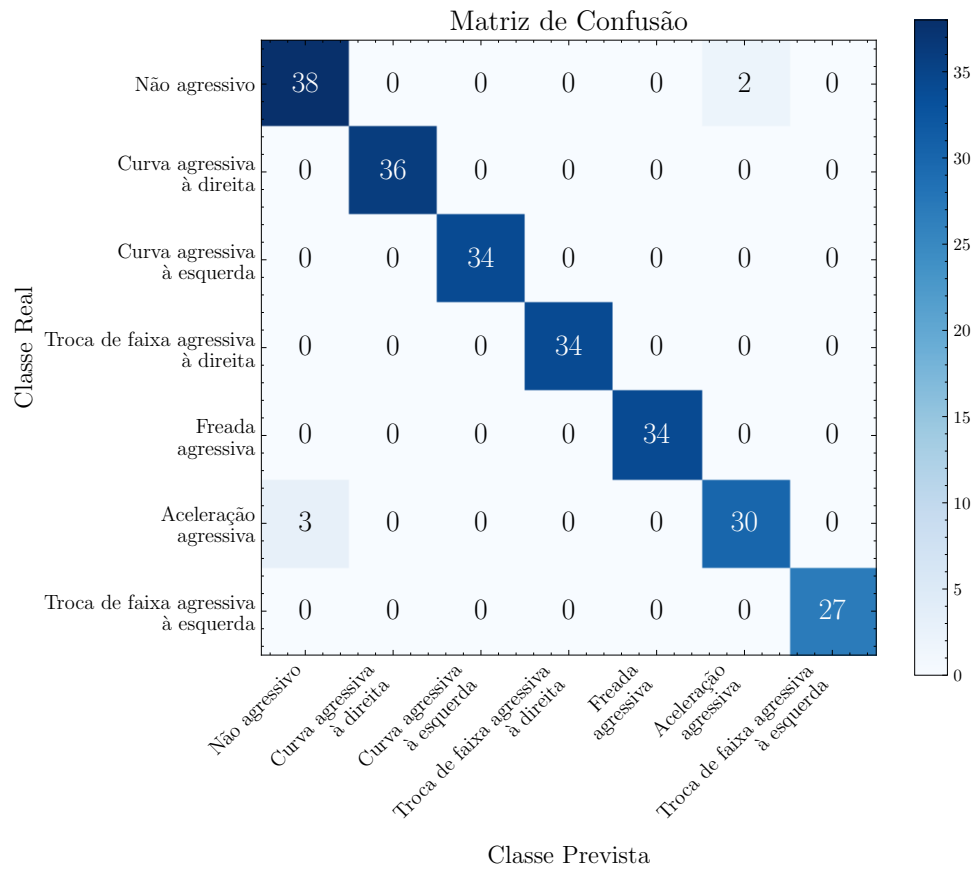
Fonte: Autor, 2022.

Tabela 15: Medida de desempenho através das métricas de acurácia e $F1$ -score (médias macro e micro) do modelo 2D-AlexNet. Avaliação realizada com o conjunto de dados unificados. Valores médios obtidos através de 10-fold.

Modelo	Otimizador	Acurácia	$F1$ -score macro	$F1$ -score micro
2D-AlexNet	Adam	64,35%	57,27%	64,35%

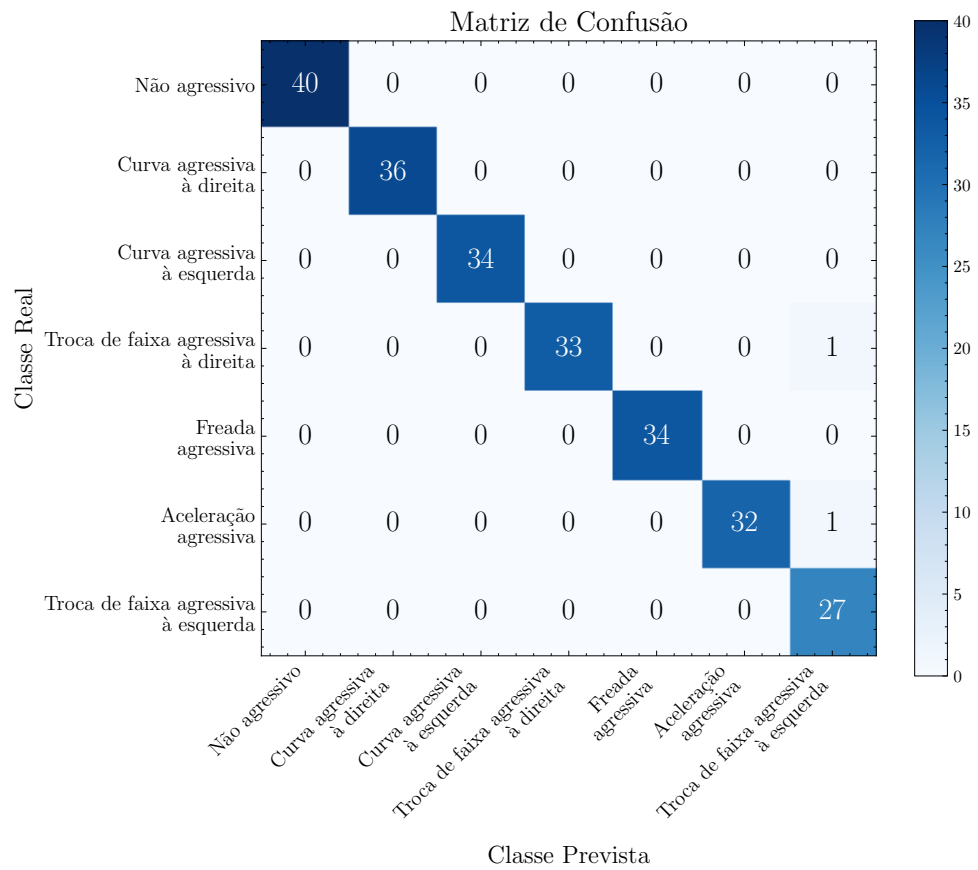
Fonte: Autor, 2022.

Figura 29: Matriz de confusão do modelo MLP de duas camadas ocultas com 512 neurônios na primeira camada, 32 neurônios na segunda camada e função de ativação tangente hiperbólica (modelo treinado e validado com validação cruzada 10-*fold*). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados unificados.



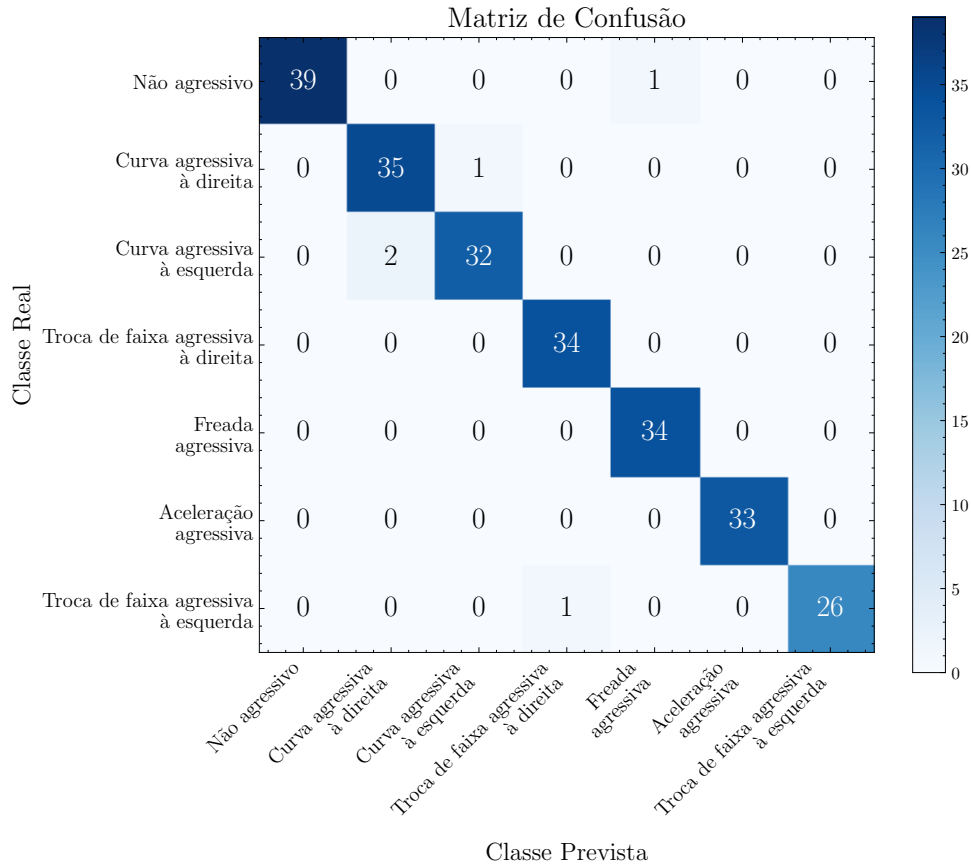
Fonte: Autor, 2022.

Figura 30: Matriz de confusão do modelo 1D-AlexNet (modelo treinado e validado com validação cruzada 10-*fold*). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados unificados.



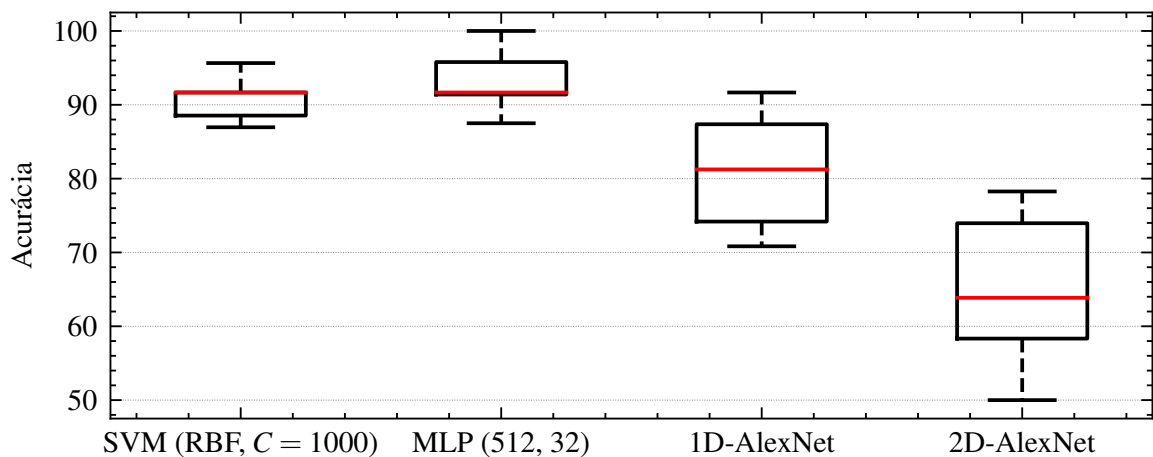
Fonte: Autor, 2022.

Figura 31: Matriz de confusão do modelo 2D-AlexNet (modelo treinado e validado com validação cruzada 10-*fold*). Matriz de confusão obtida a partir da avaliação de todo o conjunto de dados unificados.



Fonte: Autor, 2022.

Figura 32: *Boxplot* das distribuições da métrica acurácia para os 10 particionamentos. *Boxplot* que se estendem do primeiro quartil, abaixo, ao terceiro quartil, acima. Linha mediana em vermelho e *whiskers* se estendendo até os valores mínimo e máximo.



Fonte: Autor, 2022.

5 CONCLUSÃO

A implementação de sistemas para monitoramento do comportamento de motoristas tem sido uma área de pesquisa ativa com diferentes finalidades. A detecção e classificação dos eventos de direção permitem o reconhecimento de estilos de direção e potencializam a criação de serviços personalizados. Além disso, as soluções que incorporam a aplicação de algoritmos de aprendizagem de máquina e aprendizagem profunda, utilizando os dados de sensores inerciais, tem sido amplamente exploradas pela capacidade de caracterização dos eventos e a possibilidade de construção de sistemas robustos e de alto desempenho.

Este trabalho propõe e avalia diferentes estratégias de aprendizagem de máquina aplicados ao problema de reconhecimento de padrões de agressividade ao volante, através dos eventos de direção. Para tanto, realiza-se uma revisão bibliográfica sobre a caracterização do comportamento de motoristas e diferentes técnicas e algoritmos de reconhecimento de padrões que realizam a classificação a partir de sinais de sensores inerciais. Com isso, explora-se a abrangência das análises que envolvem esses métodos.

Este trabalho estabelece, detalhadamente, os conceitos necessários para o desenvolvimento dos modelos de reconhecimento de eventos de direção, bem como os algoritmos e arquiteturas utilizadas para o mapeamento dos dados obtidos. Além disso, são apresentadas as técnicas utilizadas para o pré-processamento dos dados que tornam o padrão e a tendência dos sinais explícitos. O texto apresenta uma abordagem com possíveis estratégias de extração e seleção dos atributos de sinais temporais, a fim de potencializar o desempenho da classificação. Além disso, é utilizada a técnica de gráficos de recorrência para representar as séries temporais em imagens e avaliar o modelo baseado em CNN 2D.

A primeira etapa no desenvolvimento deste trabalho envolveu a utilização de um conjunto de dados rotulados preexistente, detalhado no subitem 3.3. Desse conjunto de dados foram obtidos a parametrização do filtro Savitzky–Golay e os atributos mais relevantes dos sinais utilizados. Posteriormente, um arranjo experimental foi proposto para aumentar a quantidade de amostras para treinamento e validação. Além disso, foi possível avaliar os modelos a partir de eventos de direção obtidos por diferentes motoristas, veículos e

sensores. Com isso, os modelos baseados em algoritmos tradicionais de aprendizagem de máquina, tal com o SVM e o MLP e os modelos de aprendizagem profunda, assim como a arquitetura AlexNet, foram avaliados a partir do conjunto de dados do subitem 3.3 e do conjunto de dados unificados, utilizando as mesmas metodologias.

Os melhores resultados dos modelos SVM, MLP, 1D-AlexNet e 2D-AlexNet, baseados na validação cruzada *10-fold*, atingiram valores de acurácia 91,18%, 92,45%, 78,55% e 64,35%, respectivamente, considerando que a acurácia foi obtida através da média de acurácia das 10 partições do conjunto de validação. Ressalta-se que não foi realizada a partição dos dados em conjunto de teste, uma vez que os bancos de dados apresentam um número limitado de amostras de eventos de direção.

O alto desempenho dos modelos SVM e MLP é potencializado pelo uso do algoritmo DTW que foi fundamental no agrupamento dos dados, comprovando sua relevância junto aos atributos selecionados. No entanto, visualizando os resultados apresentados pelas matrizes de confusão que representam o desempenho dos modelos selecionados diante da inferência de todas as amostras do conjunto de dados, é possível notar que a estrutura 1D-AlexNet apresenta robustez para o reconhecimento dos eventos de direção, apresentando desempenho superior aos outros modelos. Os resultados de acurácia, obtidos pela validação cruzada *10-fold*, indicam que o modelo 1D-AlexNet apresenta uma maior variância dos resultados com relação aos algoritmos SVM e MLP.

Como benefício, o modelo 1D-AlexNet utiliza os sinais diretamente sem a necessidade de todo o processo de engenharia de atributos e representações em gráficos de recorrência. Além disso, o custo computacional do modelo CNN 1D, tanto para treinamento quanto para inferência, é consideravelmente baixo. É possível que para um conjunto de dados com maior número de amostras de eventos devidamente segmentados, o desempenho do modelo 1D-AlexNet seja potencializado.

Os resultados são consistentes para a avaliação de dados coletados através de sensores de movimento inercial, no nosso caso acelerômetros e giroscópios. Os dados abrangem os eventos de direção mais recorrentes durante a conduta veicular e caracterizam o comportamento agressivo de motoristas. Ressalta-se que os experimentos para a coleta de dados foram realizados sob condições de estradas regulares e clima estável. Diferentes condições da estrada (por exemplo, buracos e declives) e condições climáticas (por exemplo, chuva e neblina) podem afetar os comportamentos de direção.

5.1 Trabalho publicado

Uma comparação entre modelos fim-a-fim (*end-to-end*) foi apresentado à Revista Sensors-Basel (<https://www.mdpi.com/1424-8220/22/11/4226>) no artigo:

ESCOTTÁ, A.T.; BECCARO, W.; RAMÍREZ, M. A. Evaluation of 1D and 2D Deep Convolutional Neural Networks for Driving Event Recognition. **Sensors**, v. 22, p. 4226-4247, 2022.

REFERÊNCIAS

- AGARWAL, Rahul. **The 5 Classification Evaluation metrics every Data Scientist must know**. [S.l.: s.n.], 2019. Disponível em: <https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226>.
- ALI, Aya Hamdy; ATIA, Ayman; MOSTAFA, Mostafa-Sami M. Recognizing driving behavior and road anomaly using smartphone sensors. **International Journal of Ambient Computing and Intelligence (IJACI)**, IGI Global, v. 8, n. 3, p. 22–37, 2017.
- ALTIN, Cemil; ER, Orhan. Comparison of different time and frequency domain feature extraction methods on elbow gesture's EMG. **European journal of interdisciplinary studies**, v. 2, n. 3, p. 35–44, 2016.
- BECARI, Wesley et al. Microwave-based system for non-destructive monitoring water pipe networks using support vector machine. **IET Science, Measurement & Technology**, IET, v. 10, n. 8, p. 910–915, 2016.
- BENGIO, Yoshua; GOODFELLOW, Ian; COURVILLE, Aaron. **Deep learning**. [S.l.]: MIT press Massachusetts, USA: 2017. v. 1.
- BENJAMINI, Yoav; YEKUTIELI, Daniel. The control of the false discovery rate in multiple testing under dependency. **The Annals of Statistics**, Institute of Mathematical Statistics, v. 29, n. 4, p. 1165–1188, 2001. DOI: 10.1214/aos/1013699998. Disponível em: <https://doi.org/10.1214/aos/1013699998>.
- BHORASKAR, Ravi et al. Wolverine: Traffic and road condition estimation using smartphone sensors. In: IEEE. 2012 fourth international conference on communication systems and networks (COMSNETS 2012). [S.l.: s.n.], 2012. P. 1–6.
- BILLAH, Baki et al. Exponential smoothing model selection for forecasting. **International journal of forecasting**, Elsevier, v. 22, n. 2, p. 239–247, 2006.
- BOSE, Bernhard E; GUYON, Isabelle M; VAPNIK, Vladimir N. A training algorithm for optimal margin classifiers. In: PROCEEDINGS of the fifth annual workshop on Computational learning theory. [S.l.: s.n.], 1992. P. 144–152.

- BOYRAZ, Pinar; ACAR, Memis; KERR, David. Signal modelling and hidden markov models for driving manoeuvre recognition and driver fault diagnosis in an urban road scenario. In: IEEE. 2007 IEEE Intelligent Vehicles Symposium. [S.l.: s.n.], 2007. P. 987–992.
- CARVALHO, Eduardo et al. Exploiting the use of recurrent neural networks for driver behavior profiling. In: IEEE. 2017 International Joint Conference on Neural Networks (IJCNN). [S.l.: s.n.], 2017. P. 3016–3021.
- CASTELLANOS, Juan C; SUSIN, Altamiro A; FRUETT, Fabiano. Embedded sensor system and techniques to evaluate the comfort in public transportation. In: IEEE. 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). [S.l.: s.n.], 2011. P. 1858–1863.
- CASTIGNANI, German et al. Driver behavior profiling using smartphones: A low-cost platform for driver monitoring. **IEEE Intelligent Transportation Systems Magazine**, IEEE, v. 7, n. 1, p. 91–102, 2015.
- CHAI, Jing; ZHAO, Guozhen. Effect of exposure to aggressive stimuli on aggressive driving behavior at pedestrian crossings at unmarked roadways. **Accident Analysis & Prevention**, Elsevier, v. 88, p. 159–168, 2016.
- CHANDRASHEKAR, Girish; SAHIN, Ferat. A survey on feature selection methods. **Computers & Electrical Engineering**, Elsevier, v. 40, n. 1, p. 16–28, 2014.
- CHAVES, Adriana da Costa Ferreira. **Extração de Regras Fuzzy para Máquinas de Vetores Suporte (SVM) para Classificação em Múltiplas Classes**. 2006. F. 225. Tese (Doutorado) – Pontifícia Universidade Católica, Pós-Graduação em Engenharia Elétrica, Rio de Janeiro.
- CHAWLA, Nitesh V; JAPKOWICZ, Nathalie; KOTCZ, Aleksander. Special issue on learning from imbalanced data sets. **ACM SIGKDD explorations newsletter**, ACM New York, NY, USA, v. 6, n. 1, p. 1–6, 2004.
- CHRIST, Maximilian et al. Time Series Feature Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). **Neurocomputing**, v. 307, p. 72–77, 2018. ISSN 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2018.03.067>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231218304843>.
- CRONE, Sven F; KOURENTZES, Nikolaos. Feature selection for time series prediction—A combined filter and wrapper approach for neural networks. **Neurocomputing**, Elsevier, v. 73, n. 10-12, p. 1923–1936, 2010.

- CYBENKO, George. Approximation by superpositions of a sigmoidal function. **Mathematics of control, signals and systems**, Springer, v. 2, n. 4, p. 303–314, 1989.
- CYGANEK, Bogusław; GRUSZCZYŃSKI, Sławomir. Hybrid computer vision system for drivers' eye recognition and fatigue monitoring. **Neurocomputing**, Elsevier, v. 126, p. 78–94, 2014.
- DAI, Jiangpeng et al. Mobile phone based drunk driving detection. In: IEEE. 2010 4th International Conference on Pervasive Computing Technologies for Healthcare. [S.l.: s.n.], 2010. P. 1–8.
- DAPZOL, Nicolas. Driver's behaviour modelling using the Hidden Markov Model formalism. In: 2.2. ECTRI Young researchers seminar, The Hague, the Netherlands. [S.l.: s.n.], 2005. v. 2, p. 2–1.
- DAZA, Iván Garcia et al. Drowsiness monitoring based on driver and driving data fusion. In: IEEE. 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). [S.l.: s.n.], 2011. P. 1199–1204.
- DEMCENKO, A et al. Vehicle's steering signal predictions using neural networks. In: IEEE. 2008 IEEE Intelligent Vehicles Symposium. [S.l.: s.n.], 2008. P. 1181–1186.
- DI LECCE, Vincenzo; CALABRESE, Marco. NN-based measurements for driving pattern classification. In: IEEE. 2009 IEEE Instrumentation and Measurement Technology Conference. [S.l.: s.n.], 2009. P. 259–264.
- ECKMANN, J.-P; KAMPHORST, S. Oliffson; RUELLE, D. Recurrence Plots of Dynamical Systems. **Europhysics Letters (EPL)**, IOP Publishing, v. 4, n. 9, p. 973–977, nov. 1987. DOI: 10.1209/0295-5075/4/9/004. Disponível em: <https://doi.org/10.1209/0295-5075/4/9/004>.
- EREN, Haluk et al. Estimating driving behavior by a smartphone. In: IEEE. 2012 IEEE Intelligent Vehicles Symposium. [S.l.: s.n.], 2012. P. 234–239.
- ESCOTTÁ, Álvaro Teixeira.; BECCARO, Wesley. **Driving Events Dataset: a smartphone inertial measurement unit for driving events**. Versão v1. [S.l.]: Zenodo, mai. 2022. DOI: 10.5281/zenodo.6570972. Disponível em: <https://doi.org/10.5281/zenodo.6570972>.
- FERREIRA JÚNIOR, Jair et al. Driver behavior profiling: An investigation with different smartphone sensors and machine learning. **PLOS ONE**, Public Library of Science, v. 12, n. 4, p. 1–16, abr. 2017. DOI: 10.1371/journal.pone.0174959. Disponível em: <https://doi.org/10.1371/journal.pone.0174959>.

- GADEPALLY, Vijay; KRISHNAMURTHY, Ashok; OZGUNER, Umit. A framework for estimating driver decisions near intersections. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 15, n. 2, p. 637–646, 2013.
- GADEPALLY, Vijay; KURT, Arda et al. Driver/vehicle state estimation and detection. In: IEEE. 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). [S.l.: s.n.], 2011. P. 582–587.
- GALARZA, Eddie E et al. Real time driver drowsiness detection based on driver's face image behavior using a system of human computer interaction implemented in a smartphone. In: SPRINGER. INTERNATIONAL Conference on Information Theoretic Security. [S.l.: s.n.], 2018. P. 563–572.
- GOODFELLOW, Ian J.; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. Cambridge, MA, USA: MIT Press, 2016. <http://www.deeplearningbook.org>.
- GORRY, Peter A. General least-squares smoothing and differentiation by the convolution (Savitzky-Golay) method. **Analytical Chemistry**, ACS Publications, v. 62, n. 6, p. 570–573, 1990.
- GU, Qiong; ZHU, Li; CAI, Zhihua. Evaluation measures of the classification performance of imbalanced data sets. In: SPRINGER. INTERNATIONAL symposium on intelligence computation and applications. [S.l.: s.n.], 2009. P. 461–471.
- GUIÑÓN, José Luis et al. Moving average and Savitzki-Golay smoothing filters using Mathcad. **Papers ICEE**, v. 2007, 2007.
- HAN, Shipeng et al. Random Error Reduction Algorithms for MEMS Inertial Sensor Accuracy Improvement—A Review. **Micromachines**, MDPI, v. 11, n. 11, p. 1021, 2020.
- HARIRI, Behnoosh et al. A yawning measurement method to detect driver drowsiness. **Technical Papers**, 2012.
- HAYKIN, Simon. **Redes neurais: principios e prática**. [S.l.]: Bookman Editora, 2007.
- HIRSCHKIND, Nameer; MOLLICK, Saruque; PARI, Jyo. **Convolutional Neural Network**. [S.l.: s.n.], 2020. Disponível em: <https://brilliant.org/wiki/convolutional-neural-network/>.
- HOSSIN, Mohammad; SULAIMAN, MN. A review on evaluation metrics for data classification evaluations. **International Journal of Data Mining & Knowledge Management Process**, Academy & Industry Research Collaboration Center (AIRCC), v. 5, n. 2, p. 1, 2015.

- HUBEL, David H; WIESEL, Torsten N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. **The Journal of physiology**, Wiley-Blackwell, v. 160, n. 1, p. 106, 1962.
- IMKAMON, T et al. Detection of hazardous driving behavior using fuzzy logic. In: IEEE. 2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. [S.l.: s.n.], 2008. v. 2, p. 657–660.
- JAIN, Jinesh J; BUSSO, Carlos. Analysis of driver behaviors during common tasks using frontal video camera and CAN-Bus information. In: IEEE. 2011 IEEE International Conference on Multimedia and Expo. [S.l.: s.n.], 2011. P. 1–6.
- JO, Jaeik et al. Detecting driver drowsiness using feature-level fusion and user-specific classification. **Expert Systems with Applications**, Elsevier, v. 41, n. 4, p. 1139–1152, 2014.
- JOHNSON, Derick A; TRIVEDI, Mohan M. Driving style recognition using a smartphone as a sensor platform. In: IEEE. 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). [S.l.: s.n.], 2011. P. 1609–1615.
- JÚNIOR, J Ferreira; PESSIN, Gustavo. Análise de perfil de motoristas: Detecção de eventos por meio de smartphones e aprendizado de máquina. In: ANAIS do WOCES 2016 Workshop de Comunicação em Sistemas Embarcados Críticos. [S.l.: s.n.], 2016. P. 76–85.
- KALSOOM, Rizwana; HALIM, Zahid. Clustering the driving features based on data streams. In: IEEE. INMIC. [S.l.: s.n.], 2013. P. 89–94.
- KIRANYAZ, Serkan et al. 1D convolutional neural networks and applications: A survey. **arXiv preprint arXiv:1905.03554**, 2019.
- KIRCHNER, Antje; SIGNORINO, Curtis S. Using support vector machines for survey research. **Survey Practice**, v. 11, n. 1, p. 2715, 2018.
- KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. ImageNet Classification with Deep Convolutional Neural Networks. In: PEREIRA, F. et al. (Ed.). **Advances in Neural Information Processing Systems**. [S.l.]: Curran Associates, Inc., 2012. v. 25.
- KUHN, Max; JOHNSON, Kjell et al. **Applied predictive modeling**. [S.l.]: Springer, 2013. v. 26.
- LAVESSON, Niklas; DAVIDSSON, Paul. Generic methods for multi-criteria evaluation. In: SIAM. PROCEEDINGS of the 2008 SIAM International Conference on Data Mining. [S.l.: s.n.], 2008. P. 541–546.

- LAZAR, Cosmin et al. A survey on filter techniques for feature selection in gene expression microarray analysis. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, IEEE, v. 9, n. 4, p. 1106–1119, 2012.
- LI, Keyong; JIN, Lisheng et al. Effects of driver behavior style differences and individual differences on driver sleepiness detection. **Advances in Mechanical Engineering**, SAGE Publications Sage UK: London, England, v. 7, n. 4, p. 1687814015578354, 2015.
- LI, Nanxiang; JAIN, Jinesh J; BUSSO, Carlos. Modeling of driver behavior in real world scenarios using multiple noninvasive sensors. **IEEE transactions on multimedia**, IEEE, v. 15, n. 5, p. 1213–1225, 2013.
- LIU, Huan; SETIONO, Rudy et al. A probabilistic approach to feature selection—a filter solution. In: CITESEER. ICML. [S.l.: s.n.], 1996. v. 96, p. 319–327.
- MA, Yongfeng et al. A Comparative Study of Aggressive Driving Behavior Recognition Algorithms Based on Vehicle Motion Data. **IEEE Access**, v. 7, p. 8028–8038, 2019. DOI: 10.1109/ACCESS.2018.2889751.
- MAENAKA, Kazusuke. MEMS inertial sensors and their applications. In: IEEE. 2008 5th International Conference on Networked Sensing Systems. [S.l.: s.n.], 2008. P. 71–73.
- MARTINS, Maria Eugénia Graça. Quantis. **Revista de Ciência Elementar**, Casa das Ciências, v. 2, n. 4, 2014.
- MARWAN, Norbert et al. Recurrence plots for the analysis of complex systems. **Physics reports**, Elsevier, v. 438, n. 5-6, p. 237–329, 2007.
- MEIRING, Gys Albertus Marthinus; MYBURGH, Hermanus Carel. A Review of Intelligent Driving Style Analysis Systems and Related Artificial Intelligence Algorithms. **Sensors (Basel, Switzerland)**, MDPI, v. 15, n. 12, p. 30653–30682, dez. 2015.
- MINDLIN, Gabriel M; GILMORE, R. Topological analysis and synthesis of chaotic time series. **Physica D: Nonlinear Phenomena**, Elsevier, v. 58, n. 1-4, p. 229–242, 1992.
- MITROVIC, Dejan. Reliable method for driving events recognition. **IEEE transactions on intelligent transportation systems**, IEEE, v. 6, n. 2, p. 198–205, 2005.
- NING, Huazhong et al. A general framework to detect unsafe system states from multisensor data stream. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 11, n. 1, p. 4–15, 2009.
- NOVAKOVIĆ, Jasmina Dj et al. Evaluation of classification models in machine learning. **Theory and Applications of Mathematics & Computer Science**, v. 7, n. 1, p. 39–46, 2017.

- OLABIYI, Oluwatobi et al. Driver action prediction using deep (bidirectional) recurrent neural network. **arXiv preprint arXiv:1706.02257**, 2017.
- OLIVER, Nuria; PENTLAND, Alex P. Graphical models for driver behavior recognition in a smartcar. In: IEEE. PROCEEDINGS of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No. 00TH8511). [S.l.: s.n.], 2000. P. 7–12.
- ORGANIZATION, World Health et al. **Global status report on road safety 2018: Summary**. [S.l.], 2018.
- ÖZEL, Mehdi. **AirSim Driver Behaviour Dataset**. [S.l.: s.n.], 2020. <https://www.kaggle.com/dasmehdixtr/airsim-driver-behaviour-dataset>. [Online; accessed 19-January-2021].
- PÄRKKÄ, Juha. **Analysis of personal health monitoring data for physical activity recognition and assessment of energy expenditure, mental load and stress**. Helsinki: VTT, 2011.
- PATAN, Krzysztof; KORBICZ, Józef. Fault detection in catalytic cracking converter by means of probability density approximation. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 20, n. 7, p. 912–923, 2007.
- PÉREZ-ENCISO, Miguel; ZINGARETTI, Laura M. A guide on deep learning for complex trait genomic prediction. **Genes**, Multidisciplinary Digital Publishing Institute, v. 10, n. 7, p. 553, 2019.
- PHINYOMARK, Angkoon et al. Feature extraction and reduction of wavelet transform coefficients for EMG pattern classification. **Elektronika ir Elektrotechnika**, v. 122, n. 6, p. 27–32, 2012.
- RASTGOO, Mohammad Naim et al. Automatic driver stress level classification using multimodal deep learning. **Expert Systems with Applications**, Elsevier, v. 138, p. 112793, 2019.
- SABET, Mehrdad et al. A new system for driver drowsiness and distraction detection. In: IEEE. 20TH Iranian Conference on Electrical Engineering (ICEE2012). [S.l.: s.n.], 2012. P. 1247–1251.
- SATHYANARAYANA, Amardeep; SADJADI, Seyed Omid; HANSEN, John HL. Leveraging sensor information from portable devices towards automatic driving maneuver recognition. In: IEEE. 2012 15th International IEEE Conference on Intelligent Transportation Systems. [S.l.: s.n.], 2012. P. 660–665.

- SENIN, Pavel. Dynamic time warping algorithm review. **Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA**, Citeseer, v. 855, n. 1-23, p. 40, 2008.
- SHAHVERDY, Mohammad et al. Driver behavior detection and classification using deep convolutional neural networks. **Expert Systems with Applications**, Elsevier, v. 149, p. 113240, 2020.
- SHARMA, Avinash. Understanding activation functions in neural networks. **Medium.com blog**, v. 30, 2017.
- SHENDRYK, Yuri et al. Deep learning for multi-modal classification of cloud, shadow and land cover scenes in PlanetScope and Sentinel-2 imagery. **ISPRS Journal of Photogrammetry and Remote Sensing**, Elsevier, v. 157, p. 124–136, 2019.
- STAACKS, Sebastian et al. Advanced tools for smartphone-based experiments: phyphox. **Physics education**, IOP Publishing, v. 53, n. 4, p. 045009, 2018.
- AL-SULTAN, Saif; AL-BAYATTI, Ali H; ZEDAN, Hussein. Context-aware driver behavior detection system in intelligent transportation systems. **IEEE transactions on vehicular technology**, IEEE, v. 62, n. 9, p. 4264–4275, 2013.
- SUN, Chao et al. Real-time driver fatigue detection based on eye state recognition. In: **TRANS TECH PUBL. APPLIED mechanics and Materials**. [S.l.: s.n.], 2014. v. 457, p. 944–952.
- TAKEDA, Kazuya et al. International large-scale vehicle corpora for research on driver behavior on the road. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 12, n. 4, p. 1609–1623, 2011.
- TAKENS, Floris. Detecting strange attractors in turbulence. In: **DYNAMICAL systems and turbulence**, Warwick 1980. [S.l.]: Springer, 1981. P. 366–381.
- TANGO, Fabio; BOTTA, Marco. Real-time detection system of driver distraction using machine learning. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 14, n. 2, p. 894–905, 2013.
- THARWAT, Alaa. Classification assessment methods. **Applied Computing and Informatics**, Emerald Publishing Limited, 2020.
- THIEL, Marco et al. Influence of observational noise on the recurrence quantification analysis. **Physica D: Nonlinear Phenomena**, Elsevier, v. 171, n. 3, p. 138–152, 2002.
- TRAN, Duy et al. Real-time detection of distracted driving based on deep learning. **IET Intelligent Transport Systems**, IET, v. 12, n. 10, p. 1210–1219, 2018.

- VAN LY, Minh; MARTIN, Sujitha; TRIVEDI, Mohan M. Driver classification and driving style recognition using inertial sensors. In: IEEE. 2013 IEEE Intelligent Vehicles Symposium (IV). [S.l.: s.n.], 2013. P. 1040–1045.
- VAPNIK, Vladimir N. **Statistical learning theory**. [S.l.]: New York: Wiley, 1998.
- VAROL, Temel; CANAKCI, Aykut; OZSAHIN, Sukru. Artificial neural network modeling to effect of reinforcement properties on the physical and mechanical properties of Al2024–B4C composites produced by powder metallurgy. **Composites Part B: Engineering**, Elsevier, v. 54, p. 224–233, 2013.
- WANG, Wenshuo et al. Driving style classification using a semisupervised support vector machine. **IEEE Transactions on Human-Machine Systems**, IEEE, v. 47, n. 5, p. 650–660, 2017.
- WEI, Liang et al. Multi-source information fusion for drowsy driving detection based on wireless sensor networks. In: IEEE. 2013 Seventh International Conference on Sensing Technology (ICST). [S.l.: s.n.], 2013. P. 850–857.
- WU, Yunan et al. A comparison of 1-D and 2-D deep convolutional neural networks in ECG classification. **arXiv preprint arXiv:1810.07088**, 2018.
- XIE, Shenglong; REN, Guoying; ZHU, Junjiang. Application of a new one-dimensional deep convolutional neural network for intelligent fault diagnosis of rolling bearings. **Science Progress**, SAGE Publications Sage UK: London, England, v. 103, n. 3, p. 0036850420951394, 2020.
- XING, Mengmeng et al. Human Motion Recognition Based On Inertial Sensor. **MS&E**, v. 782, n. 3, p. 032099, 2020.
- YAN, Jia et al. Electronic nose feature extraction methods: A review. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 15, n. 11, p. 27804–27831, 2015.
- YU, Jiadi et al. Fine-grained abnormal driving behaviors detection and identification with smartphones. **IEEE transactions on mobile computing**, IEEE, v. 16, n. 8, p. 2198–2212, 2016.
- YUKSEL, Asim Sinan. **Driving Behavior Dataset**. [S.l.]: Mendeley, 2020.
- ZBILUT, Joseph P; ZALDIVAR-COMENGENES, José-Manuel; STROZZI, Fernanda. Recurrence quantification based Liapunov exponents for monitoring divergence in experimental data. **Physics Letters A**, Elsevier, v. 297, n. 3-4, p. 173–181, 2002.

ZHANG, Jun et al. A deep learning framework for driving behavior identification on in-vehicle CAN-BUS sensor data. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 19, n. 6, p. 1356, 2019.

ZHOU, Cheng; ZHANG, Weidong. A New Process Monitoring Method Based on Waveform Signal by Using Recurrence Plot. **Entropy**, Multidisciplinary Digital Publishing Institute, v. 17, n. 9, p. 6379–6396, 2015.

APÊNDICE A – DADOS SUAVIZADOS PELO ALGORITMO DE SAVITZKY-GOLAY

Considerando um sinal, cujos dados são descritos por um conjunto de pontos $\{x_j, y_j\}$, sendo $j = 1, \dots, N$, onde x_j é variável independente e y_j é o valor observado, o sinal de saída ys_j suavizado pelo filtro Savitzky-Golay pode ser representado por um conjunto de M coeficientes de convolução C_i , conforme equação A.1.

$$ys_j = \sum_{i=-\frac{M-1}{2}}^{\frac{M+1}{2}} C_i y_{j+i}, \quad \frac{M+1}{2} \leq j \leq N - \frac{M-1}{2} \quad (\text{A.1})$$

Como exemplo, suponha a suavização através de um polinômio quadrático com $M = 5$ pontos e vetor de entrada $y = [2, 2, 5, 2, 1, 0, 1, 4, 9]$ com $N = 9$ pontos. Os coeficientes de convolução são calculados (GORRY, 1990) e disponíveis em tabelas, cujos valores dependem do grau do polinômio e do tamanho da janela M . Neste exemplo, com janela de tamanho $M = 5$ e o polinômio quadrático, o conjunto de M coeficientes de convolução C_i , onde $-2 \leq i \leq 2$, apresentam o valores de $C_{-2} = -\frac{3}{35}$, $C_{-1} = \frac{12}{35}$, $C_0 = \frac{17}{35}$, $C_1 = \frac{12}{35}$ e $C_2 = -\frac{3}{35}$.

Nessas condições, as saídas suavizadas ys_j podem ser obtidas utilizando a equação A.2.

$$ys_j = \frac{1}{35}(-3y_{j-2} + 12y_{j-1} + 17y_j + 12y_{j+1} - 3y_{j+2}) \quad (\text{A.2})$$

Dada a restrição de j equação A.1, temos que os pontos suavizados estão contidos no intervalo de $3 \leq j \leq 7$. Isto posto, a saída suavizada ys_3 , por exemplo, pode ser obtida da seguinte forma:

$$ys_3 = \frac{1}{35}(-3y_1 + 12y_2 + 17y_3 + 12y_4 - 3y_5) \quad (\text{A.3})$$

$$ys_3 = \frac{1}{35}(-3 \times 2 + 12 \times 2 + 17 \times 5 + 12 \times 2 - 3 \times 1) \rightarrow ys_3 = 3,543 \quad (\text{A.4})$$

Utilizando o mesmo equacionamento, outros dados suavizados podem ser determinados. Com isso, alguns valores de saídas resultantes são: $ys_4 = 2,857$; $ys_5 = 0,657$; $ys_6 = 0,171$ e $ys_7 = 1,000$.

A depender da implementação do código, pode-se utilizar diferentes estratégias para tratar os dados que estão nas fronteiras. No caso da biblioteca Python SciPy são descritos cinco modos: “mirror”, “nearest”, “constant”, “wrap” e “interp”. Para este trabalho foi utilizado o modo “interp”, no qual nenhuma extensão é utilizada e os valores das fronteiras são considerados para ajuste dos dados.

APÊNDICE B – ATRIBUTOS DE MAIOR RELEVÂNCIA SELECIONADOS ATRAVÉS DO *TSFRESH*

Os resultados apresentados pela metodologia de seleção de características dos sinais pré-processados de aceleração e velocidade angular, indicou que os atributos de maior relevância são: a média, o valor máximo, o valor mínimo, o p-quantil e o coeficiente da regressão linear. Todos esses atributos envolvem operações no domínio do tempo e são obtidos diretamente das janelas de amostras dos sinais.

O valor de média é resultante da soma de todas as amostras presentes na janela do sinal, dividindo o somatório pelo número de amostras envolvidas. Sua definição matemática é apresentada na equação B.1.

$$\bar{x} = \frac{1}{N} \sum_{j=0}^{N-1} x_j \quad (\text{B.1})$$

na qual, N denota o número de amostras do sinal e x_j são j -ésimas amostras do sinal.

O valor máximo retorna o maior valor entre as amostras de dados do sinal, ou seja, $x_{max} = \max(x)$, na qual x é o sinal temporal. Da mesma forma, o valor mínimo retorna o menor valor entre as amostras de dados do sinal, sendo $x_{min} = \min(x)$, onde x é o sinal temporal.

O p-quantil é uma medida de localização que define através de uma ordem p , sendo $0 \leq p \leq 1$, o valor da amostra que proporciona visualizar elementos do conjunto de dados que são inferiores e superiores a $p \times 100\%$. Para obtenção do p-quantil é necessário que as observações das amostras sejam ordenadas de forma crescente. A equação B.2 permite calcular o quantil x_p de ordem p .

$$x_p = \begin{cases} x_{(Np+1)}, & \text{se } Np \text{ não for inteiro} \\ \frac{1}{2} \times (x_{(Np)} + x_{(Np+1)}), & \text{se } Np \text{ for inteiro} \end{cases} \quad (\text{B.2})$$

na qual, N é o número de amostras do sinal e p é a ordem do quantil.

Como exemplo (MARTINS, 2014), suponha uma amostra ordenada $x = (0, 1, 1, 2, 2, 3, 4, 4, 5, 6)$, cujo o número de elementos N é igual a 10. Deseja-se determinar o quantil de ordem $p = 0,25$ da amostra x . Para isso, é necessário determinar o índice do elemento da amostra que representa a ordem do p-quantil estipulado. Inicialmente, calcula-se o valor de Np , onde $Np = N \times p = 2,5$. Observa-se que Np não é um valor inteiro e a primeira condição da equação B.2 é satisfeita. Assim, o índice de $x_{(Np+1)}$ é igual a 3,5, no entanto, somente a parte inteira desse resultado é considerada. Dessa forma, o elemento de índice 3 da amostra define o p-quantil de ordem $p = 0,25$. Consequentemente, o valor do p-quantil é $x_{0,25} = 1$. Verificando a definição, observa-se que 30% dos elementos são iguais ou inferiores ao valor 1 e 90% dos elementos são iguais ou superiores ao valor 1.

O atributo de coeficiente da regressão linear pode retornar diferentes valores. O cálculo desse atributo divide o sinal temporal em blocos de tamanho pré-estabelecido e os dados de cada bloco são ajustados através do método de mínimos quadrados. Com isso, cada bloco apresenta um modelo de regressão linear, descrito pela equação

$$Y_i = \alpha + \beta X_i + \varepsilon_i \quad (\text{B.3})$$

na qual, Y_i é a variável dependente, α é o intercepto ou coeficiente linear, β é o coeficiente angular, X_i é a variável independente e ε_i é o termo de erro que representa outros fatores que influenciam na variável dependente Y_i .

A partir das equações determinadas, pode-se utilizar diferentes funções de agregação (máximo, mínimo e média) para os valores de intercepto, coeficiente angular, entre outros. Para o presente trabalho utilizou-se o cálculo da média dos valores de intercepto.

APÊNDICE C – APLICATIVO *PHYPHOX* PARA EXPERIMENTOS COM *SMARTPHONES*

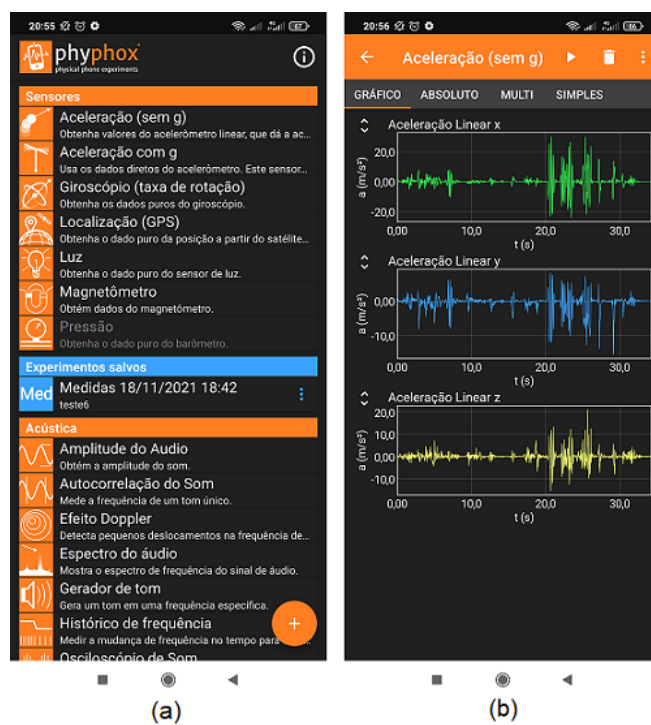
O *phyphox*¹ é um aplicativo criado pela *RWTH Aachen University* e foi projetado para explorar uma variedade de experimentos relacionados à física utilizando os sensores disponíveis em *smartphones*. Dessa forma, o aplicativo tem grande potencial de aplicações por facilitar a difusão e o acesso aos sensores embarcados nos *smartphones*. Além disso, é um aplicativo gratuito, sem publicidades e apresenta o mesmo conjunto de recursos para as interfaces Android e iOS.

Do ponto de vista das funcionalidades, a leitura dos dados brutos dos sensores são disponibilizados em arquivos no formato *CSV* (*Comma-separated values*). Além disso, outros recursos disponíveis no aplicativo são as análises de dados personalizáveis e o acesso remoto para observações experimentais em tempo real a partir de um segundo dispositivo.

A Figura 33 apresenta capturas de tela do *phyphox* no Android. Em (a) tem-se o menu principal com as opções de acesso aos dados brutos dos sensores e os experimentos com análise de dados, categorizados por tópicos. Em (b) tem-se um exemplo de dados coletados de aceleração linear nas direções x , y e z , a partir do acelerômetro.

¹<https://phyphox.org/>

Figura 33: Capturas de tela do *phyphox* no Android. Em (a) tem-se o menu principal com as opções de acesso aos dados brutos dos sensores e os experimentos com análise de dados, categorizados por tópicos. Em (b) tem-se um exemplo de dados coletados de aceleração linear nas direções x , y e z , a partir do acelerômetro.



Fonte: Autor, 2022.